ITC 3/47
Journal of Information Technology and Control
Vol. 47 / No. 3 / 2018
pp. 551-563
DOI 10.5755/j01.itc.47.3.19045

Self-Tuning Method of PID Parameters Based on Belief Rule Base Inference

Received 2017/09/14 | Accepted after revision 2018/07/26

http://dx.doi.org/10.5755/j01.itc.47.3.19045

Self-Tuning Method of PID Parameters Based on Belief Rule Base Inference

Xiao-Bin Xu, Xue Ma, Cheng-Lin Wen
School of Automation, Hangzhou Dianzi University, Hangzhou, China
Da-Rong Huang
School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, 40074, China
Jian-Ning Li
School of Automation, Hangzhou Dianzi University, Hangzhou, China

Corresponding author: xuema1992@163.com; tel.: +86-150-8878-3292

As a generic inference mechanism, the belief rule-based (BRB) system can effectively integrate quantitative information with qualitative knowledge to model causal relationships of complex application systems. Based on the BRB, this paper develops a novel self-tuning strategy of PID parameters such that the output of closed-loop control system generated by PID controller can accurately follow control input. Firstly, the initial belief rule base is abstracted from expert's control experiences to depict the highly nonlinear relationship between the variables of control system and each PID parameter. Secondly, the objective function is established to minimize the error between the given control input and the closed loop output, and then the online optimization method via sequential linear programming is presented to optimize the parameters of BRB system so as to adaptively adjust PID parameters by the optimized BRB system in real time. Typical control simulation experiments of DC motor are implemented to illustrate the advantages of the proposed BRB-PID over widely used neural network-based PID.

KEYWORDS: Belief Rule Base (BRB), PID controller, Sequential linear programming (SLP) algorithm, Evidence Reasoning(ER).

1. Introduction

PID is one of the earliest control strategies proposed in classical control theory, which has been widely used in industrial control systems and achieved good control effect because of its simple form, available robustness and reliability [14]. To a great extent, the performance of PID controller depends on the appropriate selection of PID parameters. Hence, in order to obtain the satisfactory control effect, one has to study on available methods for determining the value of PID parameters, which are the key link in the design of PID controller, and what's more, with the increasing complexities of the structures, functions and operating conditions of controlled objects, this issue becomes more and more important for applications of PID strategy [17]. So experts and scholars have been concentrating on the self-tuning methods for PID parameters, so that the adaptive PID control could be chosen to adapt to the complex and changeable controlled objects, and meet the control requirements with high performance and high precision [16]. In essence, these methods all attempt to explore appropriate models to establish a nonlinear mapping relationship between the variables of a control system (including the given input, actual output, deviation and the deviation change rate, etc) and PID parameters (the proportional, integral, differential coefficients K_P, K_I, K_D, respectively), also there exists various uncertainties in this mapping relationship due to the complexity of controlled objects and various disturbances [8].

Nowadays, artificial intelligence methods have been widely used to turn or set PID parameters adaptively, which greatly enhance tuning effect and efficiency [2], mainly including Expert system-based PID (ES-PID), Fuzzy inference-based PID (FI-PID) and Artificial neural network-based PID (ANN-PID) etc. ES-PID abstracts heuristic rules from expert's knowledge about controlled object and control experience to depict the nonlinear relationship between the control variables and PID parameters, and then, the designed inference engine can infer the corresponding values of PID parameters from the rules activated by the online values of control variables [9]. However, there are some difficult issues one has to face, such as, how to distinguish good knowledge from bad knowledge because the latter will lead to useless, conflicting, even counter-intuitive rules; how to enhance the online learning and updating abilities of expert system and improve completeness and adaptability of the constructed rule base [13]. In order to deal with fuzzy uncertainty of human knowledge, FI-PID introduces the fuzzy rules to model the imprecise relationship between the control variables and PID parameters, and then uses the fuzzy inference engine to adjust the values of PID parameters [7]. Comparing with traditional ES-PID, FI-PID can capture more useful information with uncertainty in expert's knowledge and has better generalization capability. However, similar with ES-PID, it also suffers from some difficulties including poor online learning and updating abilities and incompleteness of fuzzy rule base and so on [10].

ANN-PID uses the hidden layer network structure to construct the connection between the input layer (control variables) and the output layer (PID parameters), and online optimizes network weights to obtain desired values of PID parameters which is a kind of typical adaptive PID control [20]. However, the neural network is a black box system, in which, the physical meanings of network nodes are obscure and even hard to understand for control engineers. Although, when objective functions are given, so many optimization strategies can be used to online adjust the networks weights, the optimized results are easy to fall into local minimum in training process because of the improper initial values of weights or other reasons [15].

It can be concluded that fuzzy methods and ANN methods all have their specific advantages and disadvantages when they are used for self-tuning of PID parameters. We tend to search for such a method which can integrate and magnify advantages of these methods, and avoid their disadvantages. The belief rule base (BRB) system can provide such an inference mechanism to satisfy our desire since it can synthesize available methodologies including fuzzy set theory, expert system, evidential reasoning, multi-attribute decision making and utility theory [1]. The BRB system consists of two main parts: knowledge base (belief rule base) and fusion reasoning model (evidential reasoning). Its superiorities are reflected in the following three aspects [4]: (1) based on utility equivalent principle, it can uniformly describe subjective/objective multi-source uncertainty information with randomness, fuzziness and incompleteness under the framework of belief rules. Compared with the traditional fuzzy rule, the consequent of belief rule is no longer a single hypothesis, but is a belief distribution about all hypotheses. Therefore, it can describe the uncertain relationship between rule antecedent (attributes) and consequent (hypotheses) more flexibly and precisely; (2) it uses Evidence Reasoning (ER) rule to fuse the belief distributions of the activated rules. Some details (rules and attributes weights and so on) are considered in ER fusion process [3]; (3) compared with neural network, the physical meanings of the parameters in BRB (value and weight of attribute, belief distribution, rule weight, etc.) are not obscure, but very clear and easy to be understood by control engineers. Therefore, they can adjust BRB system according to themselves intentions. These parameters can also be trained and optimized using available methods. Therefore, the BRB methodology has attracted wide attentions in various industrial applications including alarm monitoring, fault diagnosis, risk and decision analysis and so on [21, 24].

This paper aims to design a novel adaptive BRB-PID controller to deal with self-tuning of PID parameters. Firstly, the initial belief rule base is abstracted from expert's control experiences to describe the highly nonlinear relationships between the variables of control system and PID parameters. Secondly, in order to reflect the real-time changes of this nonlinear relationship with sampling time, the online optimization method via sequential linear programming is presented to optimize the parameters of BRB system, and then the optimized BRB can adaptively reason out the values of PID parameters such that the output of closed-loop control system generated by the proposed adaptive BRB-PID controller can accurately follow the given input. The typical control experiments of DC motor are implemented to illustrate the advantages of the proposed adaptive BRB-PID control over the widely used adaptive neural network-based PID control.

2. Self-Tuning of PID Parameters via BRB Inference

2.1. Incremental PID Control

The well-known discrete-time incremental PID controller can be expressed as [18]:

$$u(k)=u(k-1)+K_P(e(k)-e(k-1))+\frac{K_IT}{2}(e(k)+e(k-1))$$
$$+\frac{K_DT}{2}(e(k)-2e(k-1)+e(k-2)), \quad (1)$$

where $e(k)=r(k)-y(k)$ is the deviation that the closed-loop output $y(k)$ tracks the given input (the desired closed-loop output) $r(k)$. $u(k)$ is the control effort at time step k. K_P, K_I and K_D are proportional, integral and derivative gains, respectively. T is the sampling period. From (1), the increment can be given as

$$\Delta u(k)=u(k)-u(k-1)=K_Pe_P(k)+K_Ie_I(k)+K_De_D(k), \quad (2)$$

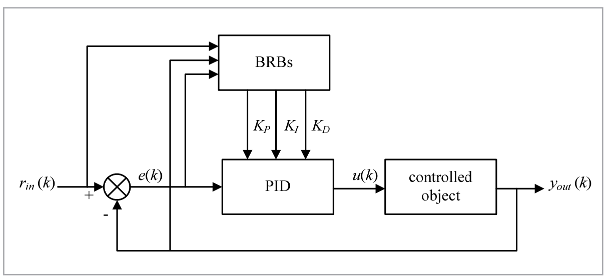
where $e_P=e(k)-e(k-1)$, $e_I=T(e(k)+e(k-1))/2$, $e_D=(e(k)-2e(k-1)+e(k-2))/T$.

2.2. The Design of BRB-PID Controller

Fig. 1 shows the structure of BRB-PID controller, in which, PID has the incremental form as given in (1). The values of the parameters K_P, K_I and K_D can be online estimated by, respectively, constructed three BRBs with the same inputs (r_{in}, e, y_{out}) and the different outputs (K_P, K_I, K_D). The whole modeling and inference procedure can be described as follows. Firstly, based on the expert's control experiences, we can construct BRB models which consist of rules and their parameters including the reference valued of the input variables r_{in}, e, y_{out} (antecedent attributes), the output variables K_P, K_I and K_D (consequent hypothesis) and the values of rule weights and attribute weights; secondly, when the input data are online obtained, they are inputted into the BRB models, and then the activated rules and the corresponding activation weights can be acquired. The ER algorithm is used to fuse the consequent belief distributions of the activated rules. The values of PID parameters can be estimated from the fused belief distribution via utility principle [1]. In this inference process, it should be noted that we must online optimize the parameters of BRB such that the error between the closed loop output deduced from BRB-PID and the control input is minimized because the initial BRB model coming

Figure 1

The structure of BRB-PID controller



from the expert's knowledge may be imprecise. Here, the sequential linear programming algorithm is presented to realize the optimization.

2.2.1. Construction of BRB for the Estimation of PID Parameters

The rth rule R_r in BRB can be modeled as [12]:

$$R_r: \text{If } x_1 \text{ is } A_1^r \wedge x_2 \text{ is } A_2^r \wedge \cdots \wedge x_M \text{ is } A_M^r \\ \text{then}\{(D_1,\beta_{1,r}),(D_2,\beta_{2,r}),\cdots,(D_N,\beta_{N,r})\} \\ \text{with a rule weight } \theta_r \text{ and attribute} \\ \text{weight } \delta_1,\delta_2,\cdots,\delta_M, \quad (3)$$

where x_i (j=1,2,...,N) denotes the ith antecedent attribute with the referential value A_i^r. $\beta_{n,r} \in [0,1]$ (j=1,2,...,N) represents the belief degree to which D_n is believed to be true given the precondition "x_1 is $A_1^r \wedge x_2$ is $A_2^r \wedge ... x_M$ is A_M^r". The belief distribution $\{(D_1,\beta_{1,r}), (D_2,\beta_{2,r}),..., (D_N, \beta_{N,r})\}$ reflects uncertainties caused by the imprecise mapping relationship since it never requires to assign complete belief (β=1) to a certain D. When belief rule base is used to establish the relationship model between r_{in},e,y_{out} and K_P,K_I,K_D, respectively, the physical meanings of BRB parameters are listed in Table 1.

2.2.2. Estimation of PID Parameters Based on Evidential Reasoning

When the input variable is online obtained at time k, denoted as $x_i(k)$, it can be inputted into BRB to estimate the values of PID parameters by the following step.

Step 1: Input transformation

$x_i(k)$ can be transformed to the belief distribution [12]

$$S(x_i(k)) = \{(A_{i,j},\alpha_{i,j}^r(k)),\ j=1,\cdots,J_i\}, \quad (4)$$

here, $\alpha^r_{i,j}(k)$ represents the degree to which $x_i(k)$ approaches $A_{i,j}$ in R_r with $\alpha^r_{i,j}(k) \geq 0$. In detail, if $A_{i,j} \leq x_i(k) \leq A_{i,j+1}$, then

$$\alpha_{i,j}^r = \frac{(A_{i,j+1}-x_i(k))}{(A_{i,j+1}-A_{i,j})},\ \alpha_{i,j+1}^r = 1-\alpha_{i,j}^r. \quad (5)$$

If $x_i(k) \leq A_{i,1}$ or $x_i(k) \geq A_{i,Ji}$, then $\alpha^r_{i,1}$=1 or $\alpha^r_{i,Ji}$=1. Otherwise, $\alpha^r_{i,p}$=0, p=1,2,j-1,j+2,J_i.

Step 2: Calculation of activation weights of belief rules

The activation weight of the rth rule R_r is given as

$$\omega_r = \frac{\theta_r \prod_{i=1}^{M} (\alpha_{i,j}^r)^{\overline{\delta_i}}}{\sum_{l=1}^{L} \theta_l \prod_{i=1}^{M} (\alpha_{i,j}^l)^{\overline{\delta_i}}} \quad (6)$$

where the relative attribute weight $\overline{\delta_i} = \delta_i / \max_{i=1,2,3}\{\delta_i\}$, $\omega_r(k) \in [0,1]$.

Step 3: Estimation of PID parameters using ER algorithm

If R_r is activated, then w_r>0 which can be used to discount the belief distribution $\{(D_1, \beta_{1,r}), (D_2, \beta_{2,r}),..., (D_N,$

Table 1

The parameters of the BRBs for tuning K_P,K_I,K_D

BRB system	The variables and parameters of PID controller
Antecedent attribute	Input variables X=(x_1,x_2,x_3), where x_1=r_{in}, x_2=y_{out}, x_3=e
Reference value set of antecedent attribute A_i={$A_{i,j}$| i=1,2,3; j=1,2,...,J_i}	Reference value of input variable x_i
Antecedent of the rth rule, i=1,2,3, r=1,2,...,L	Reference vector of X in the rth rule, A^r=(A_1^r, A_2^r, A_3^r), $A_i^r \in A_i$
Consequent of the rth rule, $\{(D_1, \beta_{1,r}), (D_2, \beta_{2,r}),..., (D_N, \beta_{N,r})\}$, $\sum_{n=1}^{N} \beta_{n,r} \leq 1$	D_n is the reference value of output K_P or K_I or K_D, when X= A^r, $\beta_{n,r}$ is the belief degree of D_n
Rule weight $\theta_r \in$[0,1]	Relative importance of the rth rule
Attribute weight $\delta_i \in$[0,1]	Relative importance of antecedent attributes

Table Note: for K_P, K_I, K_D, one needs to establish tree different BRBs respectively with different values of BRB parameters, respectively denoted as BRB_P, BRB_I, BRB_D.

$\beta_{N,r})\}$. Using the analytical ER algorithm to fuse these [22], we can obtain the fused belief degree $\hat{\beta}_n$ of the consequent hypothesis D_n.

$$\hat{\beta}_n = \frac{\mu \times \left[\prod_{r=1}^{L} (\omega_r \beta_{n,r} + 1 - \omega_r \sum_{i=1}^{N} \beta_{i,r}) - \prod_{r=1}^{L} (\omega_r \sum_{i=1}^{N} \beta_{i,r}) \right]}{1 - \mu \times \left[\prod_{r=1}^{L} (1 - \omega_r) \right]} \tag{7}$$

$$\mu = \left[\sum_{n=1}^{N} \prod_{r=1}^{L} (\omega_r \beta_{n,r} + 1 - \omega_r \sum_{i=1}^{N} \beta_{i,r}) - (N-1) \prod_{r=1}^{L} (1 - \omega_r \sum_{i=1}^{N} \beta_{i,r}) \right]^{-1} \tag{8}$$

so, the inference output can be represented as the fused belief structure

$$O(X(k)) = \{(D_n, \hat{\beta}_n), n = 1, 2, \cdots, N\}, \tag{9}$$

here $X(k)=(x_1(k),x_2(k),x_3(k))$. As a result, if we know the utility $u(D_n)$ for the consequent hypothesis D_n, the estimated output can be calculated by the expected utility theorem [1].

$$\hat{K}(x(k)) = \sum_{j=1}^{N} u(D_n) \hat{\beta}_n \,. \tag{10}$$

It is noted that the above inference procedure is fit for the three BRBs systems about K_P,K_I,K_D, respectively, so $\hat{K}(x(k)) = K_P(k)$ or $K_I(k)$ or $K_D(k)$.

3. Parameter Optimization of BRB Model Based on SLP

After $K_P(k)$, $K_I(k)$ and $K_D(k)$ are obtained by BRB inference given in Section 2.2.2, PID controller will generate the control value $\hat{u}(k)$, and then $\hat{u}(k)$ is applied to the controlled object to get the closed loop output $\hat{y}(k)$. Although it is possible to respectively establish three fixed belief bases BRB_P,BRB_I,BRB_D by extracting knowledge from experts for getting $K_P(k)$, $K_I(k)$ and $K_D(k)$ at each time step, the performance of the control system can be improved if the rules are fine tuned in real time through the following control objective function

$$\xi(P(k)) = (r_{in}(k) - \hat{y}(k))^2, \tag{11}$$

where, $P(k)=\{ A_{i,j}^T , \beta_{n,r}^T \mid i=1,2,3; j=1,2,...,J_i^a; n=1,2,...,N; r=1,2,...,L^a; T=1,2,3\}$ is an adjustable parameter set about all the activated rules in BRB_P,BRB_I,BRB_D, the superscript a in J_i^a and L^a denotes the number of the activated rules at time step k. "1,2,3" in the superscript T denote BRB_P,BRB_I,BRB_D, respectively. As a result, the optimization objective is to minimize $\xi(P(k))$ by adjusting $P(k)$.

Notice that in most of researches on BRB system, all the parameters of BRB are off-line optimized by training sample set and such a global optimization needs high computational burden [19]. However, in our context, only the parameters of the rules activated by $x(k)$ need to be optimized so what we acquire is partial and dynamic optimization strategy. Hence, we choose the sequential linear programming (SLP) algorithm to realize such strategy because of its fast processing speed, low computational complexity [19]. The specific process is settled as follows:

Step 1: Linearization of the objective function.

According to the above optimization model of the BRB-PID control system, the first-order derivation of the objective function $\xi(P(k))$ about $P(k)$ needs to be calculated and then the first-order Taylor expansion of $\xi(P(k))$ can be obtained as follows

$$\xi(P(k)) = \xi(P_0(k)) + \xi'(P_0(k))(P(k) - P_0(k)), \tag{12}$$

where $P_0(k)$ represents a given initial point. Thus, the nonlinear optimization problem $\min_P \xi(P(k))$ is converted into such a linear programming problem $\min_P \xi'(P_0(k))(P(k)-P_0(k))$.

Step 2: Determination of move limits.

The proper move limits are critical for the successful implementation of SLP. Here, the upper bounds $UB(P(k))$ of adjustable parameters can be acquired as follows:

$$UB(\beta_{n,r}^T) = 1 \,,\; n = 1,..., N; r = 1,..., L^a; T = 1, 2, 3 \tag{13a}$$

$$UB(A_{i,j}^T) = A_{i,J_i} \,,\; j = 1,..., J_i; T = 1, 2, 3 \,. \tag{13b}$$

Then, the initial move limits are set to be 10% of the above upper bounds.

Step 3: Acquisition of the optimal solution using linear programming.

After above two steps, the nonlinear objective func-

Table 2
Pseudo-code of the SLP algorithm

- For k=1:H; %% H denotes Sampling number of control system
  - Determine $P(k)$ about all activated rules in BRB_P,BRB_I,BRB_D;
  - Calculate the first order derivation of $\xi(P(k))$ with respect to $P(k)$ and linearize $\xi(P(k))$ by Taylor expansion;
  - Set up the move limits of $P(k)$ for linear search;
  - While any stopping criterion is satisfied, the SLP iteration process will be stopped;
  - Obtain the optimal solution $\hat{P}_o(k)$ using linear programming;
  - k=k+1;
- End For;

tion $\xi(P(k))$ can be linearized at a given initial point $P_0(k)$, a search space can be established around $P_0(k)$ using its initial move limits. Therefore, linear programming technology (such as Interior-point method) can be adopted for this search process. If the intersection between the established search space and the linearized feasible space is not empty, then the optimal solution of the linearized programming problem will be searched [11]. Otherwise, the move limits need to be increased for expanding the search space until the intersection is not empty. The obtained optimal solution is subsequently used as a new basic point to re-linearize $\xi(P(k))$. This process is repeated recursively until some stopping criterion is satisfied.

Step 4: Stopping criteria.

The SLP iteration process will be stopped if a) the move limits of all adjustable parameters have been reduced to be significantly small, or b) the values of both parameters and objective function in two successive iterations do not markedly change [23]. The pseudo-code of the SLP algorithm is described in Table 2. After the SLP algorithm, one can obtain optimal $\hat{y}_o(k)$, namely the actual system output $y_{out}(k)$, which can be used to recursively predict next $y_{out}(k+1)$ and $e(k+1)$. The detail of the iterative procedure is shown in the following experiments.

4. Experiments

In this section, we conduct two experiments on the excited DC motor system with the proposed adaptive BRB-PID controller. The first experiment mainly shows the precision of the BRB-PID controller. The second one emphasizes the robustness of the BRB-PID controller when the system inputs are disturbed. In both experiments, BRB-PID controller is compared with ANN-PID controller to demonstrate its superiority in adaptivity by experimental data analysis.

4.1. Excited DC Motor System Model

The input and output of controlled object are armature voltage (u) and the speed (y_{out}) of motor, respectively. In the case of no load, the transfer function of the controlled object is [6]

$$G(s) = \frac{y_{out}(s)}{u(s)} = \frac{K_u}{T_a T_m s^2 + T_m s + 1}, \quad (14)$$

where the gain factor of $G(s)$ is K_u=1/C_e, electromagnetic time constant T_a=L_a/R_a, L_a and R_a denote the armature inductance and resistance, respectively. The motor time constant T_m=JR_a/C_eC_m, J is the total moment of inertia in the motor shaft corresponding to rotational part, C_e and C_m denote the potential and torque constants, respectively. It is known that the initial rated armature voltage of the motor C_H=220V, the rated armature current I_H=55mA, R_a=9.2Ω, J=2.4N·m·s^2, T_a=0.0017s and C_e=0.192V·s/rad. The specific transfer function is

$$G(s) = \frac{5.2083}{0.000804s^2 + 1.0473s + 1}. \quad (15)$$

As a result, the transfer function $G(s)$ is discretized as

$$\begin{aligned} y_{out}(k) = {} & 0.9811 * y_{out}(k-1) - 4.8490e^{-12} * y_{out}(k-2) \\ & + 0.0948 * u(k-1) + 0.0038 * u(k-2) \end{aligned}. \quad (16)$$

4.2. Simulation Experiment and Comparative Analyses

Experiment 1: For the closed-loop system shown in Fig.1, the transfer function of controlled object is given in (15). Here, the excitation signal of the BRB-PID controller is set as $r_{in}(k)=sin(\varpi kt_s)$ to generate $y_{out}(k)$.

Here, the classical sine function ($r_{in}(k)=sin(\varpi kt_s)$, k=1,2,...H, sampling period t_s=0.02s, H=400) is adopted as system input. According to the control experiences, the initial forms of BRB_P, BRB_I, BRB_D are constructed respectively with model parameters as shown in Table 1. In this experiment, the three BRBs have the same inputs $X=[r_{in},y_{out},e]$, but different outputs K_P, K_I and K_D, respectively. The initial values of the reference values of r_{in}, y_{out}, e are shown in Tables 3-8, respectively. After Tables 3-8, Table 9 gives the initial rules in BRB_P as an example. In the same way, BRB_I and BRB_D can be constructed. At step k, $X(k)=[r_{in}(k),y_{out}(k),e(k)]$ is substituted into BRB_P,BRB_I,BRB_D to calculate the outputs $K_P(k)$, $K_I(k)$ and $K_D(k)$, respectively according to the reasoning procedure in Section 2.2.2. At the same time, the SLP in Section 3 is performed to optimize the adjustable parameters $P(k)=\{ \beta_{n,r}^{T} | n=1,2,...,N; r=1,2,...,L^a; T=1,2,3\}$ such that the optimal values of $K_P(k)$, $K_I(k)$ and $K_D(k)$ and the corresponding $\hat{y}_o(k)$, namely $y_{out}(k)$ can be obtained. Note that, in this experiment, only partial parameters are selected to be optimized because it is enough for acquiring ideal control performances. Here, the BRB-PID controller is compared with the ANN-PID controller using general BP network proposed in [25]. Here, the inputs and outputs of BP network are, respectively, $X=[r_{in},y_{out},e]$ and K_P, K_I , K_D. Fig.2 gives the comparative results of the two methods to show that $y_{out}(k)$ tracks $r_{in}(k)$ in one experiment and Fig.3 gives the corresponding tracking errors.

Here, we give an iterative optimization procedure (k=4, 5) of the motor control and the related parameter optimization analysis for example.

When k=4, $X(4)$=[0.2478,0.0678,0.1803] is substituted into BRB_P, $X(4)$ activates 8 rules "R_{23}, R_{24}, R_{27}, R_{28}, R_{39}, R_{40}, R_{43}, R_{44}" as listed in Table 10. It means that only local parameters ($P(k=4)=\{\beta_{ir}|$ i=1,2,3; r=23,24,27,28,39,40,43, 44}) need to be optimized using SLP algorithm as listed in Table 11. When k=5, $X(5)$=[0.3090,0.1151,0.1939] is substituted into BRB_P, $X(5)$ activates 8 rules "R_{39}, R_{40}, R_{43}, R_{44}, R_{55}, R_{56}, R_{59}, R_{60}"as listed in Table 12. It can be seen that R_{39}, R_{40}, R_{43}, R_{44} are re-activated, in this case, the initial values of $\beta_{ir}|_{r=39,40,43,44}$ at step k=5 are the optimized values obtained at step k=4 as shown in Table 11. Certainly, such an iterative optimization process also is suitable for BRB_I and BRB_D, respectively. As a result, the different parameters in BRB_P will be optimized at each step using the SLP algorithm according to different activated rules. Table 14 shows the corresponding pseudo-code of BRB-PID iteration for motor control. Finally, Table 10 (Table 12) shows the initial (optimized) activated rules in BRB_P at k=4. Table 11 (Table 13) shows the initial (optimized) activated rules in BRB_P at k=5.

Table 3

The semantic values and reference values of r_{in} for BRB_P, BRB_I, BRB_D

Input1(r_{in})	Semantic Value	S($A_{1,1}$)	NS($A_{1,2}$)	PL($A_{1,3}$)	L($A_{1,4}$)
	Reference Value	-1.01	-0.25	0.25	1.01

Table 4

The semantic values and reference values of y_{out} for BRB_P, BRB_I, BRB_D

Input2(y_{out})	Semantic Value	S($A_{2,1}$)	NS($A_{2,2}$)	PL($A_{2,3}$)	L($A_{2,4}$)
	Reference Value	-1.20	-0.25	0.5	1.15

Table 5

The semantic values and reference values of e for BRB_P, BRB_I, BRB_D

Input3(e)	Semantic Value	S($A_{3,1}$)	NS($A_{3,2}$)	PL(($A_{3,3}$)	L(($A_{3,4}$)
	Reference Value	-0.16	-0.07	0.06	0.20

Table 6

The semantic values and reference values of K_P for BRB_P

Output1(K_P)	**Semantic Value**	S($D_{1,1}$)	NS($D_{1,2}$)	PL($D_{1,3}$)	L($D_{1,4}$)
	Reference Value	0.60	0.62	0.64	0.68

Table 7

The semantic values and reference values of K_I for BRB_I

Output1(K_I)	**Semantic Value**	S($D_{2,1}$)	NS($D_{2,2}$)	PL($D_{2,3}$)	L($D_{2,4}$)
	Reference Value	0.62	0.63	0.66	0.70

Table 8

The semantic values and reference values of K_D for BRB_D

Output1(K_D)	**Semantic Value**	S($D_{3,1}$)	NS($D_{3,2}$)	PL($D_{3,3}$)	L($D_{3,4}$)
	Reference Value	0.56	0.62	0.68	0.78

Table Note: The emantic values (S, NS, PM and M) in tables 3-8 denote “small”, “negative small”, “positive large” and “large”, respectively.

Table 9

The initial rules in BRB_P

No.	**r_{in} AND y_{out} AND e**	**Belief structure of K_P**			
		$\beta^1_{1,1}$	$\beta^1_{1,2}$	$\beta^1_{1,3}$	$\beta^1_{1,4}$
1	S AND S AND S	0	0.5130	0.4870	0
2	S AND S AND NS	0	0.3983	0.6017	0
3	S AND S AND PL	0	0.3856	0.6144	0
4	S AND S AND L	0	0.3856	0.6144	0
⋮	⋮	⋮	⋮	⋮	⋮
31	NS AND L AND PL	0	0	0.3715	0.6285
32	NS AND L AND L	0	0	0.3662	0.6338
33	PL AND S AND S	0	0	0.8896	0.1104
34	PL AND S AND NS	0	0	0.8896	0.1104
⋮	⋮	⋮	⋮	⋮	⋮
61	L AND L AND S	0	0	0.0699	0.9301
62	L AND L AND NS	0	0	0.0755	0.9245
63	L AND L AND PL	0	0	0.0755	0.9245
64	L AND L AND L	0	0	0.0755	0.9245

Table Note: BRB_I and BRB_D have the same structure as BRB_P.

Table 10
The initial activated rules in BRB_P at k=4

No.	r_{in} AND y_{out} AND e	Belief structure of activated rules in BRB_P			
		$\beta^1_{1,1}$	$\beta^1_{1,2}$	$\beta^1_{1,3}$	$\beta^1_{1,4}$
23	NS AND NS AND PL	0.0081	0.0189	0.7528	0.2202
24	NS AND NS AND L	0.0089	0.0102	0.7623	0.2186
27	NS AND PL AND PL	0.0015	0.0018	0.4743	0.5224
28	NS AND PL AND L	0.0097	0.0101	0.4598	0.5204
39	PL AND NS AND PL	0.0096	0.0102	0.4625	0.5177
40	PL AND NS AND L	0.0096	0.0102	0.4625	0.5177
43	PL AND PL AND PL	0.0105	0.0114	0.4456	0.5325
44	PL AND PL AND PL	0.0075	0.0105	0.4397	0.5423

Table 11
The optimized activated rules in BRB_P at k=4

No.	r_{in} AND y_{out} AND e	Belief structure of activated rules in BRB_P			
		$\beta^1_{1,1}$	$\beta^1_{1,2}$	$\beta^1_{1,3}$	$\beta^1_{1,4}$
23	NS AND NS AND PL	0.0076	0.0185	0.7025	0.2714
24	NS AND NS AND L	0.0083	0.0106	0.7223	0.2588
27	NS AND PL AND PL	0.0101	0.0104	0.3751	0.6044
28	NS AND PL AND L	0.0093	0.0091	0.3688	0.6128
39	PL AND NS AND PL	0.0087	0.0098	0.3601	0.6214
40	PL AND NS AND L	0.0087	0.0098	0.3601	0.6214
43	PL AND PL AND PL	0.0095	0.0113	0.3653	0.6139
44	PL AND PL AND PL	0.0104	0.0119	0.3596	0.6181

Table 12
The initial activated rules in BRB_P at k=5

No.	r_{in} AND y_{out} AND e	Belief structure of activated rules in BRB_P			
		$\beta^1_{1,1}$	$\beta^1_{1,2}$	$\beta^1_{1,3}$	$\beta^1_{1,4}$
39	PL AND NS AND PL	0.0087	0.0098	0.3601	0.6214
40	PL AND NS AND L	0.0087	0.0098	0.3601	0.6214
43	PL AND PL AND PL	0.0095	0.0113	0.3653	0.6139
44	PL AND PL AND PL	0.0104	0.0119	0.3596	0.6181
55	L AND NS AND PL	0	0	0.3662	0.6338
56	L AND NS AND L	0	0	0.3662	0.6338
59	L AND PL AND PL	0	0	0.2146	0.7854
60	L AND PL AND PL	0	0	0.2146	0.7854

Table 13

The optimized activated rules in BRB_P at k=5

No.	r_{in} AND y_{out} AND e	Belief structure of activated rules in BRB_P			
		$\beta^1_{1,1}$	$\beta^1_{1,2}$	$\beta^1_{1,3}$	$\beta^1_{1,4}$
39	PL AND NS AND PL	0.0089	0.0093	0.3517	0.6301
40	PL AND NS AND L	0.0089	0.0093	0.3517	0.6301
43	PL AND PL AND PL	0.0087	0.0096	0.3512	0.6305
44	PL AND PL AND PL	0.0102	0.0103	0.3428	0.6367
55	L AND NS AND PL	0.0105	0.0103	0.3796	0.5996
56	L AND NS AND L	0.0105	0.0103	0.3796	0.5996
59	L AND PL AND PL	0.0088	0.0096	0.2322	0.7494
60	L AND PL AND PL	0.0088	0.0096	0.2322	0.7494

Table 14

Pseudo-code of BRB-PID-based motor control

initial value : $u(0)= u(-1)=0$; $y(0)= y(-1)=0$;

- when k=1;
  - $r_{in}(1)=x$, $y_{out}(1)=0$, $e(1)= r_{in}(1)- y_{out}(1)=x$; %% x is the value of r_{in} at k=1;
  - $X(1)=[r_{in}(1),y_{out}(1),e(1)]$ is substituted into BRB_P,BRB_I,BRB_D to calculate the outputs $K_P(1)$, $K_I(1)$ and $K_D(1)$, respectivley;
  - $K_P(1)$, $K_I(1)$ and $K_D(1)$ are substituted into the incremental PID in (1) to calculate $u(1)$;
  - use SLP algorithm in Table 2 to minimize $\xi(P(1))$ by adjusting parameter set $P(1)$, then obtain the optimal $\hat{P}_o(1)$ and the corresponding optimal $\hat{K}_p(1), K_I(1), K_D(1)$ and $\hat{u}_o(1)$;
  - obtain $y_{out}(2)$ and $\hat{y}_o(1)$ by using (16);

    $$y_{out}(k) = 0.9811 * y_{out}(k-1) - 4.8490e^{-12} * y_{out}(k-2) + 0.0948 * u(k-1) + 0.0038 * u(k-2)$$
- For k=2 to H; %% H denotes sampling number of control system;
  - $X(k)=[r_{in}(k),y_{out}(k),e(k)]$ is substituted into BRB_P,BRB_I,BRB_D to calculate the outputs $K_P(k)$, $K_I(k)$ and $K_D(k)$;
  - $K_P(k)$, $K_I(k)$ and $K_D(k)$ are substituted into the incremental PID in (1) to calculate $u(k)$;
  - use SLP algorithm to minimize $\xi(P(k))$ by adjusting $P(k)$, then obtain the optimal $\hat{P}_o(k)$ and the corresponding optimal $\hat{K}_p(k), K_I(k), K_D(k)$ and $\hat{u}_o(k)$;
  - obtain $y_{out}(k+1)$ and $\hat{y}_o(k+1)$ by using (16);

    $$\hat{y}_o(k) = 0.9811 * y_o(k-1) - 4.8490e^{-12} * y_o(k-2) + 0.0948 * u_o(k-1) + 0.0038 * u_o(k-2)$$
  - $k=k+1$;
- End For;

Figure 2

Control results for BRB-PID and BP-PID models

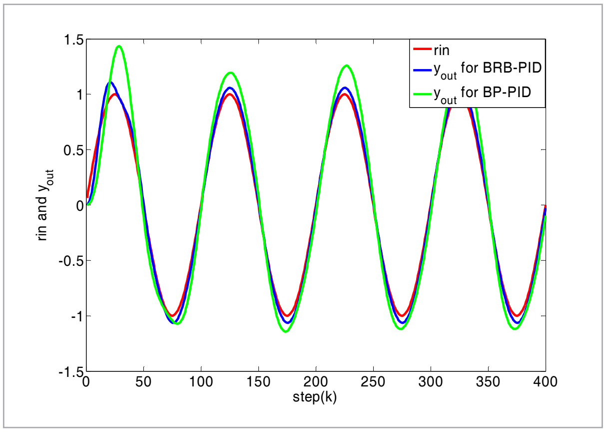
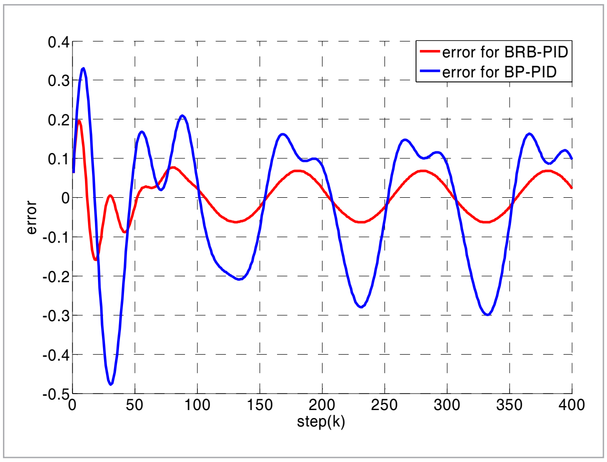


Figure 3

Closed-loop system error based on BRB-PID/BP-PID model



Furthermore, such an experiment is done 100 times, in which, the initial values of BP-PID parameters are given randomly. Finally, the mean values of tracking mean square error (MSE) for BRB-PID and BP-PID are 0.0031 and 0.0057, respectively. From these results in single experiment as shown in Figs. 2-3 and statistical experiments, it can be seen that the former are superior to the latter in control accuracy. This is because the BRB models as a nonlinear approximator, have more optimizable parameters (total 96 parameters in the activated rules of BRB_P,BRB_I,BRB_D) than the BP model has (35 parameters in BP network) for modeling the complex relationship between r_{in} ,y_{out}, e and K_P, K_I , K_D. It has been proved that the BRB can approximate any nonlinear function on a compact set when the number of parameters increase [5], but this conclusion is hardly suitable for BP. Besides, the physical meanings of the parameters in BRB are easier to understand and the users can adjust the numbers and values of these parameters to obtain a desirable approximator. However, for BP network, it is relatively difficult to distinctly interpret importance and physical meaning of each parameter.

In this experiment, at each step, the BRB model needs to optimize 96 parameters, the BP model needs to optimize 35 parameters, although the BRB model is more complex than the BP model, because of the usage of SLP algorithm, the average single step optimization time of BRB is 0.015s (0.008s for BP). Obviously, BRB-PID controller is applicable for general time-varying models. Certainly, with the rapid development of data processing and storage capabilities, computer hardware will make the optimization time less of an issue.

On the other hand, although the parameters in BP network can be optimized at step k whose initial values are set as the optimal values at k-1, how to determine the initial values of these parameters at initial step k=1 is a challenge because there is no good experience for reference. Hence, when the initial values are generated randomly, in some cases, the results of BP-PID controller are hardly satisfactory. For example, Figs.4-5 show a typical case in which y_{out} cannot approximate to r_{in}. On the contrary, y_{out} of BRB-PID always maintains desirable performance.

Figure 4

Closed-loop output tracking system input based on BP-PID model

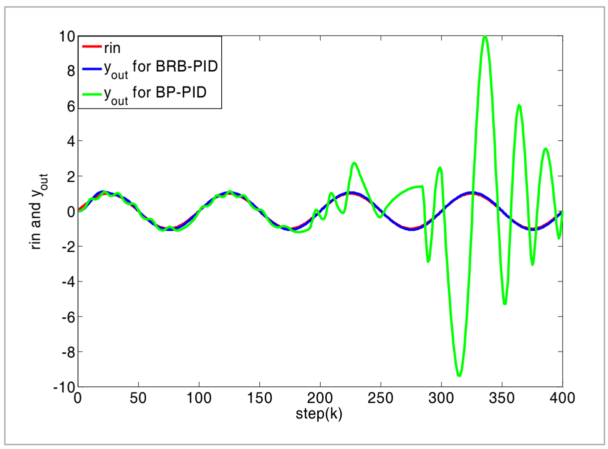


Figure 5

Closed-loop system error based on BP-PID model

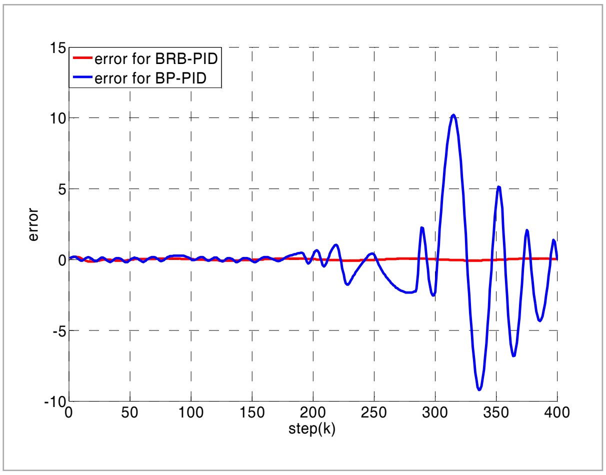
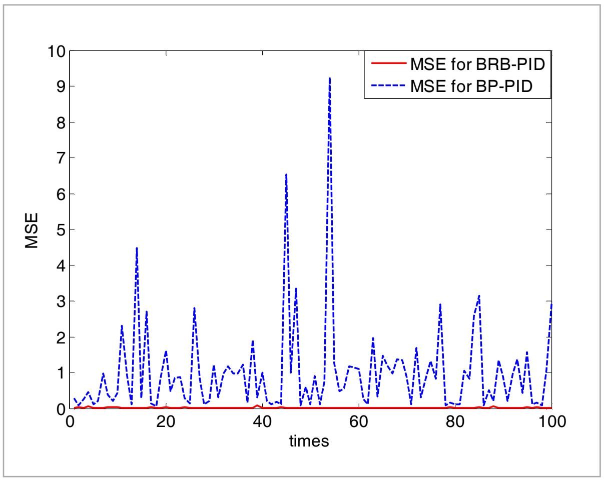


Figure 6

The tracking mean square errors of BRB-PID and BP-PID



Experiment 2: In order to illustrate the robustness of the BRB-PID controller, we conducted this experiment in which the system inputs are disturbed, namely $r_{in}(k)=sin(\varpi kt_s)+A*sin(wkt_s)$, k=1,2,...H, sampling interval t_s =0.02s, H=400, the noise variables A~U(0,0.1), w~U(0,10ϖ).

Such an experiment is done 100 times in which the values of A and w are randomly generated according to their own distributions. Fig.6 shows the tracking mean square errors of BRB-PID and BP-PID, respectively, in the 100 times experiments. From these data, we can get that the mean values of tracking MSE for BRB-PID and BP-PID are 0.0191 and 1.0072, respectively. From the above statistical results, it can be concluded that BRB-PID has better adaptability for input interference than BP-PID.

5. Conclusion

In this paper, the idea of knowledge-based belief rule reasoning is introduced into PID controller design. Based on BRB system, a novel self-tuning strategy of PID parameters is presented which is a new kind of adaptive PID control strategy. The main contributions of the paper are as follows:(1) The initial belief rule bases are abstracted from expert's control experiences to model the highly nonlinear relationships between control variables r_{in}, y_{out}, e and PID parameters K_P, K_I, K_D, respectively; (2) In order to reduce the imprecision of expert's knowledge, the online optimization method via SLP is presented to optimize the parameters of BRB system so as to adaptively adjust PID parameters by the optimized BRB system in real time;(3) The typical control simulation experiments of DC motor are implemented to illustrate the advantages of BRB-PID controller over the widely used adaptive ANN-PID controller.

Acknowledgements

This work was supported by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1709215), the NSFC (No.61433001, 61733009,61573076,61573275), the Zhejiang Province "One Belt and One Road" technology cooperation project (No.2018C04020), the University Students' Scientific and Technological Innovation Activity Plan of Zhejiang Province (Xin-Miao Talent Plan.No.2017R407064), the Zhejiang Province Key R&D project (No. 2018C01031) of Hangzhou Yan Shi Technology Co., Ltd.

References

1. Abudahab, K., Xu, D. L., Chen, Y. W. A New Belief Rule Base Knowledge Representation Scheme and Inference Methodology Using the evidential Reasoning Rule for Evidence Combination. Expert Systems with Applications, 2016, 51, 218-230. https://doi.org/10.1016/j.eswa.2015.12.013
2. Alexandrov, A., Palenov, M. Self-Tuning PID-I Controller with a New Algorithm of Tuning of Test Signal. IFAC

Proceedings Volumes, 2013, 46(9), 1798-1803. https://doi.org/10.3182/20130619-3-RU-3018.00344

3. Chang, L., Zhou, Y., Jiang, J., Li, M., Zhang, X. Structure Learning for Belief Rule Base Expert System: A Comparative Study. Knowledge-Based Systems, 2013, 39(2), 159-172. https://doi.org/10.1016/ j.knosys.2012.10.016

4. Chang, L., Zhou, Z. J., You, Y., Yang, L., Zhou, Z. Belief Rule Based Expert System for Classification Problems with New Rule Activation and Weight Calculation Procedures. Information Sciences, 2016, 336, 75-91. https://doi.org/10.1016/ j.ins.2015.12.009

5. Chen, Y. W., Yang, J. B., Xu, D. L., Yang, S. L. On the Inference and Approximation Properties of Belief Rule Based Systems. Information Sciences, 2013, 234, 121-135. https://doi.org/10.1016/j.ins.2013.01.022

6. Gao, G., Yu, W. Automatic Control Theory. South China University of Technology Press, Guangzhou, 75-91, 2009.

7. Gautam, D., Cheolkeun, H. Control of a Quadrotor Using a Smart Self-Tuning Fuzzy PID Controller. International Journal of Advanced Robotic Systems, 2013, 10. https://doi.org/10.5772/56911

8. Huang, Y. R., Qu, L. G. PID Controller Parameter Tuning and Implementation. Science Press, Beijing, 2010.

9. Jiménez, T., Merayo, N., Andrés, A., Durán, R. J., Aguado, J. C., de Miguel, I., Fernández, P., Lorenzo, R. M., Abril, E. J. An Auto-Tuning PID Control System Based on Genetic Algorithms to Provide Delay Guarantees in Passive Optical Networks. Expert Systems with Applications, 2015, 42(23), 9211-9220. https://doi.org/10.1016/j.eswa.2015.07.078

10. Karasakal, O., Guzelkaya, M., Eksin, I., Yesil, E., Kumbasar, T. Online Tuning of Fuzzy PID Controllers via Rule Weighing Based on Normalized Acceleration. Engineering Applications of Artificial Intelligence, 2013, 26(1), 184-197. https://doi.org/10.1016/j.engappai.2012.06.005

11. Lamberti, L., Pappalettere, C. Move Limits Definition in Structural Optimization with Sequential Linear Programming. Part II: Numerical Examples. Computers & Structures, 2003, 81(4), 215-238. https://doi.org/10.1016/S0045-7949(02)00443-1

12. Li, G. L., Zhou, Z. J., Hu, C. H., Chang, L. L., Zhou, Z. G., Zhao, J. F. A New Safety Assessment Model for Complex System Based on the Conditional Generalized Minimum Variance and the Belief Rule Base. Safety Science, 2017, 93, 108-120. https://doi.org/10.1016/j.ssci.2016.11.011

13. Li, W. Design of PID Controller Based on an Expert System. International Journal of Computer, Consumer and Control (IJ3C), 2014, 3(1), 31-40.

14. Rivera, D. E., Morari, M., Skogestad, S. Internal Model Control: PID Controller Design. Industrial & Engineering Chemistry Process Design & Development, 1986, 25(1), 252-265. https://doi.org/10.1021/i200032a041

15. Rodrigo, H. A., Govinda, G. V. L., Tomás, S. J., Alfonso, G. E., Fernando, F. N. Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. Sensors, 2016, 16(9). https://doi.org/10.3390/s16091429 https://doi.org/10.3390/s16091429

16. Saxena, S., Hote, Y. V. Simple Approach to Design PID Controller via Internal Model Control. Arabian Journal for Science & Engineering, 2016, 41(9), 3473-3489. https://doi.org/10.1007/s13369-016-2027-4

17. Skogestad, S. Simple Analytic Rules for Model Reduction and PID Controller Tuning. Journal of Process Control, 2003, 13(4), 291-309. https://doi.org/10.1016/S0959-1524(02)00062-8

18. Tzafestas, S., Papanikolopoulos, N. P. Incremental Fuzzy Expert PID Control. IEEE Transactions on Industrial Electronics, 1990, 37(5), 365-371. https://doi.org/10.1109/41.103431

19. Willis, M. J., von Stosch, M. L0-Constrained Regression Using Mixed Integer Linear Programming. Chemometrics & Intelligent Laboratory Systems, 2017, 165, 29-37. https://doi.org/10.1016/ j.chemolab.2016.12.016

20. Xia, C. L., Mei, X., Chen, Z. Adaptive PID Control and On-line Identification for Switched Reluctance Motors Based on BP Neural Network. IEEE International Conference Mechatronics and Automation, 2005, 4, 1918-1922. Xu, X. B., Wen, C. L., Sun, X.Y., Ji, Y. D. Evidence Fusion and Decision Making Methods in Equipment Fault Diagnosis. Science Press, Beijing, 2017.

21. Xu, X. B., Liu, Z., Chen, Y. W., Xu, D. L., Wen, C. L. Circuit Tolerance Design Using Belief Rule Base. Mathematical Problems in Engineering, 2015, 3, 1-12. http://dx.doi.org/10.1155/2015/ 908027

22. Xu, X. B., Zheng, J., Yang, J.B., Xu, D. L., Chen, Y. W. Data Classification Using Evidence Reasoning Rule. Knowledge-Based Systems, 2017, 116, 144-151. https://doi.org/10.1016/j.knosys. 2016.11.001

23. Yang, J.B., Xu, D.L. Evidential Reasoning Rule for Evidence Combination. Artificial Intelligence, 2013, 205, 1-29. https://doi.org/10.1016/j.artint.2013.09.003

24. Zhang, Z. X. Neural Network Control and MATLAB Simulation. Harbin Institute of Technology Press, Harbin, 2011.