# Improved Support Vector Machine Regression for Analog Circuits' Macromodeling using Efficient Kernel Functions

D. Boolchandani* Mohammad Abrar Ahmed † Vineet Sahula ‡

## Abstract

***Support Vector Machines (SVM) have been used for creating fast and efficient models for quickly predicting the performance parameters of analog circuits. These models have proved to be not only effective and fast but accurate also for performance estimation. Kernels are an integral part of SVM to obtain an optimized and accurate model. The most commonly used kernels are RBF, polynomial, spline, multilayer perceptron. In this paper, many other kernels are explored, some of which are less popular, while others are modified compositions of the some of the standard kernels. These kernel functions are tested on different analog circuits to check for their robustness and for improved performance they provide. SPICE has been used for generation of learning data. Least Square SVM toolbox interfaced with MATLAB was used for regression. Models containing modified compositions of kernels were found to be more accurate and having lower mean square error than those containing standard kernels.***

**Presenting Author's Biography**

D. Boolchandani has 19 years of teaching experience and is currently an Assistant Professor in the Department of Electronics & Communication Engineering at Malaviya National Institute of Technology Jaipur, India. He obtained Bachelor's degree in Electronics Engineering from Malaviya National Institute of Technology Jaipur, Master's degree in Electronic Design Technology from Indian Institute of Science Bangalore, India. He is currently pursuing Ph.D. at Malaviya National Institute of Technology Jaipur, India, in the field of Analog Synthesis and macromodeling. He has guided more than 10 Master's theses and several undergraduate projects. He is currently memebr of IEEE USA, IETE India and life member of ISTE India.

* *Member IEEE* & Assitant Professor, Dept. of ECE, Malaviya National Institute of Technology Jaipur-302017 INDIA. Email: dbool6@gmail.com

† Malaviya National Institute of Technology Jaipur-302017 INDIA, Email: abraruday@gmail.com

‡ *Semior member, IEEE* & Associate Professor, Dept. of ECE, Malaviya National Institute of Technology Jaipur-302017 INDIA. Email: sahula@ieee.org

# 1. Introduction

An analog system is typically characterized by a set of performance parameters used to quantify the properties of the circuit. During analog synthesis, macromodel of an analog circuit helps in efficient design space exploration to obtain optimally sized circuit. Given a fixed topology, circuit sizing is the process of determining numerical values for all components in the circuit such that the circuit conforms to a set of performance constraints. Performance parameters of various design instances needs to be evaluated to reach a suitable solution. Generally, SPICE is used to obtain performance parameter from circuit simulation, however it is quite time consuming. An efficient and faster way is to use macromodel, which approximates the relationship between the device sizes and performance parameters. Support vector machine (SVM) regression offers solution for such performance macromodelling. The SVM models can be trained using data generated directly from SPICE. These SVM models are build around suitable kernel functions as regression fucntions and are able to provide SPICE level accuracy. Extraction of data for use with support vector machine is although expensive yet affordable, as it is a one time cost per topology. Once the models are developed, execution times for performance evaluation are very small, leading to a considerable reduction in synthesis time. While directly employing SPICE during synthesis, any topology can be readily handled. Thus Support vector machines require an extraction step which is specific to each topology. SVMs are black box models and are unable to reveal even qualitative aspects of system behavior. However, they can be used to readily model hyperdimensional and nonlinear functionality. Support vector machines are typically trainied with a discrete set of data points called training data set. A second set of discrete data points, not present in the training data set, is used to validate the SVM model of the system.

The rest of the paper is organized as follows. We discuss previous work reported in literature in Section 2. We present our proposal for efficient kernel functions and experimental setup for model validation in Section 3. Results are presented in Section 4. We conclude in Section 5.

# 2. Previous Work

## 2.1. $\varepsilon$-Support vector regression

Our work is based on the theory from [1] [2]. Suppose we are given a training data $\{(x_1, y_1), ...(x_l, y_l)\} \subset R^N \times R$, where $R^N$ represents input space. By a certain nonlinear mapping $\phi$, the training pattern $x_t$ is mapped into some feature space, in which a linear function is defined as follows.

$$f(x) = < \omega, \phi(x) > +b \text{ with } \omega \in R^N, b \in R \tag{1}$$

In $\varepsilon$-SVR, the goal is to find a function $f(x)$ that has at most $\varepsilon$ deviation from the actually obtained targets $y_i$ for all the training data, and at the same time, is as flat as possible. One way to ensure this is by requiring:

$$\text{minimize } \frac{1}{2} \parallel \omega \parallel^2 + C \sum_{i=1}^{l} (\xi + \xi^*)$$
$$\text{subject to } \begin{cases} y_i - < \omega, \phi(x_i) > -b \le \varepsilon + \xi \\ < \omega, \phi(x_i) > +b - y_i \le \varepsilon + \xi^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{2}$$

The constant $C > 0$ determines the trade off between the flatness and the amount up to which deviations larger than $\epsilon$ are tolerated. By using Lagrange multiplier techniques and $k(x, x') =< \phi(x), \phi(x') >$ instead of $\phi(.)$ explicitly, the optimization problem of (2) leads to the following dual optimization problem.

$$\text{maximize } \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j)$$
$$-\varepsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i (\alpha_i - \alpha_i^*) \tag{3}$$
$$\text{subject to } \begin{cases} \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

The decision function takes the form

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(x_i, x) + b \tag{4}$$

The function $k(x, x')$corresponds to a dot product in some feature space.

## 2.2. Mercer kernel

The Mercer theorem provides the conditions to be a support vector kernel [1, 2].

**Theorem 1** Suppose k $\varepsilon L_\infty$ $(R^N \times R^N)$ such that the integral operator $T_k : L_2(R^N) \to T_k : L_2(R^N)$

$$T_k f(.) := \int_{R^N} k(., x) f(x) d\mu(x) \tag{5}$$

is positive. Let $\psi_j \in L_2(R^N)$ be the eigen function of $T_k$ associated with eigenvalue $\lambda_j \neq 0$ and normalized such that $\parallel \psi_j \parallel_{L_2} = 1$ and let $\bar{\psi_j}$ denote the complex conjugate. Then

1. $(\lambda_j(T_k))_j \in l_i$
2. $\psi_j \in L_\infty(R^N)$ and $sup_j \parallel \psi_j \parallel_{L_\infty} \leq \infty$
3. $k(x, x') = \sum_j \lambda_j \bar{\psi_j}(x)\psi_j(x')$

holds for almost all$(x, x')$, where the series converges absolutely and uniformly for almost all $(x, x')$.

A kernel satisfying the conditions of this theorem is called a Mercer kernel. This theorem means that if

$$\int_{R^N \times R^N} k(x, x')f(x)f(x')dxdx' \geq 0 \text{ for all } f \in L_2(x)$$

holds $k(x, x')$can be written as a dot product in some feature space. From this condition the simple rules for composition of kernels can be concluded, which also satisfy Mercer's condition.

**Corollary 1** (Linear combinations of kernels): Let $k_1(x, x'), k_2(x, x')$ be Mercer kernels and $c_1, c_2 \geq 0$,then

$$k(x, x') = c_1 k_1(x, x') + c_2 k_2(x, x') \tag{6}$$

is also called a Mercer kernel. Moreover, the product of two Mercer kernels is a Mercer kernel, which can be proven on the basis of the equivalent definition of Mercer kernel. This proof can be seen in [3]. Similarly, it has been proposed earlier in [5] that we can modify the kernel functions by multiplying it by a positive factor, adding bias, or taking exponential of the kernel. The new kernel obtained is also a Mercer Kernel. Mercer condition needs to be satisfied for keeping the problem convex and hence obtaining a unique solution. These important modifications are as follows.

$$k(x, x') = \alpha k(x, x') \text{ where } \alpha > 0 \tag{7}$$

$$k(x, x') = a * exp(k(x, x')) \text{ where } a > 0 \tag{8}$$

We also discuss about two kernels here, as they have procided with good results for analog design- Power kernel [7] given by:

$$k(x, x') = - \parallel x - x' \parallel^\beta \tag{9}$$

and Log kernel [8] given by:

$$k(x, x') = -log(1+ \parallel x - x' \parallel^\beta) \tag{10}$$

where the kernels are conditionally positive definit for $0 < \beta \leq 1$. All the kernels discussed above, satisfy the Mercer's condition, which is a necessary condition for the problem to be convex, and hence giving a unique and optimal solution. We compare proposed composite kernels with RBF kernel all along the further discussion. as we have apply modifications on RBF kernel. Similar modifications can be carried out for other standard kernels, based on the application. An RBF kernel is given by:

$$K(x, x_j) = e^{-\gamma|x - x_i|^2} \tag{11}$$

### 2.3. Related Work

Performance macromodeling mainly falls into three categories- knowledge based approaches, symbolic analysis and various regression techniques. Knowledge based methods [11] rely on manual derivation of mathematical equations by expert designers. Symbolic analysis methods generate these equation automatically. However, in both cases the performance parameters are not necessarily the direct function of controllable design parameters, rather they are dependant on the small signal parameters. Regression techniques have gained more and more research interest lately, as they use the least amount of knowledge about the circuit topology and are therefore more general. Examples include various neural networks [15], fitting approach to generate symbolic equations [20] and least squares support vector machines [19]. Performance feasibility models are useful in topology selection. Examples of these work are shown in [18].

Log and Power kernels have been discussed in [8], where it is shown that these kernels satisfy the Mercer's condition. The authors in [8] have compared these kernels with RBF and Laplace kernels for image recognition problem. The log and power kernels have provided with good results for pattern classification problems. The modification by composing or mixing of the kernels have been discussed in [9, 10]. These modifications were tried out for regression problems, and were found to be more accurate. These modifications have also resulted in improved performance.

## 3. Proposed Work

In this section we compare the kernels described in the Section 2 alongwith the standard kernels like RBF and polynomial. The modification shown in eq. (7) was done on RBF kernel and eq. (8) represents the modification done on polynomial kernel. RBF and polynomial kernels are chosen for modifications as these are most commonly used kernels. The circuits used in the procedure are described in the next section. The support vector machine model was trained using the data generated from HSpice. Previous section describes the parameters considered for optimization by the model constructed with different kernels for specified performance parameter.

Least Square Support Vector Machine Toolbox [4] interfaced with MATLAB was used for function estimation. Toolbox was trained using the data generated from HSpice. The toolbox provides the values of optimized $\alpha$ and bias as output. These values can be use to estimate the function using eq. (4). As we can see in eq. (4) the kernel has an important role to play in function estimation. The toolbox was trained using RBF, Multiplied RBF, Bias RBF, Log and Power kernels. The model generated was then verified using test data generated from HSpice. The basis of comparison of the trained model are mean square error which is the deviation from Spice and the computation time.

## 3.1. Experimental Setup

The models constructed using different kernels were tested on analog and mixed signal circuits. It was made sure that circuits operate in the feasible design space. The feasible design space is defined as a multidimensional space in which every design point satisfies a set of design constrains. Feasibility macromodels define the feasible design space, whereas Performance macromodels are mathematical models that approximate the relationship between controllable design parameters and performance parameters. The feasible design space for the circuits was obtained after the application of geometry and functional constraints. The performance constraints were then applied for the circuits. The circuits used for analysis are listed below. Netlists of these circuits were simulated using Hspice of the Synopsys tool. The outputs from the HSpice in the form of performance parameters were taken as the expected benchmark results, while comparing the results generated by macromodels constructed with different kernels in order to check for their accuracy.

1. Continuous Time Comparator: The circuit diagram is shown in Figure 1. The width of M1, M2, M3, M4 were taken as design parameters. As we aim at optimizing W/L ratio, the length is kept constant as $0.2\mu m$. Their values were swept for a feasible range to obtain specified performance parameter which is slew rate. The width of the transistors were swept in the range shown in Table 1.

**Table 1. Design Parameters- Comparator**

| *Design Parameter* | *Range* |
|---|---|
| W1,W2,W3,W4 | 40$\mu$m to 160$\mu$m |
| W5-W9,W14 | 200$\mu$m |
| W10,W12 | 100$\mu$m |
| W11,W13 | 350$\mu$m |
| L1-L14 | 1.2$\mu$m |

2. Differential Operational Amplifier: The circuit diagram is shown in Figure 2. The design variables chosen are the width of the transistors M1, M2, M3, M4. The slew rate of the differential opamp has been taken as the performance parameter. The design parameters are shown in Table 2.

**Table 2. Design Parameters- Differential Opamp**

| *Design Parameter* | *Range* |
|---|---|
| W1,W2,W3,W4 | 30$\mu$m to 120$\mu$m |
| W5,W10 | 100$\mu$m |
| W6,W7,W8,W9 | 50$\mu$m |
| L1-L10 | 0.2$\mu$m |

3. Two Stage Opamp: The circuit diagram is shown in Figure 3. The design parameters are shown in Table 3, whereas the performance parameters are shown in Table 4. The performance parameters shown, need to satisfy constraints based on the topology selected.

**Table 3. Design Parameters- Two Stage Opamp**

| *Width* | *Range* |
|---|---|
| W1 | 120$\mu$m to 170$\mu$m |
| W2 | 120$\mu$m to 170$\mu$m |
| W3 | 50$\mu$m to 100$\mu$m |
| W4 | 50$\mu$m to 100$\mu$m |
| W5 | 120$\mu$m to 170$\mu$m |
| W6 | 150$\mu$m |
| W7 | 120$\mu$m to 170$\mu$m |
| W8 | 2*W3*W7/W5 |
| L1-L8 | 3$\mu$m |

Similar operations were carried out for mixer, bandpass filter and voltage controlled oscillator.

# 4. Results

In this section, we present comparison of performance the kernels in terms of mean square error and computation time. Table 7 shows the comparison of

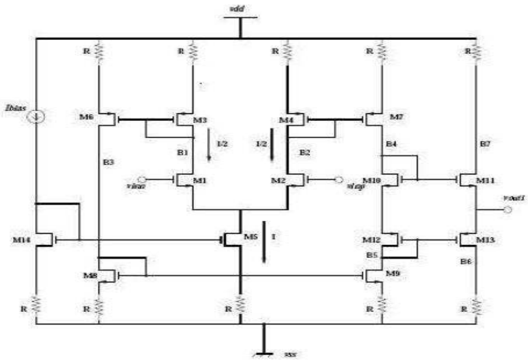


**Figure 1. Continuous Time Comparator**

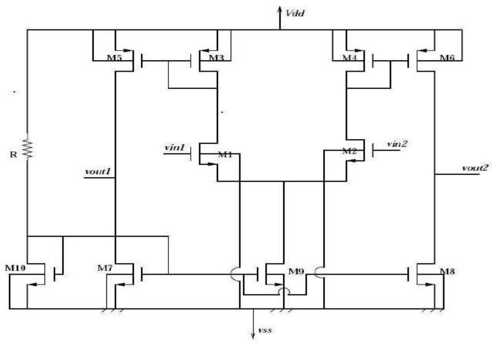


**Figure 2. Differential operational Amplifier**

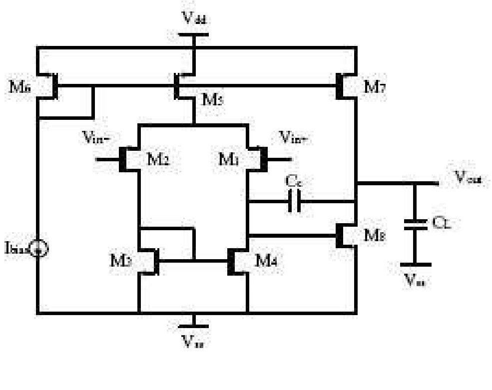



**Figure 3. Two Stage operational amplifier**

**Table 4. Performance Parameters- Two Stage Opamp**

| *Performance* | *Constraint* |
|---|---|
| CMRR | >80dB |
| PSRR | >150dB |
| Phase Margin | $> 65^o$ |
| Open Loop Gain | >20000 |
| Unity Gain Frequency | $> 5 \times 10^6$ |
| Slew Rate | $> 6 \times 10^6$V/sec |

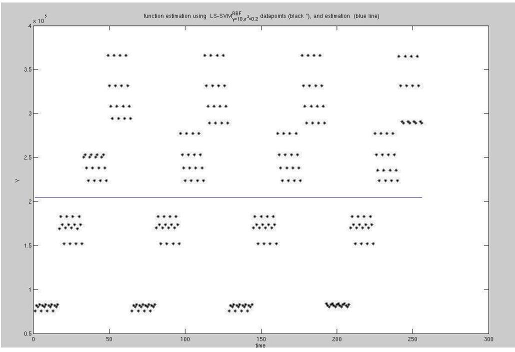
the performance of the kernels for two stage operational amplifier. We observe reduction in the error for Log and Power kernels. Multiplied and Bias Kernels further decrease the error compared to RBF and polynomial, respectively. The comparison was carried out for all the performance parameters. The modifications effected while composing the kernels have resulted in decreased error for all the performance parameters. Table 6 shows comparision of computational time of HSPICE and Macromodels. Similar results are shown in Table 7, wherein comparison is shown for the slew rate of comparator and differential opamp circuit. The results show a larger decrease in the error for multiplied and bias kernel. The reason for such a decrease in error is that RBF kernel fails to estimate an exact function as shown in Figure 4(a). These plots in Figure 4 show the extent to which the SVM model has been trained using different kernel functions. The dots shows the data patterns and the line shows the function estimated by the macromodels. As can seen from the Figure 4 that log and multiplied kernels follow the data more closely than RBF kernel. As observed in Table 8, there is a slight increase in computational time for log, power, multiplied and bias kernel compared to conventional kernel. This slight increase in time is tolerable when the accuracy has risen greatly.

**Table 6. Comparing computation times of HSpice and Macromodels**

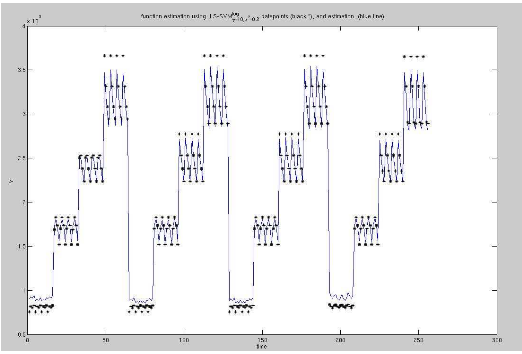
| *Performance* | *HSpice (sec)* | *Macromodel (sec)* |
|---|---|---|
| CMRR | 46.89 | 0.61 |
| PSRR | 12.61 | 0.65 |
| PHM | 12.51 | 0.61 |
| OLG | 13.60 | 0.61 |
| UGF | 14.16 | 0.65 |
| SLEW | 12.89 | 0.63 |

**Table 5. Mean Square Error- Performance Parameters for Two Stage Opamp**

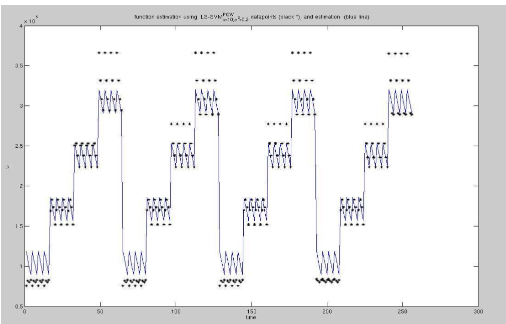
| *Kernels* | *CMRR* | *OLG* | *Phase Margin* | *PSRR* | *Slew Rate* | *UGF* |
|---|---|---|---|---|---|---|
| RBF | 1.28 | 8.99 | 1.02 | 0.71 | 10.52 | 2.97 |
| Log | 0.97 | 6.08 | 0.87 | 0.35 | 06.69 | 2.81 |
| Power | 0.96 | 5.07 | 0.80 | 0.34 | 06.59 | 2.81 |
| Multiplied | 0.86 | 4.75 | 0.26 | 5.50 | 02.79 | 0.85 |
| Polynomial | 1.28 | 8.99 | 0.71 | 10.52 | 02.97 | 1.02 |
| Bias | 1.15 | 8.98 | 1.02 | 0.70 | 10.10 | 2.96 |



(a) RBF kernel



(b) Log kernel



(c) Multiplied Kernel

**Figure 4. HSpice data vs estimated Function- Comparator Slew Rate**

**Table 7. Mean Square Error- Slew Rate**

| *Kernels* | *Comparator* | *Differential Opamp* |
|---|---|---|
| RBF | 34.33 | 106 |
| Log | 21.72 | 70.2 |
| Power | 20.86 | 67.5 |
| Multiplied | 7.16 | 6.06 |
| Polynomial | 34.34 | 106 |
| Bias | 10.38 | 16.59 |

**Table 8. Total Time (training and testing) in seconds for SVM macromodels**

| *Kernels* | *Comparator* | *Differential Opamp* |
|---|---|---|
| RBF | 0.145 | 0.139 |
| Log | 0.209 | 0.199 |
| Power | 0.172 | 0.177 |
| Multiplied | 0.148 | 0.142 |
| Polynomial | 0.587 | 0.539 |
| Bias | 0.595 | 0.562 |

## 5. Conclusions and Future Work

We have proposed SVM macromodels with efficient kernel functions. These kernel functions have shown consistency for all the analog circuits. There has been a reduction in the mean square error for proposed composite kernels. These analog models supposedly replace Spice simulators, consuming very little time and are almost as accurate as Spice. Our further work includes the tuning of kernel parameters which has been the biggest issue as far as accuracy of models is concerned. Further extending these efficient kernels approach to the classification problem in analog domain si also being pursued.

**Acknowledgements** We are grateful to Prof. R. Sharan, LNM-IIT, Jaipur (Ex-professor Indian Institute of Technology Kanpur, India) and Prof. D. Nagchoudhuri, DA-IICT Ghandinagar (Ex-professor Indian Institute of Technology Delhi, India) for very helpful suggestions during the work. We are thankful for and acknowlege financial support provided by Ministry of Communication & Information Technology, Govt. of India through phase-2 of Special Manpower Development Project for VLSI Design & related manpower.

## References


[1] V. Vapnik. The nature of statistical theory, Springer-Verlag, New York, 1995.

[2] A. Smola and B. Scholkopf. A tutorial on support vector regression, NeuroCOLT Technical Report Series, NC-TR-98-030, Oct. 1998.

[3] L.Y. Deng, Y.J. Tian. A New Method of Data Mining-Support Vector Machines, Science Publishing Company, Beijing, 2004, pp. 352-355.

[4] Least Squares Support Vector Machine Matlab/C Toolbox. http://www.esat.kuleuven.be/sista/lssvmlab.

[5] Johan A.K. Suykens, Tony Van Gestel, Jos De Brabenter, Bart De Moor and Joos Vandewalle. Least Square Support vector Machines, World Scientific Publishing Co.Pte. Ltd 2002.

[6] Bernhard Scholkopf and Alexander J. Smola. Learning with kernels, MIT Press 1999.

[7] B. Scholkopf. The kernel trick for distances, NIPS, 2000, pp. 301-307.

[8] Sabri Boughorbel, Jean-Philippe Tarel, Nozha Boujemaa. Conditionally Positive Definite Kernels for SVM Based Image Recognition. ICME 2005: 113-116.

[9] Jing-Xu Liu, Jin Li, Yue-Jin Tan. An Empirical Assessment on the Robustness of Support Vector Regression with Different Kernels, Proceedings of International Conference on Machine Learning and Cybernetics, Page(s):4289 - 4294 Vol. 7 IEEE 2005 .

[10] G.F. Smits, E.M. Jordaan. Improved SVM Regression using Mixtures of kernels. IJCNN '02, Volume 3, Page(s):2785 - 2790 IEEE 2002.

[11] M Hershenson, S. Boyd, and T.Lee. Optimal design of a CMOS op-amp via geometric programming. IEEE Trans. on CAD of Integrated Circuits and Systems 20(1): 1-21, 2001.

[12] Ovidiu Ivanciuc. Applications of Support Vector Machines in Chemistry, edited by Kenny B. Likowitz and Thomas R. Cundari, 2007 Wiley-VCH, John Wiley and Sons.

[13] Chih-Chung Chang, Chih-Jen Lin. LIBSVM-a library for support vector machines, Available: http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/.

[14] S.Boyd and L. Vandenberge. A Convex Optimization, Cambridge University press, 2004.

[15] Gleen Wolfe and Ranga Vemuri. Extration and use of Neural Network Model in Automated Synthesis of Opamp, IEEE transaction on CAD of ICs and System, vol 22, No.2, Feb 2003.

[16] Vojislav Kecman. Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT Press, 2001.

[17] Bernhard Scholkopf, Christiopher J.C. Burgs and Alexander J. Smola. Advance in Kernel methods Support Vector Learning, MIT Press Cambridge, USA,1998.

[18] R. Harjani and J. Shao. Feasibility and peformance region modeling of analog and digital circuits. In Analog Integrated Circuits and Signal Processing, volume 10, pages 23-43. Kluwer Academic Publishers, 1996.

[19] T. Kiley and G. Gielen. Performance modeling of analog integrated circuits using least-squares support vector machines. In proceeding of Design, Automation and Test in Europe Conference and Exhibition, pages 448-453, 2004.

[20] W. Daems and G. E. Gielen and W. Sansen. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. IEEE Trans. on CAD of Integrated Circuits and Systems 22(5), pages 517-534, 2003.

[21] S.R. Gunn. Support vector machines for classification and regression, Technical Report, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science, University of Southampton, 1998.