Analog Circuit Feasibility Modeling using Support Vector Machine with Efficient Kernel Functions

D. Boolchandani, *Member IEEE*[*] Chandrakant Gupta Vineet Sahula, *Senior Member IEEE* [†]

Abstract—**Performance Macromodeling facilitates accelerated analog circuit synthesis. It usually consist of two steps: feasibility design space identification and performance macromodels generation. A feasibility design space is defined as a multidimensional space in which every design satisfies all the design constraints. The minimum set of constraints is the one that ensures the correct functionality of the given circuit topology. Performance macromodels are only constructed and thereby valid in the functionally correct design space. Support vector machines (SVMs) are used as classifier to identify the feasible design space of analog circuits. A kernel is an integral part of the SVM and contributes in obtaining an optimized and accurate classifier. The most commonly used kernels are Radial Basis Function (RBF), polynomial, spline, multilayer perceptron. In this paper, some new kernels and some other kernels composed through modifications on the some of the standard kernels, are explored. The classifiers using these kernel functions have been tested on different analog circuits in order to identify the feasible design space. HSPICE has been used for generation of learning data. Least Square SVM toolbox interfaced with MATLAB was used for classification. We found that use of modified kernels improves classification accuracy as well as shortens classifier generation time.**

Keywords: Analog synthesis, macromodels, Support Vector Machine, kernel, feasibility classification

1 Introduction

Given a circuit topology, we can pose three types of constraints [1].

Geometric constraints, C_g are posed directly on the resistor, capacitor, bias voltage and currents and devices sizes e.g. width and lengths. The matching constraints on the devices are satisfied by assigning one design variable to the matched devices. After matching is taken in to account, the controllable device sizes are abstracted into a vector of independent design variables $x = x_1, ..., x_n \in R^n$. The constraints on the device sizes are usually given in the form of lower and upper bounds. The lower bounds can be determined by the feature size of a technology. The upper bounds can be selected by the designer such that the devices are not excessively large. Geometry constraint are transformed into the form of eqn. (1).

$$C_g = \{lb_i \leq X \leq ub_i, i = 1...n_g\} \tag{1}$$

Functional constraints C_f ensure the desired functionality of the given circuit topology. They are often biasing constraints posed on the nodal voltages v and branch currents i in analytic form. A circuit level simulator is required to obtain these values in order to check functional constraints. These constraints can be represented via simple transformation, as in eqn. (2).

$$C_f = \{x : f_i(v, i) \leq 0, i = 1...n_f\} \tag{2}$$

Performance constraints $C_p$are posed on the performance parameters p chosen according to the applications, viz. open loop gain, unity gain frequency, phase margin for an op amp.

$$C_p = \{x : f_i(p) \leq 0, i = 1...n_p\} \tag{3}$$

Device size ranges and functional constraints take part in defining the feasibility design space, while performance constraints don't. The feasibility design space $S \subseteq R^n$ is defined as in eqn. (4). Note that x is a vector of all the design variables.

$$S = \{\ x : x \in R^n, C\}\,;\quad C = C_g \cup C_f \cup C_p \tag{4}$$

We define a feasibility function $y(x)$, which only takes two values {+1,-1} depending on whether $x \in S$,

$$y(x) = \begin{cases} +1 \text{ if } x \in S \\ -1 \text{ if } x \notin S \end{cases} \tag{5}$$

Feasibility design space identification is necessary in building performance macromodels since it screens out infeasible designs of which performance parameters are essentially noise to a regression based macromodeling technique. It is also essential during analog circuit design and synthesis, in general since it insures the functional correctness of the circuits. Feasibility function is approximated, since checking whether a design is feasible or not requires computationally expensive simulation. So it is called as feasibility macromodeling. Feasibility macromodeling is treated as classification problem and existing classification techniques are applied to solve it. Instances from simulations are used to train a selected model with objective of minimizing the classification error on the training set.

[*]He is currently an Assistant Professor in the Dept. of ECE at Malaviya National Institute of Technology Jaipur-302017 INDIA. Email: dbool6@gmail.com

[†]He is currently an Associate Professor in the Dept. of ECE at Malaviya National Institute of Technology Jaipur-302017 INDIA. Email: sahula@ieee.org

The authors would like to gratefully acknowledge the financial support received for research work under project SMDP-VLSI phase-2 sponsored by Ministry of Comm. & IT, Govt. of India.

The technique of Support vector machines (SVMs) that has been successfully applied to solve many practical problems in various fields is used for generation of feasibility classifier models.

2 Previous Work

2.1 Support Vector Classification

SVMs [2] were proposed originally in the context of machine learning, for classification problems on (typically large) sets of data which have an unknown dependence on (possibly many) variables. We consider each of N data points $x_k \in R^n, k = 1, ..., N$ to be associated with a label $y_k \in \{+1, -1\}$ which classifies the data into one of two sets. In the simplest SVM formulation, the problem of finding a general representation of the classifier $y(x)$ becomes that of the construction of a hyper-plane $\omega^T x_k + b$ which provides 'maximal separation $\frac{2}{\|\omega\|^2}$ between points x_k belonging to the two classes. This give rise to an optimization problem of the form

$$P : min_{\omega,b} \quad \frac{1}{2}\omega^T\omega \quad s.t. \quad y_k[\omega^T x_k + b] \geq 1, \tag{6}$$

where the $\frac{1}{2}\omega^T\omega$ term represents a cost function to be minimized in order to maximize separation. The constraints are formulated such that the nearest points x_k with labels [either +1 or -1] are (with appropriate input space scaling) at least $\frac{1}{\|\omega\|^2}$ distant from the separating hyper-plane. To solve this 'primal minimization problem, we construct the dual maximization of eqn. (6) using the Lagrangian form

$$D : \max_{\alpha} \mathcal{L}(w, b; \alpha), \tag{7}$$

where

$$\mathcal{L}(\omega, b; \alpha) = \frac{1}{2}\omega^T\omega - \sum_{k=1}^{N} \alpha_k(y_k[\omega^T x_k + b] - 1), \tag{8}$$

and α_k are the Lagrange multipliers. After applying the conditions for optimality

$$\frac{\partial\mathcal{L}}{\partial\omega} = 0, \frac{\partial\mathcal{L}}{\partial b} = 0, \frac{\partial\mathcal{L}}{\partial\alpha_k} = 0, \tag{9}$$

and eliminating ω by expressing it in terms of $\alpha = [\alpha_1, ..., \alpha_N]$, we arrive at a Quadratic Programming (QP) problem

$$min(\alpha Q\alpha + B\alpha), \tag{10}$$

for suitably defined matrices Q, B. Having solved for α, the following classifier representation is obtained

$$y(x) = sign\left[\sum_{k=1}^{\#SV} \alpha_k y_k x_k^T x + b\right]. \tag{11}$$

Here #SV represents the number of non-zero Lagrange multipliers α_k, called support vectors, corresponding to input data x_k. The SVM representation will be sparse if only a few of the input data, called support vectors, are 'near to the separating hyper-plane. A key feature of the Support Vector Machines is the ability to replace the input data by a non-linear function $\phi(x)$ operating on the input data. This may be viewed as mapping the input data to higher dimensional space, to enable classification of data that is not linearly separable in the original input space. To do this, we formally replace $x_k^T x$ (the dot product between a support vector x_k and any point x of the input space) in eqn. (11) by $\phi(x_k)^T\phi(x)$ to represent the action of this mapping, obtaining

$$y(x) = sign\left[\sum_{k=1}^{\#SV} \alpha_k y_k \phi(x_k)^T\phi(x) + b\right]. \tag{12}$$

In the cases where $\phi(.)$ is infinite-dimensional, we invoke the so-called 'kernel trick': the expression $\phi(x_k)^T\phi(x)$ may under certain conditions be replaced by a Kernel function $K(x_k, x)$. An equivalent interpretation is that the kernel function is a suitably-defined dot product $< x_k, x >$ replacing $x_k^T x$ in the Hilbert space defined by the mapping ϕ. In this way, we avoid ever having to represent the mapping ϕ explicitly. In either case, the use of a kernel function allows the SVM representation to be independent of the dimensionality of the input space. There are different kernel functions that provide the SVM, the ability to model complicated separation hyperplanes. However, because there is no theoretical tool to predict which kernel will give the best results for given data set, experimenting with different kernels is only way to identify the best function. These kernel functions must satisfy certain criteria known as Mercer conditions for preserving the convexity of the problem. These Mercer conditions are discussed in next Section.

2.2 Mercer kernel

If the kernel *K* is a symmetric positive definite function, which satisfies the Mercer's conditions

$$K(x_k, x) = \sum_{i}^{\infty} a_i\phi_i(x_k)\phi_i(x), a_i > 0 \tag{13}$$

$$and \int\int K(x_k, x)g(x_k)g(x)dx_k dx > 0 \tag{14}$$

then the kernel K would represents an inner product in feature space

$$K(x_k, x) = \phi(x_k) \cdot \phi(x) \tag{15}$$

and is known as Mercer Kernel.

From this condition the simple rules for composition of kernels can be concluded, which also satisfy Mercer's condition [3]. Corollary 1 (Linear combinations of kernels): Let $k_1(x_k, x), k_2(x_k, x)$ be Mercer kernels and $c_1, c_2 \geq 0$, then

$$k(x_k, x) = c_1k_1(x_k, x) + c_2k_2(x_k, x) \tag{16}$$

is also called a Mercer kernel. Moreover, the product of two Mercer kernels is a Mercer kernel, which is proved based on the equivalent definition of Mercer kernel. Similarly, it has been proposed in [4] 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 so obtained is also a Mercer Kernel. Mercer condition needs to be satisfied for keeping the problem convex and hence obtaining a unique solution. Some of the useful modifications on kernels are illustrated in eqns. (17) and (18).

$$k(x_k, x) = \alpha k(x_k, x) \text{ where } \alpha > 0 \quad (17)$$

$$k(x_k, x) = k(x_k, x) + b \text{ where } b > 0 \quad (18)$$

Also, two of the other kernels that are applied in the present work are power kernel [5]

$$k(x_k, x) = - \parallel x - x_k \parallel^{\beta} \quad (19)$$

and log kernel [6]

$$k(x_k, x) = -log(1 + \parallel x - x_k \parallel^{\beta}) \quad (20)$$

where the kernels are conditionally positive definite for $0 < \beta \leq 1$ All the kernels discussed above satisfy the Mercer's condition, which is necessary for the problem to be convex, and hence providing unique and optimum solution.

2.3 Related Work

An approach to model the feasible design space and evaluate the performance of sub-blocks at all levels has been proposed in [7]. In this work, authors have used fractional factorial experiment design techniques to measure the significance of input variables. Variable screening and grouping techniques are employed to select and organize the input variables based upon their influence on the output response. An adaptive volume slicing technique is used during regression analysis to dynamically distribute regressors such that the number of experimental runs is minimized. However, it is a rule based sizing framework, resulting in less accurate solutions. In [8], authors calculate feasible design space by linear approximation. In their work, the functional constraints of an op-amp are posed by inheriting all the functional constraints of subcircuits. Sensitivity analysis is done around a feasible design to approximate the feasible design space. However, overall accuracy achieved is only 70% while other drawback is that the selection of design on which sensitivity analysis is performed, can change the approximated feasibility design space.

Authors in [9] have presented a novel approach for modeling the performance space of an analog circuits based on SVMs. An analog circuit maps a set of input design parameters to a set of performance figures. The function is evaluated through simulations and its range defines the feasible performance space of the circuit. The resulting model provides a clear separation of abstraction levels, directly modeling performance relations in place of regression on implementation parameters. In [10] Pareto-optimal hyperplane, which delimits the design space for the circuit at hand is derived by the use of multiple-objective genetic optimization and multivariate regression techniques. It helps designer in exploring the tradeoff between different competing objectives in analog and RF integrated circuit design. Results obtained can be used both in the system-level design phase for topology selection and in the circuit-level design phase for optimal design.

Proposal in [11] is for active learning scheme for feasibility design space identification. The proposed methodology uses a committee of classifiers to exclude a large portion of entire design space and samples only the feasibility region and its neighboring. It improves the accuracy of the classifier with much fewer samples, resulting in computation time reduction, compared to a passive learning scheme using uniform random samples. Authors in [12] have presented an approach for generation of yield aware Pareto surface for hierarchical circuit design space exploration. A non-dominated sorting based global optimization algorithm is used to generate the nominal Pareto front for VCO circuit. Solutions on this Pareto front with efficient Monte Carlo analysis are then used to compute the yield aware Pareto fronts. These Pareto surfaces of VCO are then used to synthesize PLL with a targeted yield.

3 Proposed Work

The scope of the present work is identification of feasible design space for analog circuits using SVM scheme and evaluation of the scheme on two analog circuits- a two stage op-amp and a cascode op-amp. Widths of the transistors, Coupling Capacitor and Bias currents for above two circuits are taken as design variables. A known instance of all the design variable is considered a tuple. Values of these design variables for both circuits were randomly generated within upper and lower bound to get a set of 10000 tuples of design variables. These 10000 tuples of design variable serve as input data. HSPICE is run for this set of 10000 tuples of design variables. Functional constraints and performance constraints are verified using HSPICE simulation. For the given set of tuples which satisfy both functional and performance constraints output is taken as '1' otherwise as '-1'. This results in 10000 input and output data pair. Of these 6000 are used to train SVM classifier and 4000 are used for validation to check accuracy of classifier. Least Square Support Vector Machine Toolbox [13] interfaced with MATLAB is used for classification. The toolbox outputs the value of optimized α and bias. These values are used to form a classifier as shown in eqn. (12). As it is evident in Section 2.2 that the kernel has an important role to play in classification.Suitability of various kernels is explored. Modifications is carried out on RBF kernel and other suitable form of kernels to obtain Multiplied kernel eqn. (17) and Bias kernel eqn. (18). The model is trained using RBF, Log, Power and modified kernels. Kernels are compared for accuracy and computation time while they are used for classification.

3.1 Accuracy measurement

Modeled classifier can be made highly accurate by properly choosing the parameters of SVMs. The generalization ability

of the classifier is examined by an independent validation data set. The learned function usually deviates from the true underlying function. Let S denote the entire design space after application of geometry constraints, as illustrated in the Figure 1.

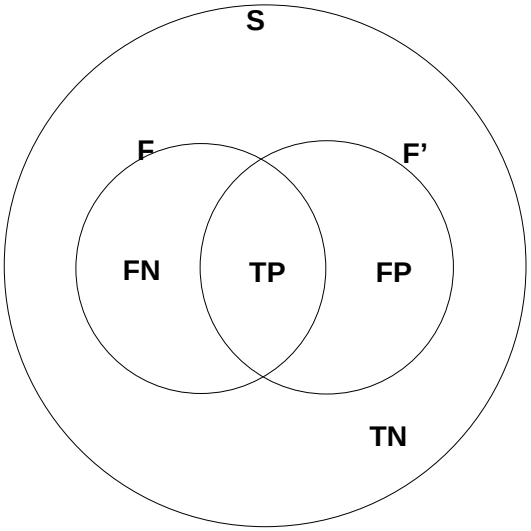

Figure 1: Design Space and its subspace

In Figure 1, F is the feasibility design space and F' is the approximated feasibility space.Thus S is divided by F and F' into four subspace: TP of true positives, TN of true negatives, FP of false positives and FN of false negatives.Accuracy is calculated using formula shown below. Here TP is true positive, predicted positive by the classifier which are actual positive, and similarly TN is true negative.

$$accuracy = \frac{(\mid TP \mid + \mid TN \mid)}{\mid S \mid} \tag{21}$$

4 Experimental setup

We show two op-amps as our illustrative examples. We will show the accuracy improvement of the feasibility classifier constructed by the proposed kernels compared to those constructed by standard kernels. The classifiers constructed using different kernels were trained and tested using data generated form HSPICE.

4.1 Two Stage op-amp

The two-stage op-amp is shown in Figure 2. As all transistors are required to operate in saturation mode,we fix the length of all transistor to a nominal minimum length. This immediately eliminates nearly half of the free design parameters. Further the size of transistor M1 should equal M2, and the size of M3 should equal M4 to equalize the currents through the differential pair. Both $W_1 = W_2$ and $W_3 = W_4$ are left as free parameters. Transistor M6 can be fixed to some minimum nominal size since its job is to simply mirror the reference current Ibias, which can also be fixed. The width of transistors M5 and M7 control the current through the differential pair and output stage respectively and are also left as free parameters.In order to minimize the DC offset voltage at the output node, width of transistor M8 is taken as $2 * W_3 * W_7/W_5$. This is because the current through $M4 = 0.5 * Ibias * W_5/W_6$. As M3 and M4 transistors are of same size, have equal drain currents, and have the same gate to source voltages, so the drain voltage of M4 is equal to the drain/gate voltage of M3. Thus the gate voltage of M8 is equal to the drain voltage of M4, which is equal to the drain/gate voltage of M3. This causes M8 to mirror the current through transistors M3 and M4 by the ratio W_8/W_3. Putting this all together we have the current through $M8 = 0.5 * (Ibias * W_5/W_6) * W_8/W_3$ and the current through $M7 = Ibias * W_7/W_6$. Equating the currents through M8 and M7 yields the necessary width of $M8 = 2 * W_3 * W_7/W_5$. Lastly the compensation capacitor is left as a free variable since it controls the inherent stability of the op-amp. The load capacitor is taken as fixed variable to simplifying the modeling problem. The above arguments result in the 5-dimensional parametric configuration for the two-stage op-amp. The design variables and geometry constraints are shown in Table 1. Functional and Performance constraints are shown in Table 2. The functional constraints ensure all the transistors are on and in saturation region with some margin. We set $V_{on,min}$ and $V_{sat,min}$ to 0.1V.

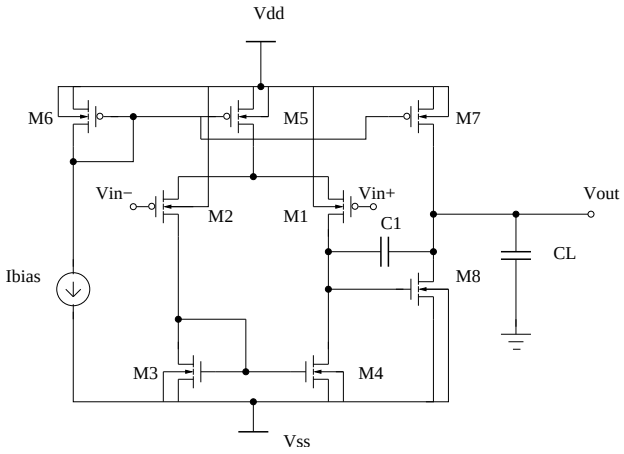

Figure 2: Two-stage op-amp

Table 1: Design Variables of Two stage Op amp

Design parameters	Geometric constraints
$W_1 = W_2$	[1μm,100μm]
$W_3 = W_4$	[1μm,50μm]
W_5	[1μm,100μm]
W_7	[1μm,100μm]
Cc	[5pF,20pF]

4.2 Cascode Op amp

The circuit of cascode op-amp is shown in Figure 3. We fix the lengths of all transistors to 1μm. Imposing sizing rules

Table 2: Design constraints of Two-stage op-amp

Functional constraints	$V_{gs} - V_{th} \geq V_{on,min}$ $V_{ds} \geq V_{gs} - V_{th} + V_{sat,min}$
Performance constraints	Phase Margin $\geq 45^o$

[14] similar to that of two-stage op-amp we get five design variables for cascode op-amp. Load capacitance is set to 1pf. The design variables and geometry constraints are shown in Table 3. Other constraints shown in Table 4. The functional constraints ensure all the transistors are on and in saturation region with some margin. We set $V_{on,min}$ and $V_{sat,min}$ to 0.1V.

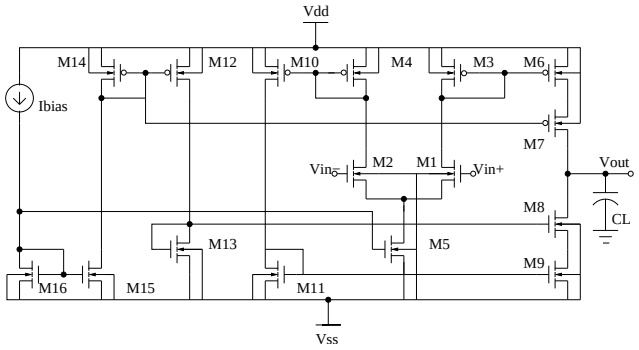

Figure 3: Cascode op-amp

Table 3: Design Variables of Cascode op-amp

Design parameters	Geometric constraints
$W_1 = W_2$	[1μm,100μm]
$W_3 = W_4$	[1μm,100μm]
$W_5 = W_6$	[1μm,100μm]
Ibias	[2μA,20μA]
C_L	[1pF,10pF]

Table 4: Design constraints of Cascode op-amp

Functional constraints	$V_{gs} - V_{th} \geq V_{on,min}$ $V_{ds} \geq V_{gs} - V_{th} + V_{sat,min}$
Performance constraints	Phase Margin $\geq 60^o$

5 Results

We have shown the improvement in accuracy of the classifier constructed with the use of proposed kernels, the results are shown in Tables 5 and 6. Table 5 and 6 show the comparison of accuracy and computation time while using different kernels for two-stage op-amp and cascode op-amp, respectively. We observe significant reduction in computation time with similar or better accuracy. These results suggest an improvement in performance of the classifier using proposed kernels.

Table 5: Comparing kernels for Two-stage op-amp

Kernels	**Accuracy** (in %)	**Computation time** (in sec)	**Speed-up**
RBF	92.1	403.37	1.0
Log	96.7	121.20	3.3
Power	96.8	157.27	2.6
Multiplied	94.3	110.31	3.7

Table 6: Comparing kernels for Cascode op amp

Kernels	**Accuracy** (in %)	**Computation time** (in sec)	**Speed-up**
RBF	90.8	415.23	1.0
Log	95.2	124.42	3.3
Power	95.4	156.94	2.6
Multiplied	95.8	112.33	3.7

6 Conclusions & future work

We have presented a feasibility macromodel, which can be used during synthesis of analog circuits. The generated model, incorporating the proposed kernels has been found to be much more efficient while computing the performance, compared to those constructed using standard kernels. We treated the feasible design space identification problem as a two-class classification problem so that comparison can be done for larger size of data set. Thus, we are able to build accurate and fast feasibility macromodels, which can tremendously save computation time during circuit sizing when circuit performance parameters are to be evaluated a large number of times in a stochastic optimization engine.

Further work using proposed kernels for regression problems as well is being pursued. Also, a method is to be adopted to tune the parameters of the kernels for different set of the application circuits. There still remains a work to be done for further improving the accuracy of macromodels.

Acknowledgments

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

References

[1] M. Ding and R. Vemuri, “A combined feasibility and performance macromodel for analog circuits,” in *IEEE Design Automation Conference*, 2005, pp. 63–68.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[3] G. Smits and E. Jordaan, "Improved svm regression using mixtures of kernels," *Proceedings of International Joint Conference on Neural Networks*, vol. 3, pp. 2785–2790, 2002.

[4] J. A. Suykens, T. Gestel, J. Brabenter, B. Moor, and J. Vandewalle, *Least Square Support vector Machines*. World Scientific Publishing Co. Pte. Ltd, 2002.

[5] J.-X. Liu, J. Li, and Y.-J. Tan, "An empirical assessment on the robustness of support vector regression with different kernels," *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 4289–4294, Aug. 2005.

[6] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Conditionally positive definite kernels for svm based image recognition," in *ICME*, 2005, pp. 113–116.

[7] R. Harjani and J. Shao, "Feasibility and performance region modeling of analog and digital circuits," *Analog Integrated Circuits and Signal Processing*, vol. 10, pp. 23–43, 1996.

[8] S. Zizala, J. Eckmuller, and H. Grab, "Fast calculation of analog circuits' feasibility regions by low level functional measures," *IEEE International Conference on-Electronics, Circuits and Systems*, vol. 2, pp. 85–88, 1998.

[9] F. D. Bernardinis, M. I. Jordan, and A. L. Sangiovanni-Vincentelli, "Support vector machines for analog circuit performance representation," in *DAC*, 2003, pp. 964–969.

[10] B. De Smedt and G. Gielen, "Watson: design space boundary exploration and model generation for analog and rfic design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 22, no. 2, pp. 213–224, Feb. 2003.

[11] M. Ding and R. Vemuri, "An active learning scheme using support vector machines for analog circuit feasibility classification," in *vlsi*, 2005, pp. 528–534.

[12] S.K.Tiwary, P.K.Tiwary, and R.A.Rutenbar, "Generation of yield-aware pareto surfaces for hierarchical circuit design space exploration," in *IEEE Design Automation Conference*, 2006, pp. 31–36.

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

[14] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The sizing rules method for analog integrated circuit design," *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 343–349, 2001.

[15] D. Boolchandani, A. Ahmed, and V. Sahula, "Improved support vector machine regression for analog circuits macromodeling using efficient kernel functions," in *IEEE International Workshop on Symbolic and Numerical Methods, Modeling and applications to circuit design*, 7-8 October 2008, pp. 61–67.

[16] T. Kiely and G. G. E. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *DATE*, 2004, pp. 448–453.

[17] A. Smola and B. Scholkopf, "A tutorial on support vector regression: NC-TR-98-030," Tech. Rep., October 1998.

[18] W. Daems, G. G. E. Gielen, and W. M. C. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.

[19] B. Scholkopf and A. J. Smola, *Learning with kernels*. MIT Press, 1999.

[20] S.Boyd and L. Vandenberge, *Convex Optimization*. Cambridge University press, 2004.

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

[22] M. Hershenson, S. Boyd, and T. Lee, "Optimal design of a cmos op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, pp. 1–21, Jan 2001.

[23] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, Feb. 2003.

[24] W. Pedrycz, "Advances in kernel methods." *Neurocomputing*, vol. 47, no. 1-4, pp. 303–304, 2002.