Exploring Efficient Kernel Functions for Support Vector Machine based Feasibility Models for Analog Circuits

D. Boolchandani, *Member, IEEE* and Vineet Sahula, *Senior Member, IEEE*

***Abstract*—Support Vector Machines (SVMs) have been used as classifier to identify the feasible design space of analog circuits. A feasibility design space is defined as a multidimensional space in which every point representing a 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 that facilitates accelerated analog circuit synthesis are constructed and thereby valid in the functionally correct design space. A kernel function is an integral part of the SVM and contributes in obtaining an optimized and accurate classifier. A kernel function serves as a separating function, a hypersurface which optimally separates input data into two classes involving minimal support vectors. The support vectors are data points in input space lying on kernel function hypersurface. There is no formal way to decide, which kernel function is suited to a class of classifier problem. While most commonly used kernels are Radial Basis Function (RBF), polynomial, spline, multilayer perceptron; we have explored many other un-conventional kernel functions and kernels composed through modifications on the some of the standard kernels functions. The classifiers using these new 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 training time.**

***Index Terms*—Analog synthesis, macromodels, Support Vector Machine, kernel, feasibility classification.**

I. 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.

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

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. 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

D. Boolchandani is currently an Associate Professor in the Dept. of ECE at Malaviya National Institute of Technology Jaipur-302017 INDIA. Email: dbool@ieee.org

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

Manuscript received August 7, 2009.

The work was supported by a research grant from Ministry of Comm. & IT, Govt. of India through sponsored project SMDP-VLSI phase-2.

feasibility classifier models. The SVMs are a class of machine learning algorithms. In the next section we discuss support vector classifiers and briefly review the work done in literature.

II. Previous Work

A. 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. However for the Least-Squares SVM classification modification is done such that upon the target value an error variable e_k is allowed so that misclassifcations can be tolerated in case of overlapping distributions and following optimization problem is formulated in the primal weight space for given a training set $\{x_k, y_k\}_{k=1}^N$

$$P : \min_{w,b,e} J_p(w, e) \quad = \quad \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 \tag{7}$$

together with the N constraints as given in equation 8. This formulation involves the trade off between a cost function term and a sum of squared errors governed by the trade-off parameter γ.

$$y_k \left[w^T\phi(x_k) + b\right] = 1 - e_k, \ k = 1, ..., N \tag{8}$$

To solve this 'primal minimization problem, we construct the dual maximization of eqn. (7) using the Lagrangian form

$$D : \max_{\alpha} \mathcal{L}(w, b, e; \alpha), \tag{9}$$

where

$$\mathcal{L} = J_p(w, e) - \sum_{k=1}^{N} \alpha_k\{y_k[w^T\phi(x_k) + b] - 1 + e_k\}, \tag{10}$$

and α_k are Lagrange multipliers. The conditions for optimality are given by

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial w} = 0 \ \rightarrow \ w = \sum_{k=1}^{N} \alpha_k y_k \phi(x_k) \\ \frac{\partial\mathcal{L}}{\partial b} = 0 \ \rightarrow \ \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial\mathcal{L}}{\partial e_k} = 0 \ \rightarrow \ \alpha_k = \gamma e_k, \qquad k = 1, ..., N \\ \frac{\partial\mathcal{L}}{\partial \alpha_k} = 0 \ \rightarrow \ y_k \left[w^T\phi(x_k) + b\right] - 1 + e_k = 0, \ k = 1, ..., N \end{cases} \tag{11}$$

After elimination of the variables w and e one gets the following solution

$$\begin{bmatrix} 0 & y^T \\ y & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_\nu \end{bmatrix} \tag{12}$$

where $y = [y_1; ...; y_N]$, $1_v = [1; ...; 1]$ and $\alpha = [\alpha_1; ...; \alpha_N]$.

The kernel trick is applied here as follows

$$\begin{aligned} \Omega_{kl} &= y_k y_l \phi(x_k)^T \phi(x_l) \\ &= y_k y_l K(x_k, x_l) \qquad k, l = 1, ..., N \end{aligned} \tag{13}$$

The resulting LS-SVM model for classifier then becomes

$$y(x) = sign\left[\sum_{k=1}^{N} \alpha_k y_k K(x_k, x) + b\right]. \tag{14}$$

where α_k, b are the solution to the linear system given by equation 12 and N represents the number of non-zero Lagrange multipliers α_k, called support vectors.

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. 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, as shown in Table I. 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.

B. Mercer kernel

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

$$K(x_k, x) = \sum_{i}^{\infty} a_i \phi_i(x_k)\phi_i(x), a_i > 0 \qquad and, \tag{15}$$

$$\int\int K(x_k, x)g(x_k)g(x)dx_k dx > 0 \tag{16}$$

TABLE I
LIST OF KERNELS WITH THEIR EXPRESSION

Name of the kernel	Expression of the Kernel
Linear kernel	$K(x, x_j) = x_k^T x$
RBF kernel	$K(x, x_j) = e^{\left(-\frac{\|x-x_k\|^2}{\sigma^2}\right)}$
Hybrid kernel	$K(x, x_j) = e^{-\frac{\|x-x_k\|^2}{\sigma^2}} \times \left(\tau + x_k^T x\right)^d$
Multiplied kernel	$k(x, x_k) = a \times k(x, x_k)$ where $a > 0$
Power kernel	$k(x, x_k) = - \parallel x - x_k \parallel^\beta \ \ 0 < \beta \leq 1$
Log kernel	$k(x, x_k) = -log(1+ \parallel x - x_k \parallel^\beta) \ 0 < \beta \leq 1$

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

$$K(x_k, x) = \phi(x_k) \cdot \phi(x) \tag{17}$$

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

$$k(x_k, x) = c_1 k_1(x_k, x) + c_2 k_2(x_k, x) \tag{18}$$

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 [5] are illustrated in equations (19), (20) and (21).

$$k(x_k, x) = a \times k(x_k, x) \text{ where } a > 0 \tag{19}$$

$$k(x_k, x) = k(x_k, x) + b \text{ where } b > 0 \tag{20}$$

$$k(x_k, x) = a \times e^{\left(\tau + x_k^T x\right)} \text{ where } a > 0 \tag{21}$$

Also, two of the other kernels that are used in the present work are power kernel and log Kernel [6] given in Table I. List of kernels that have been explored are given in Table I. All these kernels satisfy the Mercer's condition, which is necessary for the problem to be convex, and hence provides unique and optimum solution.

C. 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. The concept of hierarchical decomposition and application of functional constraints are used throughout the paper.In their work, the functional constraints of an op-amp are posed by inheriting all the functional constraints of sub-circuits. Example of current mirror is taken and its functional constraint are enumerated giving insight into the types of constraints necessary to ensure well behaved circuits. Each constraint defines a sub-space in electrical space. The intersection of all such sub-spaces forms the feasibility region for well behaved circuits. Authors present a method for linearizing the functional constraints as well as a formula for mapping these linear approximate constraints back to the design space. Since the approximate linear constraints are only valid around one quiescent point in both the design and electrical space, the linearized constraints can fail to detect pathological (ill behaved) designs. As an example a folded cascode op-amp was analyzed resulting in 18 constraints on device sizes, 59 functional constraints and 9 free parameters in the design space. Classification accuracy of the linearized constraint set by simulating random selection of points in the design space was tested. The linearized constraints statistically misclassified points inside the true feasibility region 15% of the time while misclassified points outside the true feasibility region about 10% of time. Modeling accuracy was found to be one order of magnitude better for both the linear and quadratic regression when constrained to 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.

In [9] method for the automatic sizing of integrated analog CMOS circuit is presented that prevents bad or pathologically sized circuits, that violate basic design rules. This is done by introducing circuit knowledge into sizing process. Basic sizing rules are setup on component level for transistor pairs and sub circuits and formulated as constraints. These structural constraints express general function and matching conditions. A systematic consideration of these structural constraints during the sizing significantly reduce the number of free design parameters, speed up the sizing, and prevents pathologically sized circuits. The sizing is done with an iterative trust region algorithm. In each step, the circuit performance and constraints are linearized and a parameter correction with a good ratio between error reduction and parameter deviation is calculated based on the characteristic boundary curve. The sizing result was applied to folded-cascode operational amplifier yielding 165 inequality constraints, 35 equality constraints and 9 free variables. The over all sizing time was reduced by a factor of three and resulting circuit is less sensitive to process variation.

Authors in [10] present sizing rule method for constraining CMOS analog circuits such that they are well behaved and contain a minimum of free variables. The library of analog sub-blocks developed by the authors is comprised of four levels. Level 0 recognizes single transistors operating in linear region or in saturation. Level 1 contains seven sub-circuits composed of transistor pairs, namely a simple current mirror, level shifter, voltage reference, current mirror load, differential pair, voltage reference and flip-flop. When building the Level 1

library, authors enumerated all 206 possible combinations of transistor pairs, discarding those not used in analog CMOS design. On Level 2, four different pairs of transistor pairs are identified as major building blocks. These are:level shifter bank, current mirror bank, cascode current mirror and 4-transistor current mirror. Lastly Level 3 contains only one sub-circuit, the differential stage. Once defined, all sub circuits in the hierarchy were analyzed to determine a suitable set of low-level functional constraints ensuring robust design practices as well as non-pathological behavior. An algorithm is given for sub block identification for any analog topology, there by making it possible to automatically generate all necessary functional constraints for a generic topology. As an example the sizing rules methodology is applied to three analog topology, thereby making it possible to automatically generate all necessary functional constraints for a generic topology. Three application ares are mentioned where application of sizing rules might be useful, these are circuit sizing, design centering and response surface modeling. It has been shown that designs are more robust with respect to operating tolerances when sizing rules are obeyed. Further it has been stated that sizing rules contribute to response surface modeling in several ways . First they provide an accurate and technically relevant feasible region of an analytical model. Second the function domain is reduced in size due to sizing rules constraints.Finally the performance behavior is near to linear in the region where sizing rules are satisfied. This results in an increased accuracy of the analytical models.

Authors in [11] 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 [12] 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 trade-off 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 [13] 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 [14] 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.

III. 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 four analog circuits- two stage op-amp, cascode op-amp, voltage controlled oscillator and mixer. Models used for transistors are Berkeley BSIM3 models in 180 nm technology. Widths of the transistors, Coupling Capacitor and Bias currents for above 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 [15] 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. (14). As it is evident in Section II-B 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 forms of kernels to obtain Multiplied kernel eqn. (19) and Bias kernel eqn. (20). The model is trained using Linear, RBF, Log, Power, Multiplied and Hybrid kernels. Kernels are compared for accuracy and model training time while they are used for classification. For two stage op amp, cascode op amp and mixer we have kept tuning parameters σ and γ as 1 and 10 respectively for all kernels. However, for VCO classifier, we have compared kernels for different combinations of tuning parameters σ and γ.

A. 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.

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.

$$accuracy = \frac{(\mid TP \mid + \mid TN \mid)}{\mid S \mid} \tag{22}$$

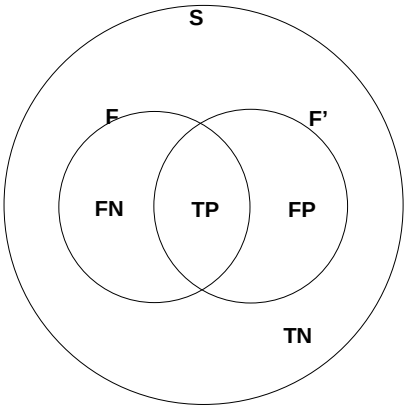

Fig. 1. Design Space and its subspace [13]

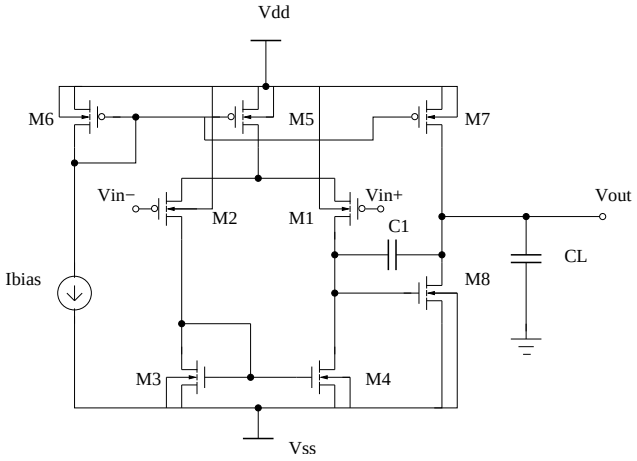

Fig. 2. Two-stage op-amp [16]

IV. Experimental setup

We show two op-amps, voltage controlled oscillator and mixer 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.

A. 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 fixed design parameters are shown in Table II. The functional constraints shown in Table III, 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.

TABLE II
Design Variables of the Two stage Op amp

Design variables	$W_1 = W_2$	[1μm,100μm]
	$W_3 = W_4$	[1μm,50μm]
	W_5	[1μm,100μm]
	W_7	[1μm,100μm]
	W_8	$\frac{2\times W_3\times W_7}{W_5}$
	C_1	[5pF,20pF]
Fixed design parameters	$L_1, \cdots, L_8$	[0.5μm]
	W_6	10μm
	I_{bias}	50μA
	C_L	5pF

TABLE III
Functional constraints of the two stage op amp.

nMOS Transistor	pMOS Transistor
$V_{gs} - V_{th} \geq V_{on,min}$	$V_{gs} - V_{th} \leq -V_{on,min}$
$V_{ds} \geq V_{gs} - V_{th} + V_{sat,min}$	$V_{ds} \leq V_{gs} - V_{th} - V_{sat,min}$

B. Cascode Op amp

The circuit of cascode op-amp is shown in Figure 3. We fix the lengths of all transistors to $0.5\mu m$. Imposing sizing rules [10] similar to that of two-stage op-amp we get five design variables for cascode op-amp. The design variables and fixed design parameters are shown in Table IV. Here W indicate the width of transistor and L indicate the length of transistor. I_{bias} is the bias current as shown in Figure 3. Functional constraints in Table III apply with $V_{on,min}$ and $V_{sat,min}$ set to $0.1V$.

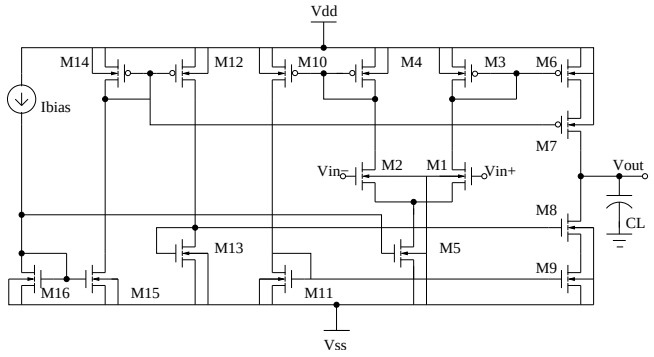

Fig. 3. Cascode op-amp [17]

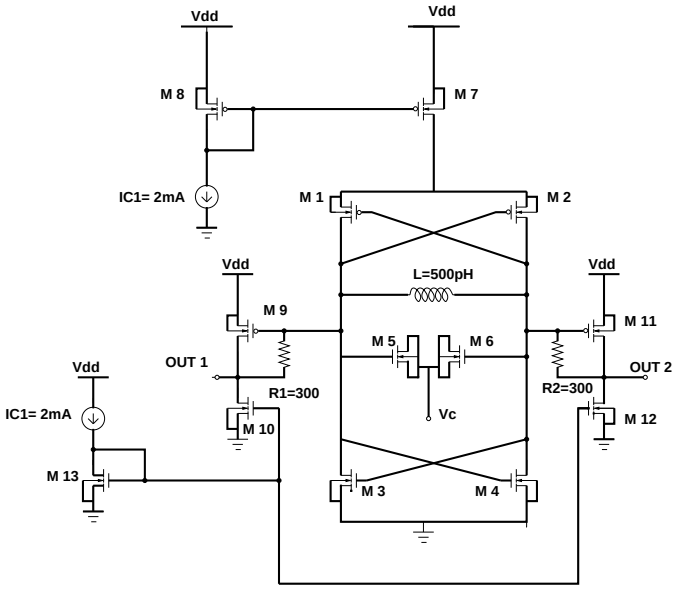

Fig. 4. Voltage controlled oscillator [18].

TABLE IV
DESIGN VARIABLES OF THE CASCODE OP-AMP

Design Variable	$W_1 = W_2$	[1μm,100μm]
	$W_3 = W_4$	[1μm,100μm]
	W_5	[1μm,100μm]
	W_6, W_7, W_8, W_9	$[W_3]$
	W_{14}	$[\frac{2\times W_5 \times W_6}{W_3}]$
	W_{13}, W_{15}, W_{16}	$[0.25 \times W_3]$
	I_{bias}	[2μA,20μA]
	C_L	[1pF,10pF]
Fixed design parameters	$L_1, \cdots, L_{16}$	[0.5μm]
	W_{10}, W_{11}, W_{12}	[10μm]

C. Voltage controlled oscillator

A voltage controlled oscillator [18] is shown in Figure 4. Widths of transistor M1, M2, M3, M4, M5, M6, M9, M10 and M11 are taken as design variables. The design variables along with fixed design parameters are shown in Table V. Functional constraints shown in Table III, ensure all the transistors are on and in saturation region except for transistor M5 and M6 which will be in linear region.

TABLE V
DESIGN VARIABLES OF THE VOLTAGE CONTROLLED OSCILLATOR

Design variables	$W_1 = W_2$	[100μm,500μm]
	$W_3 = W_4$	[50μm,300μm]
	$W_5 = W_6$	[20μm,250μm]
	$W_9 = W_{11}$	[100μm,500μm]
	W_{10}	[1200μm,2000μm]
Fixed design parameters	L_1, L_2	[0.9μm]
	L_3, L_4	[0.7μm]
	L_5, L_6	[12μm]
	L_7, L_8, L_9, L_{11}	[0.2μm]
	L_{10}, L_{12}, L_{13}	[6μm]
	W_7	[2000μm]
	W_8	[200μm]
	$W_{12} = W_{13}$	[1500μm]
	L (Inductor)	[500pH]

D. Mixer

A low voltage mixer [13] is shown in Figure 5. Length of all transistors are fixed at 1.0 μm. The design variable and fixed design parameters are listed in Table VI. It is required that all nMOS transistor are *on* and biased in saturation where as pMOS transistor should also be *on* but biased in linear region as they behave as resistors. The functional constraints for both pMOS and nMOS are shown in Table VII.

TABLE VI
DESIGN VARIABLES OF THE MIXER

Design Variable	$W_1 = W_2 = W_3 = W_4$	[50μm,200μm]
	$W_5 = W_6$	[100μm,400μm]
	$W_7 = W_8 = W_9$	[30μm,120μm]
	$WL_1 = WL_2$	[6μm,24μm]
	I_{bias}	[1mA,2mA]
	V_{RF}	[2V,3V]
	V_{LO}	[1.5V,2.5V]
Fixed Parameters	$L_1, \cdots, L_8, L_{ML_1}, L_{ML_2}$	[1.0μm]

TABLE VII
FUNCTIONAL CONSTRAINTS OF THE MIXER.

nMOS Transistor (Saturation region)	pMOS Transistor (Linear region)
$V_{gs} - V_{th} \geq 0$	$V_{gs} - V_{th} \leq 0$
$V_{ds} \geq V_{gs} - V_{th} + V_{sat,min}$	$V_{ds} \geq V_{gs} - V_{th} - V_{sat,min}$

V. RESULTS

We have observed significant improvement in accuracy of the classifiers of four circuits constructed with the use of proposed kernels. The corresponding results are shown in Tables VIII and IX which show the comparison of accuracy and model training time while using different kernels for two-stage op amp, cascode op amp, mixer and voltage controlled oscillator respectively. We observe significant speed up in model training time with a similar or better accuracy with use of particular kernels. These results suggest an improvement in performance of the classifier using proposed kernels, i.e. Log, Power and Multiplied kernels. The Log and Power kernel leads to higher accuracy where as Multiplied kernel provides speed up with moderate accuracy. Classifiers with Linear kernels are fastest but are very low on accuracy and Hybrid perform worst on both parameters in case of voltage controlled oscillator and

TABLE VIII
COMPARING KERNELS FOR ACCURACY AND MODEL TRAINING TIME (MODEL T-T) OF DIFFERENT CIRCUITS.

Kernels	Accuracy (in %)			Speed-up					
	Two stage op amp	Cascode op amp	Mixer Circuit	Two stage (Model T-T)	op amp Speed-up	Cascode (Model T-T)	op amp Speed-up	Mixer (Model T-T)	Circuit Speed-up
Linear	89.1	82.7	83.5	4.5	89	4.32	96	2.57	10.6
RBF	92.1	90.8	91.00	403.37	1.0	415.23	1.0	27.07	1.0
Log	96.7	95.2	91.15	121.20	3.3	124.42	3.3	17.09	1.5
Power	96.8	95.4	91.80	157.27	2.6	156.94	2.6	19.57	1.4
Multiplied	94.3	95.8	90.60	110.31	3.7	112.33	3.7	12.73	2.1
Hybrid	90.0	85.3	63.5	456.23	0.89	477.65	0.87	30.37	0.9

TABLE IX
ACCURACY AND MODEL TRAINING TIME FOR VCO WITH DIFFERENT σ AND γ.

Kernels	$\sigma = 0.2$	$\gamma = 10$	$\sigma = 1.0$	$\gamma = 10$	$\sigma = 10$	$\gamma = 10$	$\sigma = 27$	$\gamma = 100$
	Accuracy (in %)	**Model training time** (in sec)	**Accuracy** (in %)	**Model training time** (in sec)	**Accuracy** (in %)	**Model training time** (in sec)	**Accuracy** (in %)	**Model training time** (in sec)
Linear	79.80	3.96	80.80	3.97	80.80	3.97	78.80	3.73
RBF	96.05	46.83	99.05	121.80	98.05	123.80	99.00	145.80
Log	99.15	29.60	99.15	29.89	99.15	30.89	99.15	31.89
Power	99.10	34.23	99.10	32.80	99.10	34.80	99.10	35.48
Multiplied	96.25	35.58	97.25	35.58	96.25	49.58	97.25	49.58
Hybrid	40.45	62.25	59.55	389.25	59.55	423.25	57.55	463.25

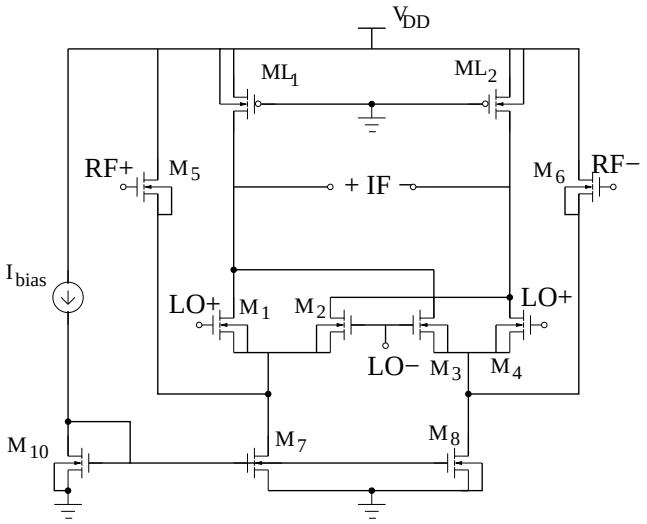

Fig. 5. Mixer circuit [13]

mixer classifiers. For VCO, different combinations of tuning parameters were explored, which are shown in Table IX. From the Table IX , we observe that the optimum values of accuracy and model training time are obtained for $\sigma = 1.0$ and $\gamma = 10$.

VI. 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 thankfully acknowledge laboratory 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 software.

REFERENCES

[1] M. Ding and R. Vemuri, "A combined feasibility and performance macromodel for analog circuits," in *Proc. 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," in *Proceedings of International Joint Conference on Neural Networks*, vol. 3, 2002, pp. 2785–2790.

[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] D. Boolchandani, C. Gupta, and V. Sahula, "Analog circuit feasibility modeling using support vector machine with efficient kernel functions," in *Proc. IAENG-ICEE: Design,Analysis and Tools for Integrated Circuits and System*, 18-20 March 2009, pp. 1609–1614.

[6] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Conditionally positive definite kernels for SVM based image recognition," in *Proc. 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," vol. 2, 1998, pp. 85–88.

[9] R. Schwencker, J. Eckmueller, H. Graeb, and K. Antreich, "Automating the sizing of analog cmos circuits by consideration of structural constraints," 1999, pp. 323–327.

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

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

[12] 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.

[13] M. Ding and R. Vemuri, "An active learning scheme using support vector machines for analog circuit feasibility classification," in *Proceedings of IEEE International Conference on VLSI Design*.

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

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

[16] 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.

[17] G. A. Wolfe, *"Performance Macro-Modeling Techniques for Fast Analog Circuit Synthesis"*. Ph.D. thesis, University of Cincinnati, 2004.

[18] V. R. Yelamanchili, *"Compilation and performance estimation of analog and mixed-signal circuits"*. Master's thesis, University of Cincinnati, 2003.

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

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

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

[22] 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.

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

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

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

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

[27] 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.

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