METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.pg.gda.pl

ANALOG CIRCUIT FAULT CLASSIFICATION USING IMPROVED ONE-AGAINST-ONE SUPPORT VECTOR MACHINES

Jiang Cui, Youren Wang

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing City, Jiangsu Province, China
(✉cuijiang@nuaa.edu.cn, Phone: +86 13770653061*)*

Abstract

This paper presents a novel strategy of fault classification for the analog circuit under test (CUT). The proposed classification strategy is implemented with the one-against-one Support Vector Machines Classifier (SVC), which is improved by employing a fault dictionary to accelerate the testing procedure. In our investigations, the support vectors and other relevant parameters are obtained by training the standard binary support vector machines. In addition, a technique of radial-basis-function (RBF) kernel parameter evaluation and selection is invented. This technique can find a good and proper kernel parameter for the SVC prior to the machine learning. Two typical analog circuits are demonstrated to validate the effectiveness of the proposed method.

Keywords: analog circuit, fault classification, Support Vector Machines classifier, fault dictionary, kernel parameter.

1. Introduction

Faults classification is an important but complicated task in analog electronic circuits. Two methods, namely the Simulation-After-Test (SAT) and Simulation-Before-Test (SBT) [1], are prevalent now. Fault dictionary (FD), which belongs to the SBT method, seems to be a reliable and important approach in the domain of analog testing [2]. The conventional FD method usually calculates the Euclidean distances between the query sample and all the centroids of fault classes [3]. In this method, a smallest distance indicates an occurrence of fault or fault-free case. Considering the tolerance effect of components or measurement inaccuracies, the FD method always defines some tolerance limits, which can be applied to the samples in measurement space or feature space [4, 5]. Sometimes, these defined limits (especially for parametric faults) can overlap, and hence, bring difficulties and uncertainties to the consequent fault decisions.

Focusing on these problems, fuzzy systems and neural networks (NN) [2, 6-9] are then employed to perform fault classification. These methods can achieve good performances, but they belong to the so-called non-deterministic methods [10]. For instance, in a fuzzy system, the designation of membership functions and inference rules depends on the experience of engineers, and thus, different designations will probably result in different fault decisions. The NN work well in the domain of analog circuit testing, but the NNs have many drawbacks. For instance, the back-propagation NN (BPNN), which is a widely used NN in analog or mixed-signal system testing, can be easily entrapped into the local minima in the training phase and different trainings will probably lead to different results. In order to improve this classifier, some additional and complicated measures always need to be taken. For instance, in [7], the authors use the genetic algorithm (GA) to optimize the training process of a classifier. The Radial-Basis-Function NN (RBFNN) is another well-known NN, and this NN is a local approximation network, whose number of hidden neurons will become very large even for a

Article history: received on Apr. 18, 2011; accepted on Sept. 14, 2011; available online on Dec. 12, 2011; DOI: 10.2478/v10178-011-0055-7.

moderate size of fault classes. Bayesian NN is another useful classifier when overlapped measurements are considered. This classifier is based on the probabilistic estimation and thus, a lot of fault feature samples are needed. Another drawback for these NNs is that many parameters (*e.g.* number of neurons and layers, activation functions and their parameters, etc.) need to be configured manually, and this will result in an unstable classifier structure.

In this paper, we employ a one-against-one SVC to perform analog electronic circuit fault classification. The standard SVC can be viewed as a special NN, but it has some good characteristics, e.g., fast convergence to the global minima, less adjustable parameters (*note*: once the training parameters are determined, different trainings will result in identical results and from this point, this classifier is deterministic), excellent generalization capability, strong learning capability with small samples, etc. In the domain of analog testing, two important SVCs are available. In [10], Grzechca and Rutkowski employ the one-against-rest SVC to perform fault detection and localization task in filter and FPAA circuits. Salat and Osowski [11] adopt a one-against-one SVC to classify hard faults in analog filters and good results are achieved. In [12], a RC ladder network is diagnosed with the one-against-one SVC and in this case, the soft faults are considered. In [13], analog circuit fault detection is performed with a binary SVC. The SVC can also be applied to the fault diagnosis of high-frequency integrated circuits, for instance, the RF low noise amplifier can be diagnosed at the transistor level [14]. A simplified review of these literatures indicates that the SVC is applicable to analog circuit fault classification. However, in most of these investigations, only conventional SVCs are used and improvements for these classifiers are not considered. Focusing on these problems, we propose an improved one-against-rest SVC to test analog circuits in [15] and this improved SVC can reduce the testing time apparently. In addition, we proposed another SVC in [16], and this classifier utilized a fault dictionary and one-against-rest SVC to achieve good results, and at the same time, the testing time can be reduced significantly. This method can be extended to the one-against-one SVC, which, according to [17], is suitable for practical use. In addition, how to select a proper kernel parameter in SVC designation is very important, but this problem is not well addressed in our previous works.

The presented method in this paper has two contributions. The first one is to improve the conventional one-against-one SVC to reduce the testing time. The second contribution depends on a proposed technique which can select a good kernel parameter of the SVC prior to the training. This second contribution can assure the SVC to have good generalization (*i.e.* diagnosis performance) without trainings for the optimal parameter. Hence, many calculations can be saved. This paper is organized in the following order. In section 2, the basic theory for SVC is outlined, and also in this section, the improvement for a one-against-one SVC is presented. The selection technique for the SVC kernel parameter is proposed in section 3. Our method is validated by the experiments from three cases in two circuits in section 4. Important conclusions are drawn in section 5.

2. The proposed classifier

2.1. Binary Support Vector Machines Classifier (BSVC)

The standard support vector machines classifier, invented by Vapnik *et al* [18] , can perform binary classification. In this study, this classifier is called a binary support vector machines classifier (BSVC).

Let $\{(\mathbf{x}_i, y_i)\}$ (*i*=1, 2, …, G and G is the number of samples) be a set of training samples. Each sample $\mathbf{x}_i \in R^d$ is assigned to a binary set $y_i \in \{+1, -1\}$, in which label +1 and -

1 indicates that training sample ***x*** belongs to the positive and negative patterns, respectively. Training of BSVC is a typical quadratic optimization problem, which tries to find the maximal margin between two groups of samples while the recognition error is minimized. Solving this optimization problem will need the anticipation of Lagrange Multipliers, slack parameters and penalty constant *C*. In order to find the solution, the so-called KKT condition must be met. Finally, the BSVC can perform pattern classification of a query sample ***x*** with the following decision function:

$$f(\mathbf{x}) = \sum_{k=1}^{n_{sv}} \lambda_k \, y_k < \phi(\mathbf{x}), \phi(\mathbf{x}_k) > + b^*, \tag{1}$$

where: $\lambda_k > 0$ is the Lagrange Multiplier corresponding to the k^{th} support vector (SV) x$_k$; y_k is the label of the k^{th} SV; <•,•> stands for the inner product; $\phi(\bullet)$ is a nonlinear mapping function which can map the samples in the measurement space to a high-dimensional space, where the samples can become linear separable; n_{sv} is the number of the SVs; b^* is the bias of the decision hyper-plane.

Generally, the explicit expression form of $\phi(\bullet)$ is hard to obtain, thus, the kernel function $K(\mathbf{x},\mathbf{y}) =< \phi(\mathbf{x}), \phi(\mathbf{y}) >$ is introduced to replace the inner product and then, the assignment of $\mathbf{x}$ can be determined with:

$$f(\mathbf{x}) = \sum_{k=1}^{n_{sv}} \lambda_k \, y_k K(\mathbf{x}, \mathbf{x}_k) + b^*. \tag{2}$$

In our experiments, the RBF function is considered. According to [19], this nonlinear kernel function can usually lead to a good performance:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-|\mathbf{x} - \mathbf{x}_i|^2 / \sigma^2), \tag{3}$$

where σ refers to the spread parameter of RBF function.

2.2. The multi-class SVC

The BSVC can perform binary classification, but this is not enough when it goes to a multi-class classification problem, such as analog faults classification. In order to solve a multi-class problem, one-against-rest SVC and one-against-one SVC can be employed. For the one-against-rest SVC with Winner Take All (WTA) rule, *N* BSVCs are constructed for *N* classes, and for each training process, the i^{th} class is separated from the rest *N*-1 classes. In the decision stage, all decision functions are calculated to determine the assignment of $\mathbf{x}$:

$$\arg \min_{i=1,2,\ldots,N} [f_i(\mathbf{x})], \tag{4}$$

where $f_i(\mathbf{x})$ is the decision function of the i^{th} BSVC.

The second conventional multi-class method is the one-against-one SVC. In the training stage, all possible pair classes are trained and altogether *N*(*N*-1)/2 BSVCs are constructed. In the decision stage, Max-wins strategy is adopted. The decision function for the BSVC, which is formed by class *i* and *j* (*i* ≠*j*), can be expressed as:

$$f_{ij}(\mathbf{x}) = \sum_{k=1}^{n_{sv}^{i,j}} y_k^{i,j} \lambda_k^{i,j} K(\mathbf{x}, \mathbf{x}_k^{i,j}) + b_{i,j}{}^*, \tag{5}$$

where: $n_{sv}^{i,j}$ is the SV number, $\lambda_k^{i,j}$ is the Lagrange multiplier of the k^{th} SV, $y_k^{i,j}$ is the label of the k^{th} SV, $b_{i,j}^{*}$ is the bias of this BSVC.

In the applications of analog faults classification, when the number of faults N becomes very large, one-against-one SVC will need too much BSVCs to perform fault decision and testing time consumed will become unendurable. In the domain of analog testing, the improvement of the one-against-one SVC for testing time reduction is seldom reported.

2.3. *Principle of the proposed SVC*

The fault classifier to be improved in our experiments is based on the one-against-one SVC. The conventional one-against-one SVC calculates all the decision functions for each testing and this seems to be unnecessary for most of diagnosis cases. In fact, it is only some of the decision functions, other than all the decision functions, that really determine the assignment of the query sample. This can be explained with Fig. 1.

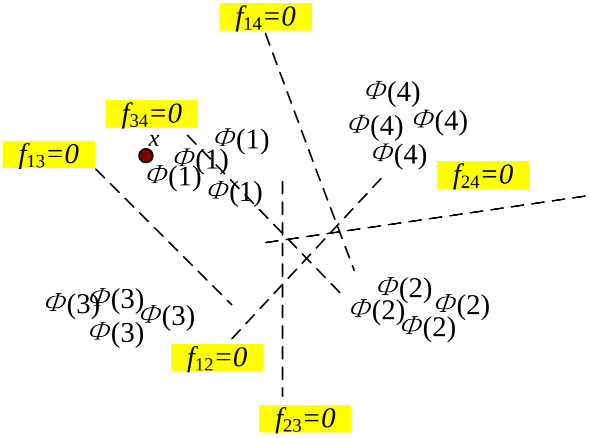


Fig. 1. One-against-one SVC for four classes in the high-dimensional space.

Fig. 1 illustrates four fault classes (‘1’,’2’,’3’ and ‘4’ respectively) in the high-dimensional space via the mapping function ϕ. In this case, the one-against-one SVC will need at least 6 BSVCs for pattern classification. Let BSVC$_{ij}$ be the BSVC that separates class ‘i’ and ‘j’ (i=1, 2, 3 and i<j), and f_{ij} the decision function of BSVC$_{ij}$, whose optimal decision hyper-plane is indicated with a dashed line. In our study, set the label of class ‘i’ and the other class ‘j’, to be -1 and +1, respectively. Assume to test a sample **x** (shown in Fig. 1) belonging to class ‘1’, and the outputs (*i.e.* the signal of f_{ij}) for these BSVCs are as follows: f_{12}=-1, f_{13}=-1, f_{14}=-1, f_{23}=+1, f_{24}=+1, f_{34}=-1. Hence, the votes for class ‘1’ is 3, the votes for class ‘2’ is 0, the votes for class ‘3’ is 2 and the votes for class ‘4’ is 1.

According to the Max-wins strategy, the query sample should be assigned to the class which gets the most of votes. In this case, the votes of BSVCs for class ‘1’ are all valid (negative) and for the other classes, however, some of their votes are invalid. This important information indicates that the query sample can be determined directly by the votes of the BSVCs corresponding to some class, not necessarily all the class concerned. If a set of BSVCs corresponding to a certain class gives effective votes (a threshold can be defined to control this effectiveness), then, the class assignment can be determined directly without calculating the subsequent decision functions. This method will save many calculations and hence, the testing time can be reduced.

But, a problem arises, namely that which set of BSVCs should be selected and calculated. We can reuse the pre-processing method from [16]. Firstly, we use a FD (**FD$_1$**) with Euclidean distance match to find this set of BSVCs. **FD$_1$** contains the centroids of all fault classes. Assume each fault class to contain L training samples. The centroids are defined as below:

$$\mathbf{C}_j = \frac{1}{L}\sum_{i=1}^{L}\mathbf{x}_i^j \, , \tag{6}$$

where $\mathbf{x}_i^j$ is the i^{th} training sample of fault class j ($j = 1,..., N$).

Secondly, the conventional one-against-one SVC is designed and each BSVC is trained and saved. All BSVCs form the second FD (**FD$_2$**), as shown in Fig. 2. For the convenience, let BSVC_{ij} ($i < j$) be the BSVC which separates fault class i from class j, f_{ij} be the decision function of BSVC_{ij} . The fault decision can be divided into following steps:

(1) Input query sample **x** to be tested;

(2) Calculation of Euclidean distances d_i (i=1,…,N) between **x** and all centroids via **FD$_1$** according to the equation as follows:

$$d_j = |\mathbf{x} - \mathbf{C}_j| \tag{7}$$

(3) Sort d_i in the ascending order, in which the corresponding index is saved to an array *index*[j] ;

(4) Initialize j=1;

(5) k=*index*[j] and implement calculations of f_{1k}, f_{2k}, …, f_{kN} via **FD$_2$**.

(6) If $\mathbf{x} \in$ fault class k, then quit; else go to next step (7);

(7) Let j++; if j>N, go to next step (8); else go to step (5);

(8) Max-wins strategy is invoked to determine the assignment of **x**.

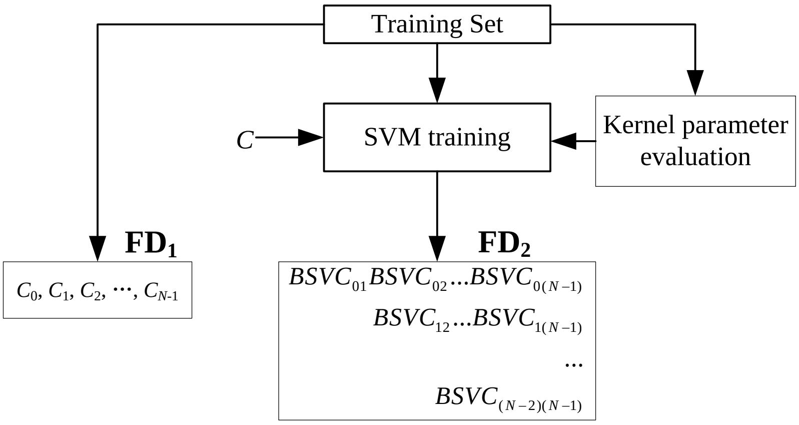

Training Set
SVM training
Kernel parameter evaluation
C
FD$_1$
FD$_2$
C_0, C_1, C_2, ⋯, $C_{N\text{-}1}$
$BSVC_{01}BSVC_{02}...BSVC_{0(N-1)}$
$BSVC_{12}...BSVC_{1(N-1)}$
...
$BSVC_{(N-2)(N-1)}$

Fig. 2. Construction of two fault dictionaries.

Generally, the closer the Euclidean distance is, the higher the probability with which **x** belongs to the corresponding fault class. In step (6), look up the anticipated BSVCs in **FD$_2$** to implement fault classification with the indicator function. The BSVCs involved here are related to fault class f_k (2<k<N):

$$\begin{array}{l} sf_1(x) = sign(f_{1k}(\mathbf{x})) \\ sf_2(x) = sign(f_{2k}(\mathbf{x})) \\ \dots \\ sf_N(x) = sign(f_{kN}(\mathbf{x})) \end{array} . \tag{8}$$

Also in this step, the vote strategy needs to be implemented. If the votes of fault class k exceeds some threshold ***Thres***, we simply assign **x** to fault class k, otherwise the next loop will begin. Sometimes, **x** cannot be assigned to any of the available fault classes with this decision method and then, for this case, the program will switch to step (8), *i.e.*, the conventional one-against-one SVC with Max-wins decision adopted in [17].

Basically, the one-against-one SVC adopts the decision strategy with Max-wins, and all BSVCs take part in each fault decision. From this point, the conventional one-against-one SVC is a classifier with stable structure (*note:* this conclusion also goes to the conventional one-against-rest SVC). Our proposed method, however, only employs a small portion of BSVCs to perform the classification task, hence the computational complexity will be reduced with no doubt. In our proposed method, the candidate BSVCs to be calculated are mainly determined by the $\mathbf{FD_1}$ and this classifier has an unstable structure, thus, the performance seems to be unstable. Despite this, according to the principle of the proposed classifier, the query sample can be claimed to be a certain fault class only when the votes reach to some predefined threshold ***Thres***. Apparently, a larger ***Thres*** can assure the SVC to classify the sample with high probability correctly. In this paper, this threshold is set to be N-1 directly and in our experiment, a smaller ***Thres*** will result in poor classification performance.

3. Kernel parameter selection algorithm

Two parameters are crucial to the generalization capability of SVC. One is the kernel function parameter and the other is the penalty constant C. The latter is usually set in the training phase, and it controls the balance of maximal margin and the classification error. In this paper, for all the classifiers, C equals to 100, with which good results can be obtained.

This paper proposes an evaluation method of the kernel function parameter. This method is enlightened by the SVC theory. According to the principle of SVC, binary classes can be separated easily if the distance between them is large. For instance, Fig. 3 gives an illustration which describes binary classes (‘t’ and ‘y’) mapping from the measurement space to a high-dimensional space, via some mapping function Φ.

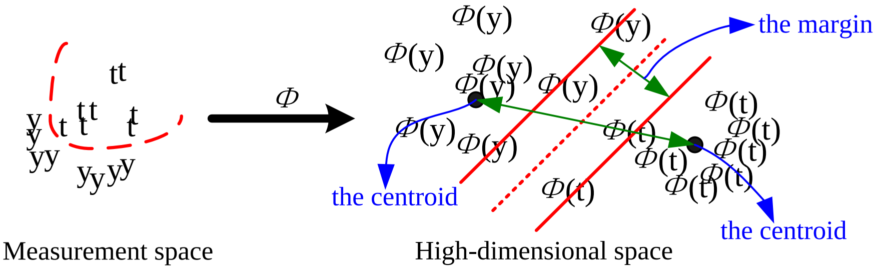


Fig. 3. Measurement space is converted to a high-dimensional space via the mapping function Φ.

In the measurement space, two classes are nonlinear separable (the decision boundary is a nonlinear dashed curve) and in the high-dimensional space, these samples are linear separable (the optimal decision boundary is a dashed beeline in this figure, and the samples located on two beelines are support vectors of two classes).

Generally, a large margin (shown in Fig. 3) always means a large distance of centroids (shown in Fig. 3) of two classes. In other words, a large distance of centroids of two classes also indicates a large SVC decision margin. Here, we employ the Euclidean distance between the centroids of two classes to evaluate the margin coarsely, namely, the generalization capability of SVC. The Euclidean distance evaluation should be implemented in the high-dimensional space, not in the measurement space, because in the high-dimensional space, most of the samples belonging to the same class will gather on one side of the decision beeline. Also, calculation of Euclidean distance in the high-dimensional space will establish a qualitative bridge between the kernel parameter and the generalization capability of SVC.

Let $\mathbf{x}_i^k$ be the i^{th} sample belonging to fault class k. Then define the Euclidean distance hd_{jk} between two fault classes in the high-dimensional space as:

$$hd_{jk} = |\phi(\mathbf{C}_j) - \phi(\mathbf{C}_k)|. \tag{9}$$

Usually, the mapping function is impossible to obtain and thus, the following trick is used:

$$hd_{jk} = \sqrt{|\phi(\mathbf{C}_j) - \phi(\mathbf{C}_k)|^2}.$$

Considering $K(\mathbf{x},\mathbf{y}) = <\phi(\mathbf{x}),\phi(\mathbf{y})>$ will lead to the following expression:

$$\begin{aligned} &|\phi(\mathbf{C}_j) - \phi(\mathbf{C}_k)|^2 \\ &= |\frac{1}{L}\sum_{i=1}^{L}\phi(\mathbf{x}_i^j) - \frac{1}{L}\sum_{i=1}^{L}\phi(\mathbf{x}_i^k)|^2 \\ &= \frac{1}{L^2}[\sum_{i=1}^{L}\sum_{l=1}^{L}K(\mathbf{x}_i^j,\mathbf{x}_l^j) - 2\sum_{i=1}^{L}\sum_{l=1}^{L}K(\mathbf{x}_i^j,\mathbf{x}_l^k) + \sum_{i=1}^{L}\sum_{l=1}^{L}K(\mathbf{x}_i^k,\mathbf{x}_l^k)] \end{aligned} \tag{10}$$

Hence, considering equation (3), the Euclidean distance measure in the high-dimensional space turns out to be:

$$\begin{aligned} hd_{jk} = \frac{1}{L}[&\sum_{i=1}^{L}\sum_{l=1}^{L}\exp(-|\mathbf{x}_i^j - \mathbf{x}_l^j|^2/\sigma^2) \\ &-2\sum_{i=1}^{L}\sum_{l=1}^{L}\exp(-|\mathbf{x}_i^j - \mathbf{x}_l^k|^2/\sigma^2) \\ &+\sum_{i=1}^{L}\sum_{l=1}^{L}\exp(-|\mathbf{x}_i^k - \mathbf{x}_l^k|^2/\sigma^2)]^{1/2} \end{aligned} \tag{11}$$

The above equation contains the kernel parameter σ and it can evaluate the effect of kernel parameter on one BSVC, which contains class *j* and *k*. The proposed SVC is composed of many BSVCs. Hence, the selection rule should be applied to all the BSVCs. Generally, a good kernel parameter should make all the BSVCs have large distances and thus a good generalization capability for the SVC can be achieved. Based on these considerations, the following rule needs to be constructed:

$$J = [\frac{1}{N(N-1)}\sum_{j}\sum_{k} hd_{jk}] \bullet \min\{hd_{jk}\}, j < k \le N \tag{12}$$

In the above expression, $[\frac{1}{N(N-1)}\sum_{j}\sum_{k} hd_{jk}]$ can evaluate the effect of kernel parameter on the whole SVC and the second part (*i.e.* $\min\{hd_{jk}\}$) can evaluate the effect of kernel parameter on the BSVC under worst case. Thus, a larger *J* indicates that the corresponding kernel parameter can assure the SVC to have a good generalization capability. The proposed technique of kernel parameter selection is implemented in the following steps:

(1) Confine the kernel parameter σ_z to some range, for example, in this paper, $\sigma_z \in$ {z=1, 2, … ,10 | 0.01, 0.1, 1, 2, 4, 8, 16, 32, 64, 128};

(2) initialize *z*=1;

(3) Calculate the Euclidean distances hd_{jk}^z in the high-dimensional space for all BSVCs under σ_z according to equation (11);

(4) Calculate J_z according to equation (12);

(5) if z <10, z=z+1, go to step (3); else, go to step (6);

(6) Select the best kernel parameter according to: $\sigma_z = \max_z(J_z)$.

4. Case study

4.1. A nonlinear circuit

The first circuit is a differential amplifier (DAmp) with 2N2222 transistors (Q1 and Q2, shown in Fig. 4) and in this nonlinear circuit, Q3 and Q4 form a basic current mirror as the current bias of the circuit. A sine wave U_i (amplitude: 50 mV, frequency: 10 Hz) is used to excite the circuit and the responses are collected via Vout as shown in this figure.

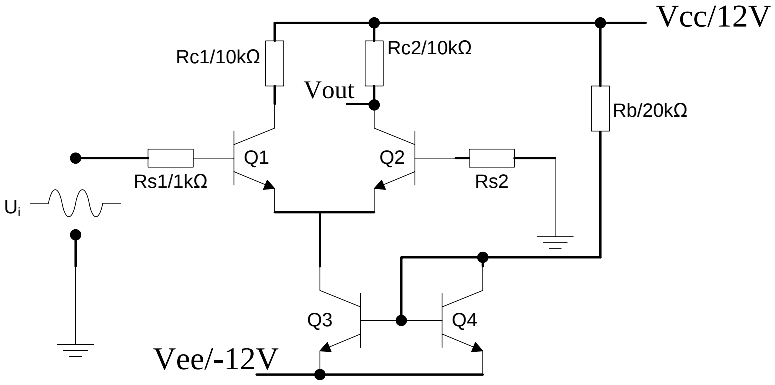


Fig. 4. A differential amplifier.

For this circuit, soft faults for resistors (Rc1, Rc2 and Rb) are considered because these soft faults can make the working point of the circuit shift. In addition, the transistors are prone to getting faulty, and hence, hard faults for the transistors (Q1~Q4) are designed. In order to simplify the problem, only the single faults are considered. Sometimes, different faults can result in identical circuit responses, and these faults are partitioned into the so-called ambiguity group (AG). Considering the AGs together with the tolerance considerations, this circuit can generate altogether 14 fault classes which are described in [16] in details.

According to the operation principle of this circuit, the waveform of Vout under normal condition is cyclic when the input stimulus is considered. When the circuit is faulty, its operation point will change and thus, the waveforms of Vout will become different. Three simple features in the time domain are extracted:

$$[f_0, f_1, f_2]$$

where, f_0=min(S) can extract the minimal value from the original sample sequence S; f_1=max(S) can extract the maximal value from S; f_2=mean(S) is the mean value of S.

Although the extracted fault features are simple, they are quite effective in future pattern classification.

4.2. A linear circuit

Another analog circuit is used in our study to validate the proposed method. The used circuit is a filter, whose nominal values for the components are shown in Fig. 5. This circuit is composed of four functional modules, whose outputs are labelled with u_1, u_2, u_3 and u_4, respectively. The final output from amplifier Q4 will function as a high-pass-filter (HPF). The node u_4 is assumed to be accessible. The nominal tolerances for the resistors and capacitors are 5% and 10%, respectively.

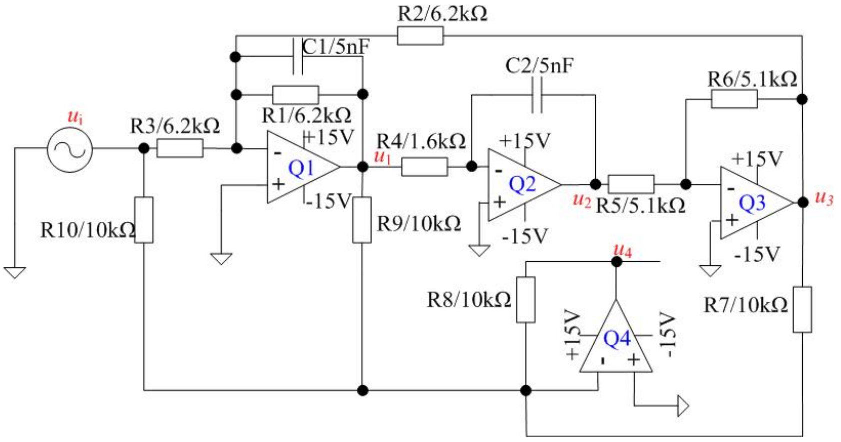


Fig. 5. An analog filter.

In this experiment, R1, R2, R3, R4, C1 and C2 are to be diagnosed because these components can affect the output of the circuit significantly.

4.2.1. Simulation experiment

In this experiment, the tolerances for the components are considered by using Monte Carlo analysis in the Pspice software. The fault classes and fault models for the faulty components are identical to those described in [8] and [16], respectively.

In this investigation, the wavelet decomposition (WD) is used to extract the features for the faulty circuit. We adopt Haar function as the mother function of the WD because this function is suitable for extracting faulty features from the circuit stimulated by a narrow pulse [8]. The WD is implemented from level one to level five and only the approximation coefficients at each level are retained.

In order to compress the fault information further, the approximation coefficients are used to calculate the so-called energy features [5]. Hence, the size of the fault feature is five.

4.2.2. Actual experiment

In order to investigate the effectiveness of our proposed method in a more practical way, we try to collect real data of the actual circuit and implement the verification experiment. Considering the difference between the simulated circuit and actual circuit [8], we did not use the simulated data to perform the faults classification for an actual circuit. Hence, we will have to collect the actual data to perform machine learning.

We designed the PCB of the circuit and the actual values of components were measured with Agilent 4263B (a LCR measurement instrument). The actual values of the components, as well as their fault values are all listed in Table 1. Each component will generate soft faults which are higher or lower than its nominal value (indicated with "↑" and "↓" respectively). The health state of the CUT can also be viewed as a special fault class "nf". Thus, altogether 13 fault classes need to be classified.

In this experiment, the tolerances for the components are not considered. However, the effect of various noise, A/D converter precision errors, power voltage fluctuations, etc, can be viewed as an equivalent of tolerance.

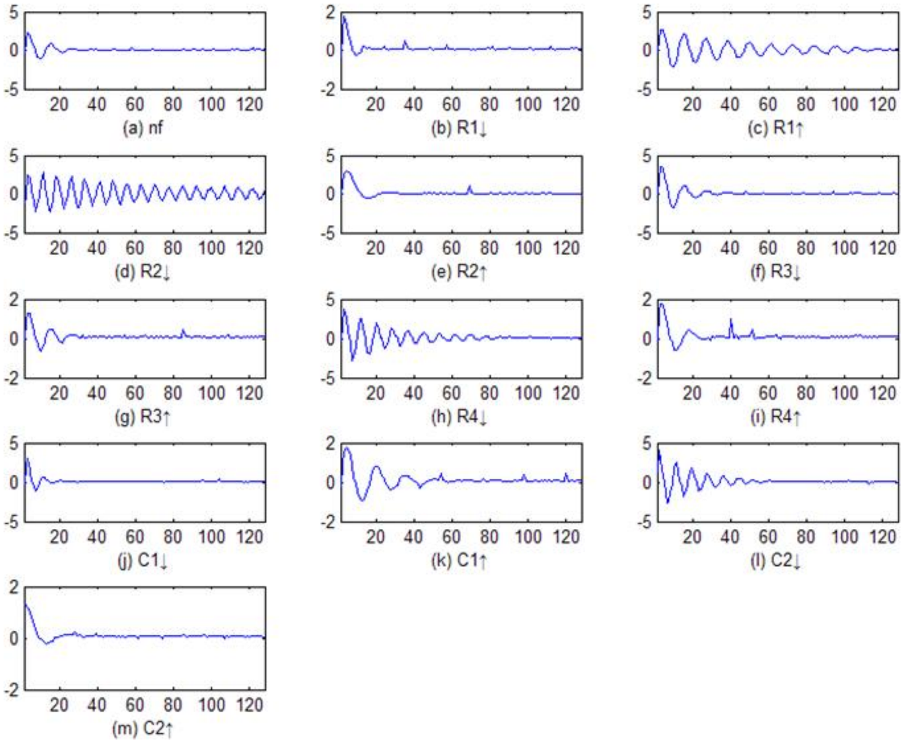
The stimulus generated by the HP 33120A is a pulse with 10 μs duration and 5 V peak. In order to generate such a stimulus with narrow duration, this instrument works under burst mode. For each fault class, 100 samples are generated while the faulty component value is set.

Table 1. Fault classes for the second CUT. Actual and fault values are also specified.

Order	Fault class	Actual value	Fault value
1	nf	-	-
2	R1↓	6.2 kΩ	3 kΩ
3	R1↑	6.2 kΩ	15 kΩ
4	R2↓	6.2 kΩ	2.43 kΩ
5	R2↑	6.2 kΩ	18 kΩ
6	R3↓	6.2 kΩ	3.8 kΩ
7	R3↑	6.2 kΩ	12 kΩ
8	R4↓	1.6 kΩ	750 Ω
9	R4↑	1.6 kΩ	2.5 kΩ
10	C1↓	5 nF	2.36 nF
11	C1↑	5 nF	9.47 nF
12	C2↓	4.6 nF	2.3 nF
13	C2↑	4.6 nF	14.9 nF

The data acquisition task is implemented with a digital signal controller (DSC) TMS320F28335, whose working performance is configured to be 150 MIPS. We use two A/D channels to collect fault samples. The first channel is to track the rising edge of the input stimulus and then, the second channel is triggered simultaneously to acquire the output of the circuit for 128 consecutive dots. For each fault class, 100 samples are collected.

Fig. 6 gives a sample illustration for each fault class. We can observe from the faulty waveforms of the circuit that a different fault class will result in different waveforms. Still, we adopt the feature extraction of Haar WD technique, which is identical to that described in the simulated filter.



(a) nf (b) R1↓ (c) R1↑ (d) R2↓ (e) R2↑ (f) R3↓ (g) R3↑ (h) R4↓ (i) R4↑ (j) C1↓ (k) C1↑ (l) C2↓ (m) C2↑

Fig. 6. One sample illustration for each fault class.

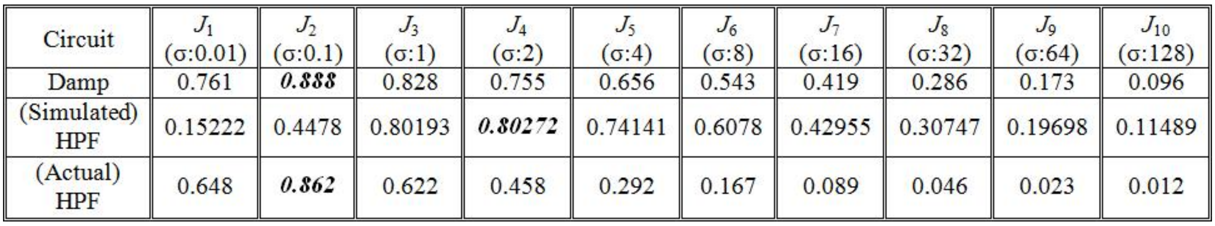
4.3. Results

We write Matlab7.0 codes for all classifiers in terms of sample pre-processing, training and testing, etc. All codes run on a P4 personal computer with 2.8 GHz dual CPUs and 2 GB RAM. Our running system is Windows XP.

Before the machine learning, the feature set should be normalized to have zero mean and unitary variance. Second, the whole feature set is divided into two parts: training set and testing set. The training set is used to generate two FDs and the centroids of fault classes. The testing set is used to evaluate the generalization capability of the SVC.

In three cases for two circuits, each fault class will contribute 10 samples to form the training set and the remaining samples are for next testing. The algorithm for kernel parameter selection is applied to the training set for the circuits, and the final evaluation values for rule J_z are listed in Table 2. For each circuit, the maximal J_z is marked in bold and italic font.

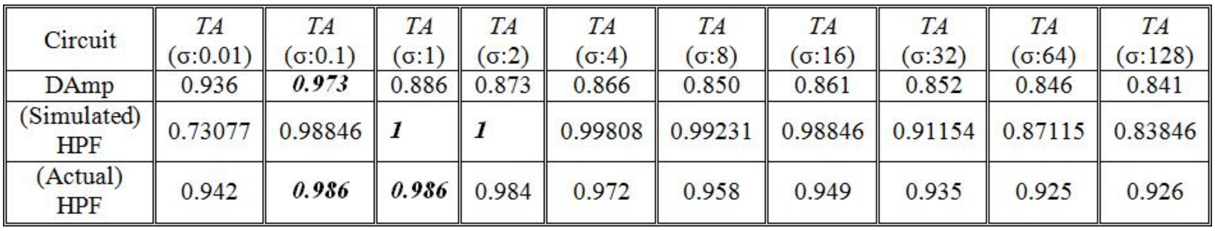
Table 2. Kernel parameter selection for three cases.



Circuit	J_1 (σ:0.01)	J_2 (σ:0.1)	J_3 (σ:1)	J_4 (σ:2)	J_5 (σ:4)	J_6 (σ:8)	J_7 (σ:16)	J_8 (σ:32)	J_9 (σ:64)	J_{10} (σ:128)
Damp	0.761	***0.888***	0.828	0.755	0.656	0.543	0.419	0.286	0.173	0.096
(Simulated) HPF	0.15222	0.4478	0.80193	***0.80272***	0.74141	0.6078	0.42955	0.30747	0.19698	0.11489
(Actual) HPF	0.648	***0.862***	0.622	0.458	0.292	0.167	0.089	0.046	0.023	0.012

In order to illustrate the effectiveness of the proposed kernel parameter selection technique, exhaustive searching for the optimal parameter is performed. In fact, the SVC training (C=100) is implemented for ten times corresponding to the kernel parameters listed in Table 2. After each training, the testing set is then utilized to examine the generalization performance of the conventional one-against-one SVC. The results are given in detail in Table 3, in which the best results are labelled in bold and italic font.

Table 3. The SVC performance with various σ for three cases (*TA:* testing accuracy).



Circuit	*TA* (σ:0.01)	*TA* (σ:0.1)	*TA* (σ:1)	*TA* (σ:2)	*TA* (σ:4)	*TA* (σ:8)	*TA* (σ:16)	*TA* (σ:32)	*TA* (σ:64)	*TA* (σ:128)
DAmp	0.936	***0.973***	0.886	0.873	0.866	0.850	0.861	0.852	0.846	0.841
(Simulated) HPF	0.73077	0.98846	***1***	***1***	0.99808	0.99231	0.98846	0.91154	0.87115	0.83846
(Actual) HPF	0.942	***0.986***	***0.986***	0.984	0.972	0.958	0.949	0.935	0.925	0.926

It is apparent that for a certain training set, the SVC will probably have several optimal kernel parameters. For instance, in the case of actual HPF, whether the kernel parameter is 0.01 or 0.1, the SVC can achieve the best classification accuracy with 98.6%. Thus, these optimal kernel parameters form an optimal set {0.1, 1}. Our selection algorithm claims that the selected kernel parameter is 0.1, which belongs to the optimal set. These above results indicate that our proposed algorithm can select a good (or even optimal) kernel parameter.

In addition, we also compare the proposed method with several other methods in terms of testing time (*TT*) and testing accuracy (*TA*). These methods include the conventional one-against-rest SVC, the conventional one-against-one SVC, the method used in [15] and the method used in [16]. For the one-against-rest SVC in two circuits, the kernel function (RBF) parameter is 0.1 with which the best result can be achieved. The results for several methods are listed in Table 4.

Table 4. The classification performance comparison for several methods in three cases.

Circuit	Method	*TA* (×100%)	*TT* (seconds)
DAmp	One-against-rest SVC	0.952	8.322
	the method in [15]	0.945	4.824
	The method in [16]	0.969	1.5951
	One-against-one SVC	0.973	40.009
	the proposed method	0.973	6.268
(Simulated) HPF	One-against-rest SVC	1	7.5796
	the method in [15]	1	4.1189
	The method in [16]	1	1.023
	One-against-one SVC	1	37.261
	the proposed method	1	5.3645
(Actual) HPF	One-against-rest SVC	0.979	24.471
	the method in [15]	0.980	13.262
	The method in [16]	0.980	4.0337
	One-against-one SVC	0.986	90.998
	the proposed method	0.986	15.124

The results by two circuits show that the one-against-one SVC achieves the best fault classification performance, but it consumes a long testing time. Our presented method can reduce the testing time significantly, while the diagnosis performance remains unchanged. The discrepancy between the test time (i.e. *TT*) vindicates the effectiveness of the presented method.

The data in Table 2 and Table 3 indicate the presented parameter selection technique for the SVC is effective. This technique can select a good kernel parameter because it considers the BSVC classification principle and the structure of the subsequent SVC which is composed of many BSVCs. In addition, the presented method only uses simple equations to obtain the kernel parameters, avoiding exhaustive searching and thus it can save many training computations. In our experiments, we observed that the conventional method could find a proper kernel parameter at the cost of many minutes by using the n-fold ($n \geqslant 2$) cross-validation technique (In this method, the training set is split into n parts. In each training process, one part is selected as the training set, and the others are used for testing sets. Exhaustive searching within the whole parameter range is performed. Finally, the averaged testing accuracy is used to evaluate the goodness of the kernel parameter and the kernel parameter corresponding to the best accuracy is selected to be optimal). However, our presented method can use only several seconds to find a good parameter.

The testing results in our experiments illustrate that the SVC is applicable to analog fault classification. Thus, in the domain of analog circuit fault diagnosis, the SVC can be considered to replace the NN, which is a widely used fault classifier. For example, in diagnosing the simulated HPF, the testing accuracy can get a perfect 100% for all the classifiers; in diagnosing the actual HPF, the SVC in Table 4 can achieve a classification accuracy of near or above 98% with 10 training samples for each fault class. The NN in [8], however, employs 20 samples for each fault class to perform training. Each sample is also with five wavelet features and the designed two-layer feed-forward NN classifier can achieve 95% classification accuracy in diagnosing the same actual circuit.

5. Conclusions

In our research, we mainly investigate the performance of the support vector machines classifier in terms of analog fault classification technique. Conclusions can be drawn by reviewing the above results:

- The conventional one-against-one SVC needs too much time to perform fault decision work with Max-wins strategy. We improved it in this study. In our investigations, the improved classifier can achieve a very close or even identical performance to the conventional one-against-one SVC, but needs far less computational complexity to perform classification operations. Thus, the presented classifier can be used as a substitution for the conventional one-against-one SVC.
- In our kernel parameter selection algorithm, the RBF kernel function is considered because this type of kernel function can achieve good classification performance. With the proposed parameter selection algorithm, we can easily obtain a sub-optimal or even optimal parameter (*i.e.* the width of the RBF) without any additional machine learning, because the presented method is enlightened by the margin theory of SVC. In theory, a large SVC margin always leads to a good generalization capability for this classifier. This technique is simple but effective, and it is not addressed in our previous works.
- The SVC is suitable for faults classification of analog electronic circuits and it can replace the conventional NN in the domain of analog testing and diagnosis.

Future work will extend the presented technique to other SVCs. Also, how to enhance the reliability of the kernel parameter selection rule will also be envisaged.

Acknowledgements

The authors would like to thank the editor and anonymous reviewers for their useful comments.

This work was partly supported by the National Nature Science Foundation of China (grant #60871009), by the Aviation Science Foundation of China (grant #2009ZD52045), and by NUAA Research Funding of China (grant #NS2010063).

References

[1] Starzyk, J.A., Pang, J., Manetti, S., Piccirilli, M.C., Fedi, G. (2000). Finding Ambiguity Groups in Low Testability Analog Circuits. *IEEE Trans. Circuits and Syst.: Fundamental Theory and Applications*, 47(8), 1125-1135.

[2] Wang, P., Yang, S. (2005). A New Diagnosis Approach for Handling Tolerance in Analog and Mixed-Signal Circuits by Using Fuzzy Math. *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 52(10), 2118-2127.

[3] Bandler, J.W., Salama, A.E. (1985). Fault Diagnosis of Analog Circuits. *Proc. IEEE*, 73(8), 1279-1325.

[4] Golonek, T., Rutkowski, J. (2007). Genetic-Algorithm-Based Method for Optimal Analog Test Points Selection. *IEEE Trans. on Circuits and Systems-II: Express Briefs*, 54(2), 117-121.

[5] Dimopoulos, M.G., Spyronasios, A.D., Papakostas, D.K., Hatzopoulos, A.A. (2009). Wavelet Energy-based Testing Using Supply Current Measurements. *IET Sci. Meas. Technol.*, 4(2), 76-85.

[6] Catelani, M., Fort, A., Alippi, C. (2002). A Fuzzy Approach for Soft Fault Detection in Analog Circuits. *Meas.*, 32(1), 73-83.

[7] Tan, Y., He, Y. (2008). A Novel Method for Fault Diagnosis of Analog Circuits Based on WP and GPNN. *Int. J. of Electron.*, 95(5), 431-439.

[8] Aminian, M., Aminian, F., Collins, H. W. (2002). Analog Fault Diagnosis of actual Circuits Using Neural Networks. *IEEE Trans. on Instrum. Meas.*, 51(3), 1546-1554.

[9] El-Gamal, M.A., Abdulghafour, M. (2003). Fault Isolation in Analog Circuits using A Fuzzy Inference System. *Comput. Electr. Eng.*, 29(1), 213-229.

[10] Grzechca, D., Rutkowski, J. (2009). Fault Diagnosis in Analog Electronic Circuits-The SVM Approach. *Metrol. Meas. Syst.*, 16(4), 583-597.

[11] Salat, R., Osowski, S. (2003). Analog Filter Diagnosis Using Support Vector Machine. *In proc. ECCTD*, Krakow, Poland, 421-424.

[12] Siwek, K., Osowski, S., Markiewicz, T. (2006). Support Vector Machine for Fault Diagnosis in Electrical Circuits. *In proc. NORSIG*, Reykjavik, Iceland, 342-345.

[13] Sun, Y., Chen, G., Li, H. (2006). Analog Circuits Fault Diagnosis Using Support Vector Machine. *In Proc. ICCCAS*, Kitakyushu, Japan, 1003-1006.

[14] Huang, K., Stratigopoulos, H.G., Mir, S. (2010). Fault Diagnosis of Analog Circuits Based on Machine Learning. *In Proc. DATE*, Dresden, Germany, 1761-1766.

[15] Cui, J., Wang, Y. (2011). A Novel Approach of Analog Circuit Fault Diagnosis Using Support Vector Machines Classifier. *Meas.*, 44(1), 281-289.

[16] Cui, J. Wang, Y. (2010). A Novel Approach of Analog Fault Classification Using A Support Vector Machines Classifier. *Metrol. Meas. Syst.*, 17(4), 561-582.

[17] Hsu, C.W., Lin, C.J. (2002). A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Trans. on Neural Networks*, 13(2), 415-425.

[18] Vapnik, V.N. (1998). Statistical Learning Theory. New York: Wiley.

[19] Chapelle, O., Haffner, P., Vapnik, V.N. (1999). Support Vector Machines for Histogram-based Image Classification. *IEEE Trans. on Neural Networks*, 10(5), 1055-1064.