

Diagnosis of incipient faults in power transformers using CMAC neural network approach

Chin-Pao Hung*, Mang-Hui Wang

Department of Electrical Engineering, National Chin-Yi Institute of Technology, 35, 215 Lane, Sec. 1, Chung Shan Road, Taiping, Taichung, Taichung, Taiwan, ROC

Received 2 September 2003; received in revised form 26 November 2003; accepted 26 January 2004

Available online 12 May 2004

Abstract

Dissolved gas analysis (DGA) is one of the most useful techniques to detect the incipient faults of power transformer. However, the identification of the faulted location by the traditional method is not always an easy task due to the variability of gas data and operational natures. In this paper, a novel cerebellar model articulation controller (CMAC) neural network (NN) method is presented for the fault diagnosis of power transformers. By introducing the IEC standard 599 to generate the training data, and using the characteristic of self-learning and generalization, like the cerebellum of human being, a CMAC NN fault diagnosis scheme enables a powerful, straightforward, and efficient fault diagnosis. With application of this scheme to published transformers data, the diagnoses demonstrate the new scheme with high accuracy and high noise rejection ability. Moreover, the results also proved the ability of multiple incipient faults detection.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Transformer fault diagnosis; Dissolved gas analysis; Neural network; CMAC

1. Introduction

Power transformers are essential devices in a transmission and distribution system. Failure of a power transformer may cause a break in power supply and loss of profits. Therefore, it is of great importance to detect incipient failures in power transformers as early as possible, so that we can switch them safely and improve the reliability of power systems.

A long in-service transformer is subject to electrical and thermal stresses, which may form byproduct gases to indicate the type of incipient failure. Dissolved gas analysis (DGA) is a common practice in the incipient fault diagnosis of power transformers [1,2], which tests and samples the insulation oil of transformers periodically to obtain the constituent gases in the oil due to breakdown of the insulating materials inside. As study results indicate, corona, overheating and arcing are the three main causes for insulation degradation in a transformer [2–4]. The energy dissipation is least in corona, medium in overheating, and highest in arcing. The fault related gases include hydrogen (H_2), methane

(CH_4), acetylene (C_2H_2), ethane (C_2H_6), carbon monoxide (CO), and carbon dioxide (CO_2).

In the past decade, various fault diagnosis techniques have been proposed that include the conventional key gas method, ratio method [2–5], and recently, the expert systems [6], neural network [7,8,13,14] and fuzzy logic approaches [9–12]. The conventional key gas or ratio method is based on experience in fault diagnosis using DGA data, where may vary from utility to utility due to the heuristic nature of methods and no general mathematical formulation can be utilized. The expert system and fuzzy logic approaches can take human expertise and DGA standards from the fault diagnosis system, and have been successfully applied in this field. However, there are some intrinsic shortcomings, such as the difficulty of acquiring knowledge and maintaining database, so, their effectiveness depends on the completeness and precision of expert expertise. The neural network can directly acquire experience from the training data, and exhibit highly non-linear input–output relationships. This can overcome some of the shortcomings of expert system. However, the amount of training data must be large enough to ensure proper training. Non-training data are easy to cause wrong diagnoses. Moreover, the multi-

* Corresponding author. Tel.: +86-4-23924505; fax: +86-4-23924419.
E-mail address: cbhong@chinyi.nct.edu.tw (C.-P. Hung).

ple faults diagnoses and fault anticipation abilities are still lacking.

In this paper, a novel CMAC NN method is presented for the fault diagnosis of power transformers. Using the characteristic of self-learning and generalization, like the cerebellum of human being, the CMAC NN diagnosis scheme enables a powerful, straightforward, and efficient fault diagnoses. First, we generated the virtual training data based on the IEC standard 599 to replace the large amount actual training data. Second, we developed a CMAC NN diagnosis model and using virtual training data to train the memory weights. Finally, the proposed scheme can be used to diagnose the fault type of power transformers. Compare the diagnosis and actual fault type, the difference is used to refine the memory weights. With application of this scheme to published transformers data, the diagnoses demonstrate the new scheme with high accuracy and high noise rejection ability. Moreover, the results also proved the ability of multiple incipient faults detection.

2. Background on CMAC neural network

Albus proposed a neural model called CMAC, which like the models of human memory, perform a reflexive processing [15]. The CMAC, in a table look-up fashion, produced a vector output in response to a state vector input. Fig. 1 shows a basic configuration of CMAC network [17], where the input states are denoted by $x \in R^n$, and the output is $y \in R^m$. Through a series of mappings, include the quantization, segment addresses coding, virtual addresses concatenation, Hash coding (if needed), and summation the fired memory addresses weights to obtain an output. The mapping processes must satisfy the similar inputs excite the similar memory addresses, i.e. if the input states are close to (similar) in input space will have their corresponding sets

Table 1
IEC gas ratio codes

Ranges of the gas ratio	Codes of different gas ratio		
	C ₂ H ₂ /C ₂ H ₄	CH ₄ /H ₂	C ₂ H ₂ /C ₂ H ₆
<0.1	0	1	0
0.1–1	1	0	0
1–3	1	2	1
>3	2	2	2

of association cells overlap. For example, if x_1 and x_2 are similar (close), x_1 excites the memory addresses a_1, a_2, a_3, a_4 , and x_2 should excite the memory addresses a_2, a_3, a_4, a_5 or a_3, a_4, a_5, a_6 . If two inputs fire up the same memory addresses, we say the similarity of the two inputs is high. Low similarity would excite fewer same memory addresses.

As described earlier, assume the number of fired memory cells is four. Then the summation of the fired memory cells weights will obtain an output. Compare the output with the desired target, the difference can be used to train the CMAC NN. A trained CMAC NN will remember the correct mapping relation for the special input states. Therefore, when the same or similar input states input again, the output will preserve the possible correct output depending on the similarity. As a result, the characteristic of generalization, local reflexive action and self-learning make the CMAC attractive to fault diagnosis system, especially to multiple faults and lack of fully training data system.

3. The configuration of CMAC NN fault diagnosis system

In dissolved gas analysis, the IEC codes have been used widely by the utilities. From IEC standard 599, the codes of different gas ratios and fault classifications according to the gas ratio codes are shown in Tables 1 and 2. IEC codes are

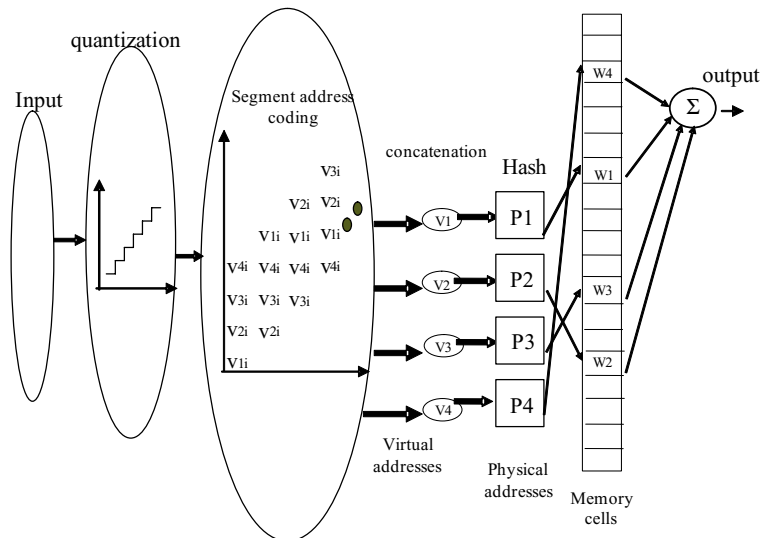


Fig. 1. The schematic of CMAC neural network.

Table 2
Fault types according to the gas ratio codes

Fault no.	Fault type	C ₂ H ₂ /C ₂ H ₄	CH ₄ /H ₂	C ₂ H ₄ /C ₂ H ₆
1	No fault	0	0	0
2	<150 °C thermal fault	0	0	1
3	150–300 °C thermal fault	0	2	0
4	300–700 °C thermal fault	0	2	1
5	>700 °C thermal fault	0	2	2
6	Low energy partial discharges	0	1	0
7	High energy partial discharges	1	1	0
8	Low energy discharges	1 or 2	0	1 or 2
9	High energy discharges	1	0	2

useful for fault diagnosis in transformers, but the number of code combinations is larger than the number of fault types, and “no match” may be indicated in the fault diagnosis. In this section, the CMAC NN fault diagnosis method is proposed for power transformers to solve the no match, noise, and multiple faults problems.

3.1. The development of CMAC NN fault diagnosis system

Fig. 2 shows the configuration of the CMAC NN fault diagnosis system of power transformers. Refer to the IEC standard 599, the gas ratio of C₂H₂/C₂H₄, CH₄/H₂ and C₂H₄/C₂H₆ are used as the input states. The output contains nine parallel memory layers and every memory layer has one output node. Every memory layer remembers one fault type feature, e.g. layer 1 store the features of fault type 1 of Table 2, layer 2 stores the features of fault type 2 of Table 2, etc. Input one group gas ratio data, through a series of mapping, the input data will generate one group fired memory addresses. To sum the excited memory addresses of each layer, output node will obtain one value to express the possibility of fault type *n*. To confirm the fault type the output value will be close to 1. Multiple nodes output 1 represents the multiple fault types exist.

3.2. The training of CMAC NN fault diagnosis system

The proposed scheme using the IEC codes of Tables 1 and 2 to generate the training data. Therefore, the large amount real data are not necessary. For example, the gas ratio code of fault type 2 (C₂H₂/C₂H₄, C₄/H₂, C₂H₄/C₂H₆) = (0, 0, 1), i.e. C₂H₂/C₂H₄ < 0.1, 0.1 ≤ CH₄/H₂ ≤ 1, 1 < C₂H₂/C₂H₆ ≤ 3. Therefore, using the program 1 (designed by MATLAB), the virtual training data can be generated. In program 1 the step value STEP_X determines the resolution of training data. High resolution will cause long training time. The training data then send to the CMAC network (layer 2), through the quantization, fired addresses coding, and sums the fired memory cells weighting to obtain an output. Compare the desired output 1, then the error used to tuning the fired memory weightings. The details will describe as follows:

```

for C2H2_C2H4=0:STEP_1:0.1
  for CH4_H2=0.1:STEP_2:1
    for C2H4_C2H6=1:STEP_3:3
      %quantization, address coding, summation and weight tuning;
    end
  end
end
    
```

Program 1. Training data generation

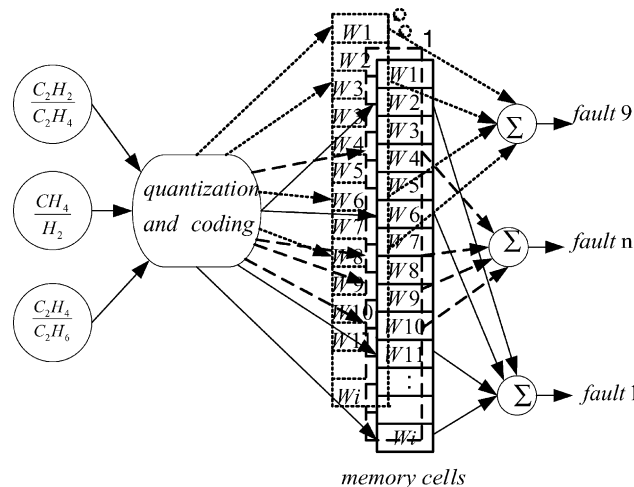


Fig. 2. The configuration of CMAC NN fault diagnosis system of power transformers.

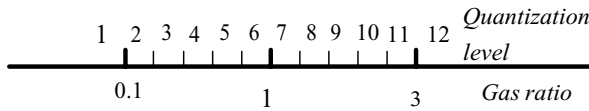


Fig. 3. Quantization mapping.

3.2.1. Quantization mapping

The quantization output can be described as follows [17]

$$q_i = Q(x_i, x_{i,\min}, x_{i,\max}, q_{i,\max}), \quad i = 1, \dots, n \quad (1)$$

where n is the input numbers. The resolution of this quantization depends on the expected maximum and minimum inputs, $x_{i,\max}$ and $x_{i,\min}$, and on the number of quantization levels, q_{\max} . Assuming the maximum quantization level q_{\max} is chosen as 12, Fig. 3, the quantization mapping diagram, shows that the input state between 0.1 and 1, 1 and 3 are divided five quantization levels, respectively. Noted we can change the q_{\max} value depending on the resolution requirements.

3.2.2. Segment address mapping

Table 3 lists the mapping relation of quantization level and the segment address, which the quantization level q_{\max} is 8 and A^* , the number of associated (fired) memory cells is 4. For example, the quantization level 3 will map a group segment addresses $[v_{11}, v_{12}, v_{13}, v_{14}] = [5, 6, 3, 4]$.

3.2.3. Concatenation and output mapping

As described above, each input state produces A^* segment addresses. The concatenation unit then concatenated these segment addresses as a virtual address. The concatenation equation can be expressed as

$$V_j = \text{concat}(v_{j1}, v_{j2}, \dots, v_{jm}), \quad j = 1, \dots, A^* \quad (2)$$

Assuming the gas ratio quantization levels of (C_2H_2/C_2H_4 , CH_4/H_2 , C_2H_2/C_2H_6) are equal to (3, 6, 8), then the segment addresses generated by C_2H_2/C_2H_4 are $[v_{11}, v_{12}, v_{13}, v_{14}] = [5, 6, 3, 4]$, by CH_4/H_2 are $[v_{21}, v_{22}, v_{23}, v_{24}] = [9, 6, 7, 8]$

Table 3
Quantization level and segment address mapping segment address

Segment address	Quantization level							
	1	2	3	4	5	6	7	8
11								v_3
10							v_2	v_2
9						v_1	v_1	v_1
8					v_4	v_4	v_4	v_4
7				v_3	v_3	v_3	v_3	
6			v_2	v_2	v_2	v_2		
5		v_1	v_1	v_1	v_1			
4	v_4	v_4	v_4	v_4				
3	v_3		v_3					
2	v_2	v_2						
1	v_1							

and by C_2H_2/C_2H_6 are $[v_{31}, v_{32}, v_{33}, v_{34}] = [9, 10, 11, 8]$. Then the $\text{concat}(\cdot)$ operation can be calculated as follows:

$$\begin{aligned} V_1 &= \text{concat}[v_{11}, v_{21}, v_{31}] = \text{concat}[5, 9, 9] \\ &= 010110011001B \\ V_2 &= \text{concat}[v_{12}, v_{22}, v_{32}] = \text{concat}[6, 6, 10] \\ &= 011001101010B \\ V_3 &= \text{concat}[v_{13}, v_{23}, v_{33}] = \text{concat}[3, 7, 11] \\ &= 001101111011B \\ V_4 &= \text{concat}[v_{14}, v_{24}, v_{34}] = \text{concat}[4, 8, 8] \\ &= 010010001000B \end{aligned}$$

Assume

$$\text{bitn} = \text{ceil}(\log_2(q_{j,\max} + A^*)) \quad (3)$$

where bitn is the minimum bit numbers to decode the segment address and $\text{ceil}(x)$ a function rounds the elements of x to the nearest integers towards infinity. Then, the general form of V_j can be expressed as follows

$$V_j = \sum_{i=1}^n v_{ji} 2^{\text{bitn}(i-1)}, \quad j = 1, \dots, A^* \quad (4)$$

Using Eq. (5) to sum the weights located at these addresses will obtain an output value

$$y = \sum_{j=1}^{A^*} w v_j \quad (5)$$

3.2.4. Hash coding

Hash coding performs a many-to-one uniform random mapping to generate a physical memory address. As described in [17], it compresses the huge virtual address space into a compact amount of memory and minimizes the probability of physical address collision (different inputs fire up the same association address). In this paper, we do not consider the Hash coding because of the memory size is acceptable.

3.2.5. Update the weighting

During training, the fired addresses are updated using the following steepest-descent update rule [16,17]

$$w_{v_i}^{\text{new}} \leftarrow w_{v_i}^{\text{old}} + \beta \frac{y_d - y}{A^*}, \quad i = 1, 2, \dots, A^* \quad (6)$$

In this equation, y_d is the desired output, y the actual output, and $0 < \beta \leq 1$ the learning gain.

3.2.6. Noise rejection

The quantization and segment mappings give the CMAC the ability to generalize (produce similar outputs in response to similar inputs). Continuous variations in input values translate into discrete variations in input quantization levels. As described in Section 3.2.3, if the quantization level of C_2H_2/C_2H_4 input change by 1 (3 \rightarrow 4), then the segment address mapping $[v_{11}, v_{12}, v_{13}, v_{14}]$ from [5, 6, 3, 4] change to

[5, 6, 7, 4]. All virtual address segments remain the same except v_{13} , which shifts from 3 to 7. Consequently, outputs associated with neighboring input quantization levels will have three of four virtual addresses in common because only one address will have changed, V_3 shifts from $\text{concat}[3, 7, 11] = 001101111011B$ to $\text{concat}[7, 7, 11] = 001101111011B$ (assuming the other two input levels remain constant). Therefore, even though the noise added to the input, the CMAC outputs still can preserve most of the correct features. Note that for a given input quantization mapping, an increase in A^* results in an increase in the amount of shared weights between neighboring input/output pairs. This will increase the generalization ability and improve the noise rejection.

3.2.7. Convergence

For a supervised learning system, the convergence is confirmed [16]. In this paper, we assume the memory size is large enough and without using the Hash coding. The collision will not happen and the convergence is guaranteed.

3.2.8. Learning performance evaluation

Assuming the i th node ($i = 1, \dots, 9$) of Fig. 2 outputs 1 represents system with fault type i . The training data number n_g generated by program 1 can be calculated as following equation

$$n_g = \text{fix} \left(\frac{0.1 - 1}{\text{STEP}_1} \right) \text{fix} \left(\frac{1 - 0.1}{\text{STEP}_2} \right) \text{fix} \left(\frac{3 - 1}{\text{STEP}_3} \right) \quad (7)$$

where $\text{fix}(x)$ function rounds the elements of x to the nearest integers towards zero.

Let

$$E_i = \sum_{j=1}^{n_g} (y_{ij} - 1)^2, \quad i = 1, \dots, 9 \quad (8)$$

where i subscript represents the i th layer. Then, we can stop the training phase when $E_i < \varepsilon$, ε is a positive small number.

3.3. Diagnosis algorithm

As described earlier, the diagnosis algorithm summarized as follows. Fig. 4 shows the diagnosis flowchart.

3.3.1. Off-line training phase

- Step 1. Build the architecture of CMAC NN fault diagnosis system, including three input states, nine layers memory and nine output nodes.
- Step 2. Specify the quantization level q_{\max} , learning gain β , and the amount of fired memory cells A^* .
- Step 3. Generate the virtual training data via IEC code of Tables 1 and 2.
- Step 4. Quantization, fired addresses coding, and sums the fired memory cells weights to obtain an output.
- Step 5. Update the fires memory cells weights using Eq. (6).
- Step 6. Does the training data finish? No, go to step 3. Yes, next step.
- Step 7. Learning performance evaluation. If $E_i < \varepsilon$, stop training and save the memory weights. Otherwise, go to step 3.

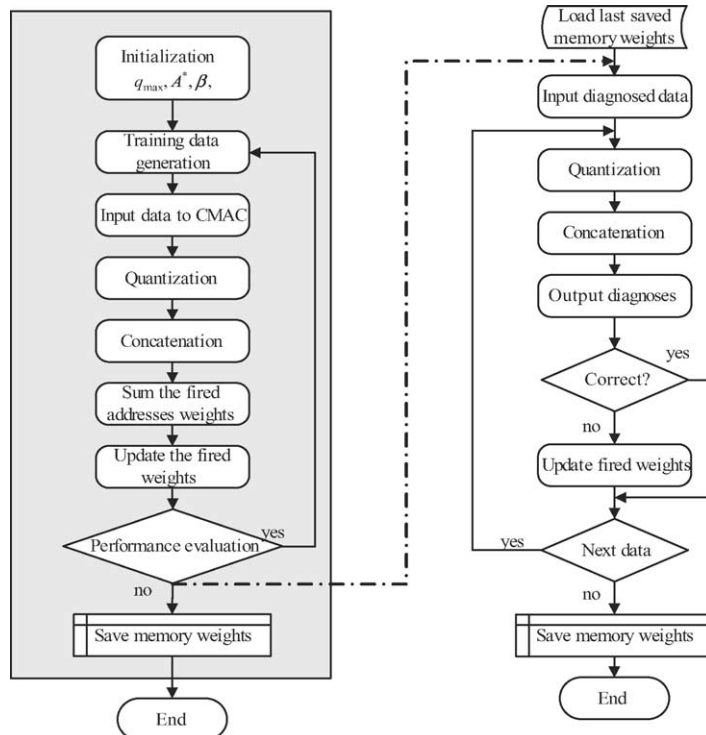


Fig. 4. Flowchart of fault diagnosis system.

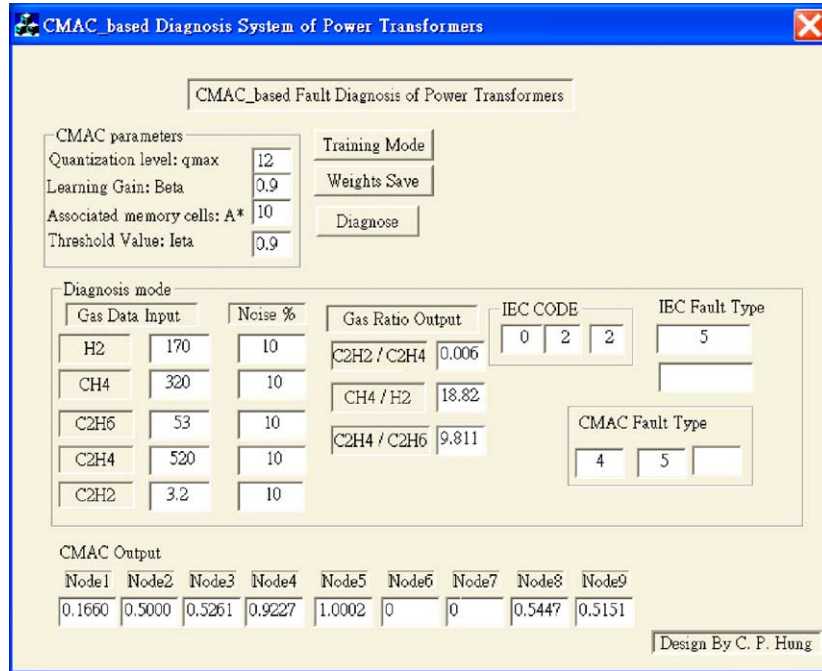


Fig. 5. Diagnosis program interface.

Steps 1–7 is off-line mode. The training time, depends on the data resolution, q_{max} , A^* , and the selection of gas ratio range, maybe shorter just few seconds or longer more than few hours (PENTIUM iii500, using MATLAB programming). Fortunately, the off-line mode just only needs to run one time. Generally, long training time will obtain better and more exact weights, just like the learning mode of human being.

3.3.2. On-line mode

Finish the off-line training mode, then the diagnosis system can be used to diagnose the fault type of transformers.

Step 8. Load the last saved memory weights and specify the threshold value (e.g. $\eta = 0.9$).

Step 9. Input the gas ratio data that to be diagnosed.

Table 4
Tested gas data of transformer and diagnoses by different method

No.	H ₂	CH ₄	C ₂ H ₆	C ₂ H ₄	C ₂ H ₂	AFC	IEC	CMC
1	14.7	3.7	10.5	2.7	0.2	1	1	1, 6
2	345	112.3	27.5	51.5	58.8	8	8	8
3	181	262	41	28	0	3	3	3
4	173	334	172	812.5	37.7	5	5	4, 5
5	127	107	11	154	224	9	9	8, 9
6	60	40	6.9	110	70	9	9	8, 9
7	220	340	42	480	14	5	5	4, 5
8	170	320	53	520	3.2	5	5	4, 5
9	27	90	42	63	0.2	4	4	4
10	565	53	34	47	0	8	N	1, 6, 8
11	56	286	96	928	7	5	5	4, 5
12	200	48	14	117	131	9	9	8, 9
13	78	161	86	353	10	5	5	4, 5
14	32.4	5.5	1.4	12.6	13.2	9	9	8, 9
15	980	73	58	12	0	6	6	1, 6
16	160	130	33	96	0	2	2	2, 8
17	650	53	34	20	0	6	6	1, 6
18	95	110	160	50	0	3	3	1, 3
19	300	490	180	360	95	4	N	4
20	200	700	250	740	1	4, 5	4	4, 5

Step 10. Quantization, fired addresses coding, and sums the fired memory cells weights to obtain an output. If the output larger than a specified threshold value, then the fault type is confirmed.

Step 11. If the diagnosis is correct, go to step 12. Otherwise, go to step 13.

Step 12. Does the new gas ratio data to be diagnosed? Yes, go to step 9. No, go to step 14.

Step 13. Update the fires memory cells weights using Eq. (6).

Step 14. Save the memory weights and exit.

In Fig. 4, the left-hand side represents the off-line mode flowchart and right-hand side is the on-line diagnosis mode. The dashed line from left to right denotes the system first

time to be started. Fig. 5 shows the diagnosis program interface which was designed by Visual C++.

4. Case studies and discussions

4.1. Tested data diagnosis

To demonstrate the effectiveness of the proposed CMAC NN fault diagnosis method, twenty power transformer DGA results of from references [7–9] are tested. The detailed gas data are shown in Table 4, where the AFC, IEC and CMC express the actual fault type, the diagnoses of the IEC method and the proposed CMAC NN scheme, respectively. Through 10 iterative times training or learning performance evalua-

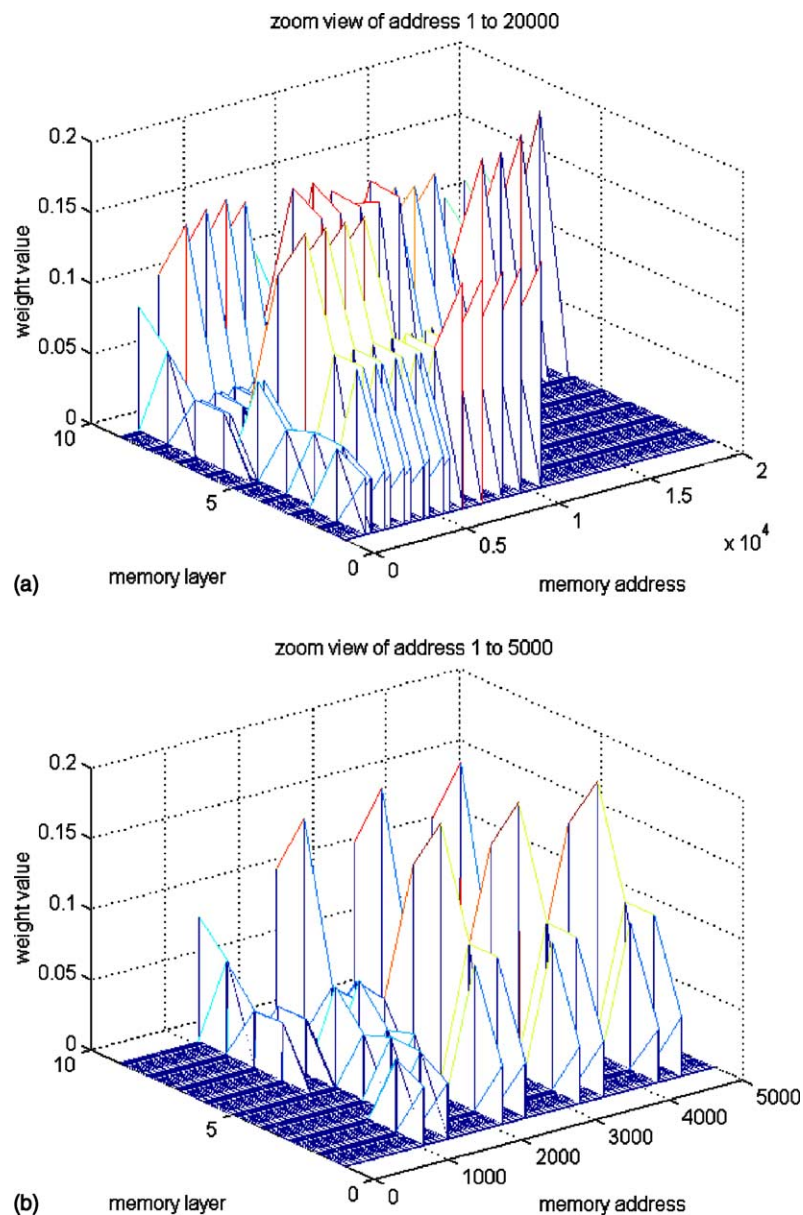


Fig. 6. (a) Weights distribution of memory layers. (b) Zoom view of weights distribution of memory layers.

Table 5
CMAC network parameters

Learning times	q_{\max}	A^*	bitn	Memory size	η	ε	β	STEP_1, STEP_2	STEP_3
10	12	10	5	32768	0.9	0.01	0.9	0.04	0.1

tion ($E_i < 0.01$), the memory weights mapping distribution drawing is shown in Fig. 6(a) and zoom view in Fig. 6(b). Fig. 6, similar to the cerebellum of human being, maps the feature of each fault type on a special memory layer. The larger difference of each layer distribution represents easier to diagnose the exact fault type. Similar distribution plot of each layer means the diagnoses with the multiple fault types easily. Table 5 lists the related parameters of CMAC network. Table 6 shows the detailed outputs of each node. The last column is the diagnosis fault types considering the threshold value $\eta = 0.9$ or 0.95 (bold type). Fault type nos. 4 and 5 are medium temperature and high temperature thermal faults, respectively. Therefore, fault type nos. 4 and 5 are similar, and the diagnoses of the 4th, 7th, 8th, 11th, and 13th data contain the multiple faults of nos. 4 and 5 and the possibility of fault type no. 5 is higher than no. 4. Observe the IEC code of Table 2, the codes of fault type nos. 8 and 9 are overlap somewhat. Consequently, the diagnoses of the 5th, 6th, 12th and 14th data output the multiple fault of nos. 8 and 9 are reasonable. Note that the 10th and 19th have no matching codes for diagnosis by the IEC method, but the results of the proposed method still detect the possible faults type. Moreover, the gas ratio of in the 5th and 6th data are far away the virtual training data bound (the bound of C_2H_2/C_2H_4 is 3, virtual training data take to 6, and the 5th and 6th data are 14 and 15.94, respectively). The diagnoses

still diagnose the fault type no. 9. The 20th data diagnosed by the IEC method only obtained the fault type no. 4, but the proposed CMAC NN scheme confirmed the multiple faults type nos. 4 and 5. The output values of nodes 4 and 5 are equal to 1 exactly. In short, the proposed method provided the most possible fault type diagnoses and never lost the actual fault type. It is proved that the proposed scheme not only diagnoses the main fault types of power transformers but also provides useful information for future fault trend analysis.

4.2. On-line training (learning)

If the wrong fault type happened in the diagnosis process, the on-line training is proceeded to refine the memory weights. The update rule is same as the Eq. (6). But since the characteristic of the proposed scheme is to provide the most possible fault type diagnosis, we do not use the anti-excited scheme to update the redundant fault type. Increase the threshold value η or just consider the maximum output value will filter the redundant fault type naturally.

4.3. Noise rejection test

To test the noise rejection ability of the proposed method, we added $\pm 5\%$ to $\pm 50\%$ random noise to the input states, i.e. added $\pm(5-50)\% \times \text{rand}(1)$ noise to the input states,

Table 6
Detail outputs of CMAC NN method (no noise)

No	Each node output									Diagnoses
	1	2	3	4	5	6	7	8	9	
1	1.0016	0.5000	0.4732	0.2321	0.0358	0.9286	0.8924	0.3351	0	1, 6
2	0.3336	0.3334	0.2102	0.1550	0.1379	0.2857	0.5179	0.9640	0.7334	8
3	0.7499	0.6666	1.0004	0.6937	0.4497	0.4286	0.3948	0.5291	0.1980	3
4	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.5447	0.5151	4, 5
5	0	0.2500	0	0.1163	0.2063	0	0.0634	0.9638	0.9774	8, 9
6	0.1698	0.6670	0.1052	0.3091	0.4125	0.1429	0.1444	1.0217	1.0175	8, 9
7	0.1660	0.5833	0.5261	0.9247	1.0016	0	0	0.6288	0.6343	4, 5
8	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.5447	0.5151	4, 5
9	0.1660	0.3333	0.8943	1.0000	0.7925	0	0	0.3861	0.2657	4
10	0.9182	0.8335	0.4199	0.1550	0	1.0000	0.8928	0.9227	0.4028	1, 6, 8
11	0.1660	0.3333	0.5261	0.9230	1.0000	0	0	0.3861	0.2657	4, 5
12	0	0.3334	0	0.1551	0.2746	0	0.3179	0.9640	0.9827	8, 9
13	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.5447	0.5151	4, 5
14	0	0.3334	0	0.1551	0.2746	0	0.3179	0.9640	0.9827	8, 9
15	0.9182	0.3334	0.4199	0.1550	0	1.0000	0.8924	0.3351	0	1, 6
16	0.3347	1.0000	0.3690	0.6941	0.7242	0.1429	0.1444	0.9206	0.8992	2, 8
17	0.9181	0.5004	0.4199	0.1550	0	1.0000	0.8927	0.5246	0.1321	1, 6
18	0.9166	0.5837	0.9993	0.3861	0.1738	0.5714	0.5346	0.4267	0.0642	1, 3
19	0.4996	0.5833	0.7363	0.9247	0.8291	0.2857	0.2653	0.7369	0.5115	4
20	0.1660	0.3333	0.5781	1.0000	1.0000	0	0	0.3861	0.2657	4, 5

Table 7
Detail outputs of CMAC NN method (noise: $\pm 10\%$ rand(1))

No.	Each node output									Diagnoses
	1	2	3	4	5	6	7	8	9	
1	1.0015	0.5000	0.4732	0.2321	0.0358	0.9286	0.8924	0.3351	0	1, 6
2	0.3336	0.3334	0.3152	0.1550	0.0682	0.2857	0.6474	0.9640	0.6143	8
3	0.7499	0.6666	0.9993	0.6937	0.4497	0.4286	0.3948	0.6214	0.1980	3
4	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.5447	0.5151	4, 5
5	0	0.3334	0	0.1163	0.2063	0	0.0634	0.9638	0.9774	8, 9
6	0.1260	0.6664	0.1052	0.3091	0.4125	0.1429	0.1444	1.0085	1.0093	8, 9
7	0.1660	0.5833	0.5261	0.9247	1.0016	0	0	0.6288	0.7639	4, 5
8	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.5447	0.5151	4, 5
9	0.1660	0.3333	0.8943	1.0000	0.7925	0	0	0.3861	0.2657	4
10	0.9182	0.8335	0.3152	0.1550	0.0682	0.8571	0.7530	0.9227	0.5323	1, 8
11	0.1660	0.3333	0.5261	0.9230	1.0000	0	0	0.3861	0.2657	4, 5
12	0	0.3334	0	0.1551	0.2746	0	0.3179	0.9640	0.9827	8, 9
13	0.1660	0.5000	0.5261	0.9227	1.0002	0	0	0.4669	0.4025	4, 5
14	0.0834	0.3334	0	0.1551	0.2746	0.0714	0.3179	0.9924	0.9991	8, 9
15	0.9182	0.3334	0.4199	0.1550	0	1.0000	0.8924	0.3351	0	1, 6
16	0.2494	0.8333	0.4209	0.8481	0.8622	0.1429	0.1444	0.9206	0.8987	8
17	0.9181	0.5004	0.4199	0.1550	0	1.0000	0.8927	0.5246	0.1321	1, 6
18	0.9166	0.5837	0.9993	0.3861	0.1738	0.5714	0.5346	0.4267	0.0642	1, 3
19	0.4996	0.5833	0.7363	0.7706	0.8291	0.2857	0.2653	0.7369	0.5115	N
20	0.1660	0.3333	0.5781	1.0000	1.0000	0	0	0.3861	0.2657	4, 5

where rand(1) is normal distribution random function between 0 and 1. Table 7 shows the detailed output of each node for input node with $\pm 10\%$ noise. Table 8 shows the diagnoses for different percentage noise added. In most test data, the diagnoses still output correct fault type even though the noise over 50%. Table 9 shows the diagnoses performance with different percentage noise, the last column are the IEC method results. As observed from Tables 8 and 9, it proved the proposed method with high noise rejection ability and capable of multiple faults detection.

Table 8
Diagnoses of $\pm 5\text{--}50\%$ random noise added using CMAC NN method

No.	Diagnoses for different percentage noise						
	5%	10%	15%	20%	25%	30%	50%
1	1, 6	1, 6	1, 6	1	1, 6	1, 6	1
2	8	8	8	8	8	8	8
3	3	3	3	3	3	3	3
4	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5
5	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9
6	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9
7	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5
8	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5, 8
9	4	4	4	4, 5	4	4	4
10	1, 6, 8	1, 8	6, 8	1, 8	2, 6, 8	1, 8	1, 8
11	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5
12	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9
13	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5
14	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9	8, 9
15	1, 6	1, 6	6	1, 6	1, 6	1, 6	1, 6
16	8	8	2, 8	8	2, 8	N	8
17	1, 6	1, 6	1, 6	1, 6	1, 6	1, 6	1, 6
18	1, 3	1, 3	1, 3	1, 3	1, 3	1	1
19	4	N	4	4	N	4	N
20	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5

Table 9
Diagnosis performance comparison

Noise (%)	CMAC NN method ($\eta = 0.9$, %)	IEC method (%)
± 0	100	86
± 5	95	81
± 10	90	72
± 15	100	68
± 20	95	58
± 25	95	38
± 30	90	38
± 50	85	34

4.4. Discussion

In this paper, we did not point out how to obtain the optimal design parameters, such as the quantization levels $q_{i,max}$, fired memory number A^* . This is related to the resolution problem and no exact answer to decide the optimal value. Fortunately, our diagnosis system provided flexible parameters setting function; it is easy to obtain proper parameters by try and error. However, the needed memory size is our major concern in real application. How to develop new fired memory addresses coding technology [18], to obtain optimal memory size for specified resolution, and to implement the diagnosis scheme by chip design are our future work and understudying.

5. Conclusion

This paper presents a novel CMAC NN fault diagnosis method for power transformers. Using the characteristic of generalization, local reflexive action and self-learning abil-

ity, the proposed scheme achieves at least the following merits: (1) do not require the actual data to train the CMAC network and high diagnosis accuracy is obtained; (2) detect the main fault type and provide useful information for future fault trends and multiple faults analysis; (3) high noise rejection ability; (4) suit to non-training data and associate the most similar fault type; and (5) do not require extra any expert experience to train the CMAC network. The tested data demonstrate the success of proposed scheme.

Acknowledgements

We gratefully acknowledge the support of the National Science Council, Taiwan, ROC, for financial support under grant NSC-91-2213-E-167-014.

Appendix A. List of symbols

A^*	the number of associated memory cells
CMAC	cerebellar model articulation controller
bitn	the minimum bit numbers to decode the segment address
DGA	dissolved gas analysis
E_i	performance evaluation value of the i th fault pattern
n_g	training data number
q_{imax}	quantization levels for the i th input
V_j	the j th fired address
V_{ji}	the i th segment address of j th fired address
$w_{v_j}^{new}$	new weight of V_j memory address
$w_{v_j}^{old}$	old weight of V_j memory address
y	the actual output
y_d	the desired output

Greek letters

β	the learning gain
ε	small positive number
η	threshold value

References

- [1] J.J. Kelly, Transformer fault diagnosis by dissolved-gas analysis, IEEE Trans. Ind. Appl. 16 (6) (1980) 777–782.
- [2] R.R. Rogers, IEEE and IEC codes to interpret faults in transformers, using gas in oil analysis, IEEE Trans. Electr. Insulation 13 (5) (1978) 349–354.
- [3] IEC Publication 599, Interpretation of the Analysis of Gases in Transformers and Other Oil-Filled Electrical Equipment in Service, first ed., 1978.
- [4] IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed transformers, IEEE Standard C57.104-1991.
- [5] E. Dornenburg, W. Strittmater, Monitoring oil cooling transformers by gas analysis, Brown Boveri Rev. 61 (1974) 238–274.
- [6] C.E. Lin, J.M. Ling, C.L. Huang, An expert system for transformer fault diagnosis using dissolved gas analysis, IEEE Trans. Power Deliv. 8 (1) (1993) 231–238.
- [7] Y. Zhang, X. Ding, Y. Liu, P.J. Griffin, An artificial neural network approach to transformer fault diagnosis, IEEE Trans. PWRD 11 (4) (1996) 1836–1841.
- [8] Z. Wang, Y. Liu, P.J. Griffin, A combined ANN and expert system tool for transformer fault diagnosis, IEEE Trans. Power Deliv. 13 (4) (1998) 1224–1229.
- [9] G. Zhang, K. Yasuoka, S. Ishii, Application of fuzzy equivalent matrix for fault diagnosis of oil-immersed insulation, in: Proceedings of the 13th International Conference on Dielectric Liquids (ICDL'99), Nara, Japan, 1999, pp. 400–403.
- [10] Q. Su, C. Mi, L.L. Lai, P. Austin, A fuzzy dissolved gas analysis method for the diagnosis of multiple incipient faults in a transformer, IEEE Trans. Power Syst. 15 (2) (2000) 593–598.
- [11] J.J. Dukarm, Transformer oil diagnosis using fuzzy logic and neural networks, in: 1993 Canadian Conference on Electrical and Computer Engineering, vol. 1, 1993, pp. 329–332.
- [12] H.T. Yang, C.C. Liao, J.H. Chou, Fuzzy learning vector quantization networks for power transformer condition assessment, IEEE Trans. Dielectr. Insulation 8 (1) (2001).
- [13] T. Yanming, Q. Zheng, DGA based insulation diagnosis of power transformer via ANN, in: Proceedings of the Sixth Conference on Properties and Applications of Dielectric Materials, 1999, pp. 133–136.
- [14] K.F. Thang, R.K. Aggarwal, D.G. Esp, A.J. McGrail, Statical and neural network analysis of dissolved in power transformers, in: IEE Dielectric Materials, Measurements and Applications Conference Publication 473, 2000.
- [15] J.S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), Trans. ASME J. Dyn. Syst. Measure. Control 97 (1975) 220–227.
- [16] Y.F. Wong, A. Sideris, Learning convergence in the cerebellar model articulation controller, IEEE Trans. Neural Netw. 3 (1) (1992) 115–121.
- [17] D.A. Handeiman, S.H. Lane, J.J. Gelfand, Integrating neural networks and knowledge-based systems for intelligent robotic control, IEEE Control Syst. Mag. (1990) 77–86.
- [18] C.P. Hung, M.H. Wang, Fault diagnosis of air-conditioning system using CMAC neural network approach, in: Advances in Soft Computing—Engineering, Design and Manufacturing, Springer, 2003, pp. 1–11.