

## Research Article

# A New Adaptive LSSVR with Online Multikernel RBF Tuning to Evaluate Analog Circuit Performance

Aihua Zhang,<sup>1</sup> Chen Chen,<sup>1</sup> and Hamid Reza Karimi<sup>2</sup>

<sup>1</sup> College of Engineering, Bohai University, Jinzhou 121013, China

<sup>2</sup> Department of Engineering, Faculty of Engineering and Science, The University of Agder, 4898 Grimstad, Norway

Correspondence should be addressed to Aihua Zhang; jsxinxi\_zah@163.com

Received 6 November 2013; Accepted 29 November 2013

Academic Editor: Ming Liu

Copyright © 2013 Aihua Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Focusing on the analog circuit performance evaluation demand of fast time responding online, a novel evaluation strategy based on adaptive Least Squares Support Vector Regression (LSSVR) which employs multikernel RBF is proposed in this paper. The superiority of the multi-kernel RBF has more flexibility to the kernel function online such as the bandwidths tuning. And then the decision parameters of the kernel parameters determine the input signal to map to the feature space deduced that a well plant model by discarding redundant features. Experiment adopted the typical circuit Sallen-Key low pass filter to prove the proposed evaluation strategy via the eight performance indexes. Simulation results reveal that the testing speed together with the evaluation performance, especially the testing speed of the proposed, is superior to that of the traditional LSSVR and  $\epsilon$ -SVR, which is suitable for promotion online.

## 1. Introduction

Although many analog electronic functions have been replaced with digital equivalents, there still exists a need to use analog circuits [1] such as voice signals conversion, sensor signals microprocession, and conversion. Actually, all of the electronic circuits are not out of the analog circuits [2].

The presence of performance evaluation or detection is vital in this age of large electronic equipment that has swarmed our living. Physical damage, manufacturing technique, aging, radiation, temperature changes, and power surges are possible reasons for such performance changing. Moreover, the further state of the electronic equipment can be forecasted via performance detection, and some catastrophic errors can be avoided such as in spacecraft engineering field. The purpose of analog circuits performance evaluation is to guarantee the electronic system to be well running states before they are put into use and/or to realize the fast performance detection of the electronic system online to assure the running status. Some researchers focus on the data-driven method and lots of pieces of literature [3–6] had attempted to use it.

To this question, some researchers have focused on analog circuits fault diagnosis and performance evaluation [7]. And they are in the early stage of development, but the technique still developed slowly for complication development of electronic equipment complex. Nowadays, the normal techniques include neural network, fuzzy logic, genetic algorithm and so forth, which offer enough developed space for the analog circuit performance evaluation [8–10]. And the neural network and support vector machine (SVM) were extensive applied and researched. Aihua and Zhongdang [11] focused the promise about analog circuit performance evaluation method's portability and low cost, the support vector regression (SVR) evaluation strategy was firstly proposed, and this inherited the evaluation precision simultaneously. However, the lower convergence rate is the largest defect and this problem can also be discussed in literature [12, 13].

For taking into consideration the realization issue of superior convergence rate, Suykens and Vandewalle [14] addressed norm LSSVR method. The primary advantage of this approach is that the training process follows the structural risk minimization principle and takes the equality constraint instead of inequality constraints, and this has made

the operation speed improved greatly. The LSSVR formulation also involves less tuning parameters. However, a drawback is that sparseness is lost in the LSSVR case. Therefore some researchers has put their eyes on investigating imposing sparseness by pruning support values from the sorted support value spectrum which results from the solution to the linear system. Suykens et al. [14, 15] in their later literature present sparse approximation strategy for modifying the defect of the norm LSSVR. Although this method can realize the decremental based ascending sort with the setting threshold constraint, the training sample is still difficult to accept or reject with facing the uniformity of LSSVR spectrum. Wang et al. [16] employed a novel LSSVR algorithm in the stability space. And another LSSVR based matrix model on the linear class problem also was discussed in literature [17]. Furthermore, Zhao and Sun [18] adopted the recursion algorithm to reduce the growing data samples of LSSVR, and the sparsity solution is obtained. Theoretically speaking, the more training samples will get the higher accuracy machine learning, and this is rarely practical.

Kernel function design is the most important part of the component in LSSVR, and it is a nonlinear mapping function from the input to the feature space [19]. The main function of the kernel is to convert a linearly nonseparable classification problem in low dimension to a separable one in high dimension and hence plays a crucial role in the modeling and control performance. Kernel functions are generally parametric and the numerical values of these parameters have significant effect on both modeling and control performance. Depending on the initial values of kernel parameters some features significant for the model may be discarded or redundant or irrelevant features may also be mapped to the feature space and better performance may be achieved by discarding some feature [13, 14]. Owing to such factors, the selection of optimal kernel parameters is vital in terms of the solution of the SVR problem. There are lots of optimization methods on the kernel parameters, such as particle swarm optimization, pattern search, and grid search [20, 21]. The goal mostly located in offline calculation kernel parameters. Literature [22] used gradient optimization method to realize the single RBF kernel function (SRBF) online adjustment variance.

This work, researched on the literature [15, 23], presents an analog circuit evaluation strategy based LSSVR which also treats the circuit and signal online but adopts multikernel RBF to realize the adjustment of kernel width, which not only contribute the regression of LSSVR, but also improve the evaluation speed greatly.

Focusing on the analog circuit performance evaluation demand of fast time responding online, a novel evaluation strategy based on adaptive Least Squares Support Vector Regression (LSSVR) which employs multikernel RBF is proposed. The superiority of the multikernel RBF has more flexibility to the kernel function online such as the bandwidths tuning. And then the decision parameters of the kernel parameters determine the input signal to map to the feature space that deduced a well plant model by discarding redundant features.

## 2. Evaluation Algorithm

**2.1. Support Vector Regression.** Support vector machine (SVM) is originally developed by Vapnik [24] for solving classification problems, and it has also been studied extensively for the solution of regression problems. Meanwhile, the superior [25] has been revealed via the structural risk minimization principle of SVM which employed by conventional neural networks. SVM also has a greater ability to generalize, which is the important task in statistical learning. SVR is the extension of SVMs to solve regression to minimize the generalized error bound so as to achieve generalized performance. When using SVM in regression tasks, the SVR must use a cost function to measure the empirical risk in order to minimize the regression error. The brief details about SVR are presented as follow.

Consider the learning sample for SVR,  $S = \{s_i \mid s_i = (x_i, y_i)\}$ , where  $x_i \in R^n$  is a vector representing a set of sample inputs at a certain instant  $i$  and  $y_i \in R$  is a vector representing the corresponding a set of sample outputs. This purpose is to find a function which can estimate output data in a better way.

**2.1.1. Linear SVR.** Consider

$$y = \omega \cdot x + b, \quad (1)$$

where “ $\cdot$ ” denotes the inner product,  $\omega$  and  $b$  are the parameters of the function, and  $x$  is the test pattern in a normalized form. The structural risk minimization principle can be realized by minimizing the empirical risk  $R_{\text{emp}}(\omega, b)$  defined by

$$R_{\text{emp}}(\omega, b) = \frac{1}{n} \sum_{i=1}^n L_e(y_i, f(x)), \quad (2)$$

where  $L_e(y_i, f(x))$  denotes error-insensitive loss function of the empirical risk, and it can be defined by

$$L_e(y_i, f(x)) = \begin{cases} e, & \text{if } |y_i - f(x)| \leq e, \\ |y_i, f(x)| - e, & \text{otherwise.} \end{cases} \quad (3)$$

$L_e(y_i, f(x))$  is the insensitive loss function; that is to say, it is the tolerance error between the target output  $y_i$  and the estimated output values  $f(x)$  in optimization process, and  $x_i$  is a training pattern. The problem of finding  $\omega$  and  $b$  to reduce the empirical risk with respect to an insensitive loss function is equivalent to the convex optimization problem that minimizes the margin  $\omega$  and slack variables  $(e_i, e_i)$  as

$$\begin{aligned} \min \quad & J(\omega, e) = \frac{1}{2} \omega^T \cdot \omega + \frac{1}{2} \gamma \sum_{i=1}^n e_i^2 \\ \text{s.t.} \quad & y_i - \omega \cdot x_i - b \leq e_i, \\ & i = 1, \dots, n, \end{aligned} \quad (4)$$

where the first term  $(1/2)\omega^T \cdot \omega$  is the margin; the parameter  $\gamma$  is a positive constant. To solve the above optimization

problem, one has to find a saddle point of the Lagrange function described as [26]

$$L_\varepsilon(\omega, b, e, a) = \frac{1}{2}\omega^T \cdot \omega + \frac{1}{2}\gamma \sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i (\omega^T x_i + b + e_i - y_i). \quad (5)$$

**2.1.2. Nonlinear SVR.** In fact, the linear SVR is not for all the real system because of the problem complex of the real world. The nonlinear SVR is as an alternative for linear SVR that has appeared. The input data sample  $x_i$  is transformed into feature space  $\varphi(x_i)$  by a nonlinear function [12]. Then, the same optimization algorithm is applied in the same way as the linear SVR. Therefore, the nonlinear function of SVR can be expressed by

$$y_i = \omega \cdot \varphi(x_i) + b + e_i, \quad (6)$$

where “ $\cdot$ ” denotes the inner product,  $\omega, b$  are the parameters of the function, and  $\varphi(x_i)$  is the mapping function from the input feature to a higher dimensional feature space.

For the regression problem of the given training set  $S = \{s_1, \dots, s_n\}$ , the classical SVR model [15] can be obtained from the following optimization problem:

$$\begin{aligned} \min \quad & J(\omega, e) = \frac{1}{2}\|\omega\|^2 + \frac{1}{2}\gamma \sum_{i=1}^n e_i^2 \\ \text{s.t.} \quad & y_i = \omega \cdot \varphi(x_i) + b + e_i \\ & i = 1, \dots, n, \end{aligned} \quad (7)$$

where  $e = (e_1, e_2, \dots, e_n)^T$  is the evaluated error of sample. For the purpose to get the evaluated formulation just like (6) from the optimization problem (1) to realize the evaluation and diagnosis for the future samples, then the optimization problem (7) with the employed Lagrange multipliers and matrix in variable method can be rewritten as

$$\begin{bmatrix} 0 & 1^T \\ 1 & A \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (8)$$

where  $\Omega_{ij} = k(x_i, x_j)$ ,  $A \equiv \Omega + (1/\gamma)I$  is the correlation matrix,  $1 = [1, \dots, 1]^T$  and  $a = [a_1, \dots, a_n]^T$  are Lagrange multipliers,  $y = [y_1, \dots, y_n]^T$  is the output vector, and  $k(\cdot, \cdot)$  is RBF kernel function which will be stated alone in the next section. The key point of solving (8) is to confirm the inverse matrix  $A^{-1}$ , once a new sample  $s_{n+1} = (x_{n+1}, y_{n+1})$  joins in the training set, we can get the predictor  $A$ , namely, SVR, as follows:

$$A_{n+1} = \begin{bmatrix} A_n & b_1 \\ b_2 & c \end{bmatrix}, \quad (9)$$

where  $A_n, A_{n+1}$  are kernel correlation matrixes of training sets  $S \cup \{s_{n+1}\}$ ,  $b_2 = (\Omega_{n+1,1}, \dots, \Omega_{n+1,n})$ ,  $b_1 = b_2^T$ , and  $c = \Omega_{n+1,n+1}$ . Once  $A_{n+1}^{-1}$  can be obtained via  $A_n^{-1}$ , the training mission of incremental SVR is done [27]. As for LSSVR, to

solve  $A_{n-1}^{-1}$  under the knowing  $A_n = (a_{ij}), A_n^{-1} = (\tilde{a}_{ij}), A_{n-1} = (a_{ij})_{i,j \neq k}$ , namely, for a given sample set, adopting inverse training algorithm. Here, we adopt strategy to remove the  $k$ th line with the  $k$ th list of  $A_n$  to eliminate part sample to get  $A_{n-1}$ . Via the algorithm of reduced order and inversion [28], cause  $A_{n-1}^{-1} = (\tilde{a}_{ij})_{i,j \neq k}$ , then the reduced order formulation can be achieved

$$\hat{a}_{i,j} = \tilde{a}_{ij} - \frac{1}{a_{kk}} \tilde{a}_{ik} \tilde{a}_{kj}, \quad i, j \neq k. \quad (10)$$

**2.2. Multikernel RBF Adjust Strategy.** For realizing the flexibility to the kernel, in this part, we modify the kernel RBF which utilizes the linear combinations of RBF kernels. The multikernel RBF is addressed as follows:

$$\ker(x_i, x_j) = \frac{\sum_{m=1}^p k_m \exp(-d_{i,j}/2\sigma_m^2)}{\sum_{z=1}^p k_z}, \quad (11)$$

where  $\sigma_m$  is the bandwidth of the kernel function,  $x_i$  is the current state vector of the plant,  $x_j$  is the test data samples, and  $d_{i,j}(n)$  is the Euclidean distance between current data which is expressed by

$$d_{i,j} = (x_i - x_j)^T (x_i - x_j). \quad (12)$$

To guarantee the fast responding, such as the computation speed, we adopt multikernel RBF which is expressed by

$$\begin{aligned} \text{Ker}(x_i, x_j) = & \frac{k_1}{k_1 + k_2 + \dots + k_m} K_1 + \frac{k_2}{k_1 + k_2 + \dots + k_m} K_2 \\ & + \dots + \frac{k_m}{k_1 + k_2 + \dots + k_{m2}} K_m, \end{aligned} \quad (13)$$

where

$$K_m = K(x_i, x_j, \sigma_j) = \exp\left(-\frac{d_{i,j}}{2\sigma_j^2}\right). \quad (14)$$

To verify the superior performance of the multikernel RBF, we also employ LSSVR with the norm kernel RBF in this paper. Firstly, the multikernel RBF was fixed bandwidths which is equivalent to saying that the norm kernel RBF with varying bandwidth depending on scaling coefficients and Euclidean distance between features, namely, the multikernel RBF, has the better flexibility to the unknown problem. Then the LSSVR function can be rewritten as follows:

$$\hat{y} = \sum_{i=n-L}^{n-1} a_i \ker(x_i, x_j) + b, \quad (15)$$

where  $L$  is Lagrange multiplier expressed by

$$\begin{aligned} L(\omega, b, e, a) = & \frac{1}{2}\|\omega\|^2 + \frac{1}{2}\gamma \sum_{i=1}^N e_i^2 \\ & - \sum_{i=1}^N a_i (\omega \varphi(x_i) + b + e_i - y_i). \end{aligned} \quad (16)$$

Partial derivatives of LSSVR model with respect to weights and bandwidths of the kernels are obtained as follows:

$$\frac{\partial \hat{y}}{\partial k_m} = \sum_{i=n-L}^{n-1} a_i \left[ \frac{\sum_{z=1}^p k_z (K_m - K_z)}{[\sum_{z=1}^p k_z]^2} \right], \quad (17)$$

$$\frac{\partial \hat{y}}{\partial \sigma_j} = \sum_{i=n-L}^{n-1} a_i \frac{k_m}{\sum_{z=1}^p k_p} K_m \frac{a_{i,j}^2}{\sigma_j^3}.$$

Then, the objective function to be minimized for improving the LSSVR model performance is chosen as follows:

$$J = \frac{1}{2} [y - \hat{y}]^2 = \frac{1}{2} [\hat{e}^2]. \quad (18)$$

The kernel width  $\Delta\sigma_j$  and scaling coefficients  $\Delta k_j$  can be adjusted via the method proposed in literature [29]:

$$\Delta\sigma_j = -\eta \frac{\partial J_p}{\partial \hat{e}_p} \frac{\partial \hat{e}_p}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \sigma_j} = \eta \hat{e}_p \frac{\partial \hat{y}}{\partial \sigma_j}, \quad (19)$$

$$\Delta k_j = -\eta \frac{\partial J_p}{\partial \hat{e}_p} \frac{\partial \hat{e}_p}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial k_j} = \eta \hat{e}_p \frac{\partial \hat{y}}{\partial k_j},$$

where  $\eta$  ( $0 < \eta < 1$ ) is the learning rate obtained by any line search algorithm. So, the kernel parameters can be adjusted as

$$\sigma_j(n+1) = \sigma_j(n) + \Delta\sigma_j(n), \quad (20)$$

$$k_j(n+1) = k_j(n) + \Delta k_j(n).$$

**2.3. Algorithm of the Multikernel Adaptive LSSVR.** Aiming the training set  $S$  that is given in Section 2.2, then the regression function is expressed by

$$f(x, a, b) = \sum_{i \in W} a_i \ker(x_i, x_j) + b \equiv f(x)|_{\bar{W}}, \quad (21)$$

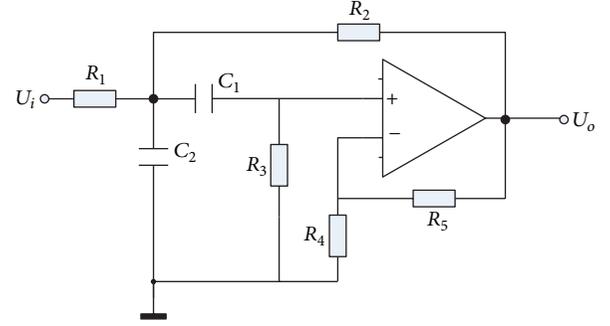
where  $a_i$ ,  $b$  are the regression parameters,  $W$  is training sample working set, and  $\bar{W}$  is the regression parameter set of  $W$ .

In this paper, multikernel RBF LSSVR algorithm includes initialization and adaptive update the design and procedure as follows [23].

### 2.3.1. Initialization

*Step 1.* Make  $W = \{s_1, s_2\}$ , and  $A^{-1}$ ,  $\bar{W}$  can be confirmed by set (8).

*Step 2.* If  $k = 3, \dots, N$ , the regression function  $f(x)|_{\bar{W}}$  can be detected by the sample  $s_k$ ; if  $|f(x_k)|_{\bar{W}} - y_k| > \theta$ , then  $W = W \cup \{s_k\}$ , and  $\bar{W}$  should be recomputed via increment algorithm, confirming least support vector spectrum  $|a_i^*| = \min_{s_i \in W} \{|a_i|\}$ , constructing temporary training set  $\bar{W} = \bar{W} \setminus \{s_i^*\}$ , utilizing inverse training algorithm computing  $(\bar{W})$  via



$$(R_1 = 1 \text{ k}\Omega, R_2 = 3 \text{ k}\Omega, R_3 = 2 \text{ k}\Omega, R_4 = R_5 = 4 \text{ k}\Omega, C_1 = C_2 = 5 \text{ nF})$$

FIGURE 1: Sallen-Key low pass filter.

$\bar{W}$ , and using the regression function  $f(x) |_{\bar{W}}$  which can be detected by the sample  $s_{k+1}$ . If  $|f(x_k) |_{\bar{W}} - y_k| \leq \theta$ , in that way  $W = \bar{W}$ ,  $\bar{W} = (\bar{W})$ .

*Step 3.* Compute the value  $J(\omega, e)|_W$  of the working set objective function.

*Note 1.* The objective function is

$$\min J(\omega, e) = \frac{1}{2} \|\omega\|_{\omega \in \bar{W}}^T + \frac{1}{2} \gamma \sum_{s_i \in W} e_i^2. \quad (22)$$

### 2.3.2. Adaptive Update

*Step 1.* If  $W = N$ , then the output regression function is exported, otherwise turn to Step 2. If  $|J_C - J_L|/J_C > \varepsilon$ , simultaneously, if  $s_k \notin W$  and  $f(x_k)|_{\bar{W}} > \theta$ , then  $W = W \cup \{s_k\}$ ,  $|a_i^*| = \min_{s_i \in W} \{|a_i|\}$ ,  $\bar{W} = W \setminus \{s_i^*\}$  and  $\bar{W}$  is computed again via increment algorithm.

*Step 2.* Compute the value  $J(\omega, e)|_W$  of the working set objective function.

*Note 2.*  $J_C$  is objective function that has been updated in this time;  $J_L$  is the objective function that has been updated in the last time.

*Note 3.* The forecast training accuracy and test precision are set to be  $\theta$  and algorithm stop parameter set to be  $\varepsilon$ .

*2.3.3. Termination Judgment.* If  $|J_C - J_L|/J_C \leq \varepsilon$ , then the training is stopped.

## 3. Simulation

*3.1. Prepare before Simulation.* The CUT in this paper is a typical circuit Sallen-Key low pass filter as shown in Figure 1 [30]. The evaluating indicator for performance includes eight indexes: gain, transmission band, cutoff frequency, lower cut-off frequency, maximum undistorted output amplitude, maximum undistorted power output, input sensitivity, and

TABLE 1: Result of data feature and comparative experiment of regression problem on experiment data.

TRSN	TESN	FN	Method	Parameter ( $\sigma, e, \gamma$ )	SVN	TRMSE	TEMSE	CPU/s
$259 \times 100$	$59 \times 100$	8	MKALSSVR	(200, 0.1, 0.75)	30	$1.6982e - 014$	$1.7001e - 028$	0.021
$259 \times 100$	$59 \times 100$	8	LSSVR	(200, 0.1, /)	1376	$4.5544e - 015$	$4.2437e - 012$	2.981
$259 \times 100$	$59 \times 100$	8	$\varepsilon$ -SVR	(200, 0.1, /)	1607	$5.4003e - 017$	$5.3025e - 004$	0.062

noise voltage. Then to confirm training set based on the eight indexes, we first define sample point  $(x, y)$  and correspondingly obtain training set  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\}$ .

3.2. *Data Selection and Standardized Processing.* Experiment adopted the typical circuit Sallen-Key low pass filter to prove the proposed evaluation strategy via the eight performance indexes which obtained by precise instrument evaluation in two years. The sample number is  $259 \times 100$ , record data set  $R$ . Before verifying the proposed method in this paper, the first thing to be done is to establish data sets of training and testing. However, the strangeness value in the data set caused by human record and other noncircuit fault factors will make great effects to model performance of LSSVR, especially the data set including the strangeness value that are used for modeling. Hence, a normalization of the data is required before presenting the input patterns to any statistical machine learning algorithm. In this experiment, 0-1 normalization method, denoted by (23), is utilized to preprocess:

$$x_i^n = \frac{x_i^a - x_i^{\min}}{x_i^{\max} - x_i^{\min}}, \quad (23)$$

where  $x_i^a$  and  $x_i^n$  are the  $i$ th components of the input vector before and after normalization, respectively, and  $x_i^{\max}$  and  $x_i^{\min}$  are the maximum and minimum values of all the components of the input vector before the normalization. Completing data processing via 0-1 normalization method, the noise has been reduced obviously.

After the above data selection and data normalization,  $200 \times 100$  samples are selected randomly to be the training samples; the rest data samples are to be a test sample. To validate the superior evaluation performance of the proposed MKALSSVR to evaluate the analog circuit performance online, the different methods such as LSSVR,  $\varepsilon$ -SVR, and the precision instrument are also carried out for the comparison purpose while the analog circuit performance evaluation is on. Meanwhile several parameters need to be introduced before applying the three SVR algorithms. First of all, it is required to denote three parameters, namely, error insensitive zone ( $\varepsilon$ ), penalty factor  $\gamma$ , and kernel specific parameters  $\sigma$ . Problem regarding the choice of  $e$ ,  $\gamma$ , and  $\sigma$  was studied by several researchers [31, 32]. The penalty factor  $\gamma$  controls the smoothness or flatness of the approximation function. If we set the value  $\gamma$  to be large, the objective is only to minimize the empirical risk, which makes the learning machine more complex. On the contrary, if we set the value  $\gamma$  to be small, the objective is to cause the errors to be excessively tolerated yielding a learning machine with poor approximation [33]. In this study, SVR models have been constructed with  $\gamma$  and  $e$  varied starting from  $\gamma = 10$  and  $e = 0.004$  which are the

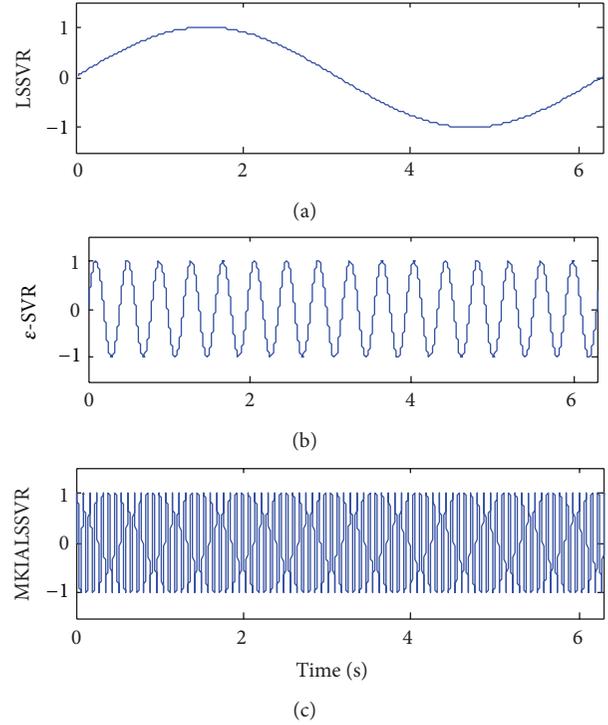


FIGURE 2: Time response of the three method comparison.

empirical values given by [33]. Via some testing, the parameters  $\gamma$  and  $e$  have been varied over a specific corresponding range in order to obtain better coefficient of correlation value, and the correlation value, denoted  $Re$ , is determined by (24). The kernel specific parameters  $\sigma$  are restricted since the value shown in Table 1 gives the better prediction for these models. The three values for each model are shown in Table 1. This study adopts RBF (11), where  $\sigma$  is width of RBF; this is also known as kernel function. The adopted  $\gamma$ ,  $e$ , and  $\sigma$  values for the four models are shown in following Table 1:

$$Re = \frac{\sum_{i=1}^n (D_{ai} - \bar{D}_a)(D_{pi} - \bar{D}_p)}{\sqrt{\sum_{i=1}^n (D_{ai} - \bar{D}_a)^2} \sqrt{\sum_{i=1}^n (D_{pi} - \bar{D}_p)^2}}, \quad (24)$$

where  $D_{ai}$  and  $D_{pi}$  are the actual and predicted values, respectively;  $\bar{D}_a$  and  $\bar{D}_p$  are mean of actual and predicted  $D$  values corresponding to  $n$  patterns. The number of support vector (SVN), the number of testing support vector (TESN), the number of training support vector (TRSN), the number of the data feature (FN), testing data mean square error (TEMSE), and training data mean square error (TDMSE) are all shown in Table 1. And  $MSE = \sqrt{(1/K) \sum_{i=1}^K (X_i - \bar{X}_i)^2}$ , where

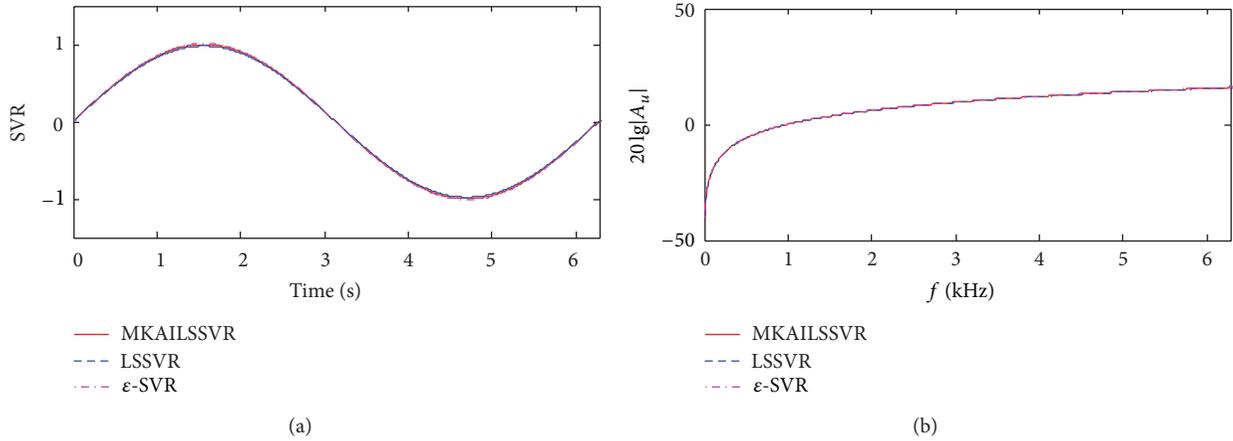


FIGURE 3: Local regression curve of output  $U_O$  and gain  $20 \lg|A_u|$  with the three methods.

TABLE 2: Result of comparative assessment.

Method	$F$							
	$20 \lg A_u $	$f_{BW}$ (KHz)	$f_L$ (KHz)	$f_H$ (KHz)	$U_{om}$ (V)	$P_{om}$ (W)	$U_S$ (mV)	$U_N$ (mV)
MKALSSVR	40.020	10.000	0.004	10.002	61.541	473.412	61.540	0.855
LSSVR	40.029	9.998	0.004	10.005	61.540	473.413	61.540	0.855
$\varepsilon$ -SVR	40.027	10.000	0.004	10.004	61.540	473.412	61.540	0.857
Instrument	40.020	10.001	0.004	10.005	61.541	473.412	61.541	0.854

$X_i$  is the real value,  $\bar{X}_i$  is the predicted value, and  $K$  is a testing sample number.

**3.3. Simulation Experiment.** To validate the superior evaluation performance of the proposed MKALSSVR, the other two different methods, LSSVR and  $\varepsilon$ -SVR, are also employed in this part. The sharp contrast about the time response of the three methods are presented in Figure 2. We take one period testing time of LSSVR as comparison and giving the other two methods testing time, respectively. Via this testing comparing, we can see clearly that the testing speed is superior greater than the other two methods. In Figure 3, we can see that the support vector density is closely bound up the curvature. If the curvature is bigger; the support vector density is also bigger, on the contrary, while in the position of the relatively smooth, the support vector density is relatively small.

For the same purpose above, Tables 1 and 2 all give out the same things to prove the evaluation precision and speed via the proposed method MKALSSVR. And for the purpose to prove the well performance of the evaluation, the precise instrument method is utilized.

## 4. Conclusion

In this paper, a novel online evaluation strategy MKALSSVR aimed to analog circuit. Via numerical simulation, we can draw a conclusion that the proposed MKALSSVR has the merit as follows: first, the adaptive training strategy can confirm the training sample number adaptively; second, the multikernel design has changed the RBF width and having

the more flexible adjust ability. And this makes the evaluation have the online processing ability. Third, this method avoids the overflow problem of norm LSSVR and support vector sparsity. Meanwhile, considering the low cost, high evaluation precision, and high operation rate of the proposed method MKALSSVR, this strategy is worth to be developed and implemented. Based on this discussion, we will take the issue about how to deal with the fault value as the future research problem.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This present work was supported partially by National Natural Science Foundation of China (Project no: 61304149) and Natural Science Foundation of Liaoning, China (Project no: 2013020044). The authors highly appreciate the above financial supports.

## References

- [1] P. Kabisatpathy, A. Barua, and S. Sinha, *Fault Diagnosis of Analog Integrated Circuits*, vol. 30, Springer, 2005.
- [2] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 109–128, 1997.

- [3] S. Yin, S. X. Ding, A. H. A. Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.
- [4] S. Yin, S. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.
- [5] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.
- [6] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2013.
- [7] H. Lin, L. Zhang, D. Ren, H. Kang, and G. Gu, "Fault diagnosis in nonlinear analog circuit based on Wiener kernel and BP neural network," *Chinese Journal of Scientific Instrument*, vol. 30, no. 9, pp. 1946–1949, 2009.
- [8] D. Sánchez, P. Melin, O. Castillo, and F. Valdez, "Modular neural networks optimization with hierarchical genetic algorithms with fuzzy response integration for pattern recognition," in *Advances in Computational Intelligence*, pp. 247–258, Springer, 2013.
- [9] S. Abdulla and M. Tokhi, "Fuzzy logic based FES driven cycling by stimulating single muscle group," in *Converging Clinical and Engineering Research on Neurorehabilitation*, pp. 173–182, Springer, 2013.
- [10] C. W. Chen, P. C. Chen, and W. L. Chiang, "Modified intelligent genetic algorithm-based adaptive neural network control for uncertain structural systems," *Journal of Vibration and Control*, vol. 19, no. 9, pp. 1333–1347, 2013.
- [11] Z. Aihua and Y. Zhongdang, "Research on amplifier performance evaluation based on support vector regression machine," *Chinese Journal of Scientific Instrument*, vol. 29, no. 3, pp. 618–622, 2008.
- [12] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [13] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1188–1193, 2000.
- [14] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [15] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1, pp. 85–105, 2002.
- [16] L. Wang, L. F. Bo, F. Liu, and L. C. Jiao, "Least squares hidden space support vector machines," *Chinese Journal of Computers*, vol. 28, no. 8, pp. 1302–1307, 2005.
- [17] Z. Wang and S. Chen, "New least squares support vector machines based on matrix patterns," *Neural Processing Letters*, vol. 26, no. 1, pp. 41–56, 2007.
- [18] Y. Zhao and J. Sun, "Recursive reduced least squares support vector regression," *Pattern Recognition*, vol. 42, no. 5, pp. 837–842, 2009.
- [19] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.
- [20] Y. C. Guo, "An integrated PSO for parameter determination and feature selection of SVR and its application in STLF," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 359–364, IEEE, July 2009.
- [21] M. Momma and K. P. Bennett, "A pattern search method for model selection of support vector regression," in *Proceedings of the SIAM Conference on Data Mining (SDM '02)*, 2002.
- [22] K. Uçak and G. Öke, "Adaptive PID controller based on online LSSVR with kernel tuning," in *Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications (INISTA '11)*, pp. 241–247, IEEE, June 2011.
- [23] K. Ucak and G. Oke, "An improved adaptive PID controller based on online LSSVR with multi RBF kernel tuning," in *Adaptive and Intelligent Systems*, pp. 40–51, Springer, 2011.
- [24] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2nd edition, 2000.
- [25] S. R. Gunn, "Support vector machines for classification and regression," ISIS Technical Report, 1998.
- [26] S. Rajasekaran, S. Gayathri, and T. L. Lee, "Support vector regression methodology for storm surge predictions," *Ocean Engineering*, vol. 35, no. 16, pp. 1578–1587, 2008.
- [27] Z. Liang and Y. Li, "Incremental support vector machine learning in the primal and applications," *Neurocomputing*, vol. 72, no. 10–12, pp. 2249–2258, 2009.
- [28] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, pp. 409–415, 2001.
- [29] D. G. Luenberger, *Linear and Nonlinear Programming*, Kluwer Academic Publishers, Boston, Mass, USA, Second edition, 2003.
- [30] M. Aminian and F. Aminian, "Neural-network based analog-circuit fault diagnosis using wavelet transform as preprocessor," *IEEE Transactions on Circuits and Systems II*, vol. 47, no. 2, pp. 151–156, 2000.
- [31] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, Wiley, 2007.
- [32] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.
- [33] P. S. Yu, S. T. Chen, and I. F. Chang, "Support vector regression for real-time flood stage forecasting," *Journal of Hydrology*, vol. 328, no. 3–4, pp. 704–716, 2006.