Add name of journal.... manuscript No.
(will be inserted by the editor)

Efficient Kernel Functions for Support Vector Machine Regression Model for Analog Circuit's Performance Evaluation

the date of receipt and acceptance should be inserted later

Abstract Support Vector Machines (SVM) have been widely used for creating fast and efficient performance macro-models for quickly predicting the performance parameters of analog circuits. These models have proved to be not only effective and fast but accurate while predicting the performance. A Kernel function is an integral part of SVM to obtain an optimized and accurate model. There is no formal way to decide, which kernel function is suited to a class of regression problem. While most commonly used kernels are RBF, polynomial, spline, multilayer perceptron; we explored many other un-conventional kernel functions and reported their efficacy and computation efficiency in this paper. These kernel functions are used with SVM regression models and these macromodels are tested on different analog circuits to check for their robustness and performance. We have used SPICE for generating the set of learning data. Least Square SVM toolbox interfaced with MATLAB was used for regression. The models which contained modified compositions of kernels were found to be more accurate and thus have lower mean square error than those containing standard kernels. We have used different CMOS circuits varying in size and complexity as test vehicles- two-stage op amp, cascode op amp, comparator, differential op amp and VCO.

Key terms- Analog synthesis, macromodels, Support Vector Machine, kernel, regression modeling

1 Introduction

In order to characterize an analog system, a set of performance parameters are 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 need to be evaluated to reach a suitable solution. Generally, SPICE is used to obtain performance parameter from circuit simulation, however it is computationally very intensive. An efficient and faster way is to use macromodel, which approximates the relationship between the

Address(es) of author(s) should be given

device sizes and performance parameters. Support vector machine (SVM) regression offers solution for such performance macromodeling. SVMs are class of machine learning approaches. An SVM model can be trained using data generated directly from SPICE. These SVM models are build around suitable kernel functions as regression functions and are able to provide SPICE like 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, whereas support vector machines require an extraction step which is specific to each topology. The only drawback with SVM models is that they are black box models and are unable to reveal even qualitative aspects of system behavior. Thus major effort while formulating SVM model for analog circuit goes into identifying kernel function suited to a particular topology of circuit. However, their biggest advantage is they can be used to readily model hyper-dimensional and non-linear functionality. Support vector machines are typically trained 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 Support Vector Machine for function estimation (Regression)

Our work is based on the theory from [**?**]. Suppose we are given a training data $\{(x_1, y_1), ...(x_k, y_k)\} \subset R^N \times R$, where R^N represents input space. By a certain non-linear mapping ϕ, the training pattern x_t is mapped into some feature space, in which a real valued function $y(x)$ is defined as follows.

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

$\phi(.) : R^n \rightarrow R^{n_h}$ is the mapping to the high dimensional and potentially infinite dimensional feature space.

For the Least-Squares SVM regression error variables for the fitting problem are:

$$e_k = w^T \phi(x_k) + b - y_k \; k = 1, ..., N \tag{2}$$

Given a training set $\{x_k, y_k\}_{k=1}^N$ following optimization problem is formulated in the primal weight space.

$$P : \min_{w,b,e} J_p(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \tag{3}$$

together with the N constraints as given in equation 2. This formulation involves the trade off between a cost function term and a sum of squared errors governed by the trade-off parameter γ. In the regression formalism the term $\frac{1}{2}w^T w$ is no longer related to hyper-plane separation, but instead determines the smoothness of the resulting model. In fact, the primal problem in the LS-SVM formalism is wholly equivalent to a ridge regression problem formulated in the feature space, with parameter γ performing the role of smoothing parameter. Proceeding to the dual Lagrangian-based formulation

$$D: \max_{\alpha} \mathcal{L}(w, b, e; \alpha) \tag{4}$$

$$\mathcal{L} = J_p(w, e) - \sum_{k=1}^{N} \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\} \tag{5}$$

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

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

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

where $[y_1; ...; y_N]$, $1_v = [1; ...; 1]$and $\alpha = [\alpha_1; ...; \alpha_N]$.
The kernel trick is applied here as follows

$$\begin{aligned} \Omega_{kl} &= \phi(x_k)^T \phi(x_l) \\ &= K(x_k, x_l) \qquad k, l = 1, ..., N \end{aligned} \tag{7}$$

The resulting LS-SVM model for function estimation becomes then

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x, x_k) + b \tag{8}$$

where α_k, b are the solution to the linear system given by equation 6.

The function $k(x, x_k)$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 [**?**].

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

$$T_k f(.) := \int_{R^N} k(., x) f(x) d\mu(x) \tag{9}$$

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 $\| \psi_j \|_{L_2} = 1$ 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 $\bar{sup_j} \parallel \psi_j \parallel_{L_\infty} \leq \infty$
3. $k(x,x') = \sum_j \lambda_j \psi_j(x)\psi_j(x')$

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

$$\int_{R^N \times R^N} k(x,x_k)f(x)f(x_k)dxdx_k \geq 0 \text{ for all } f \in L_2(x) \tag{10}$$

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 $k_1(x,x_k), k_2(x,x_k)$ be Mercer kernels and $c_1, c_2 \geq 0$,then

$$k(x,x_k) = c_1 k_1(x,x_k) + c_2 k_2(x,x_k) \tag{11}$$

is also called a Mercer kernel. Moreover, the product of two Mercer kernels is a Mercer kernel, which can be proved on the basis of the equivalent definition of Mercer kernel. This proof can be seen in [**?**]. Similarly, it has been proposed earlier in [**?**] 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.

$$k(x,x_k) = \alpha k(x,x_k) \text{ where } \alpha > 0 \tag{12}$$

$$k(x,x_k) = a * exp(k(x,x_k)) \text{ where } a > 0 \tag{13}$$

We also discuss about two kernels[**?**] here, as they have provided good results for analog design-

– Power kernel given by:

$$k(x,x_k) = - \parallel x - x_k \parallel^\beta \quad and \tag{14}$$

– Log kernel given by:

$$k(x,x_k) = -log(1 + \parallel x - x_k \parallel^\beta) \tag{15}$$

these kernels are conditionally positive definite 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 for accuracy and time of computation. Similar modifications can be carried out for other standard kernels, based on the application. An RBF kernel is given by:

$$K(x,x_j) = e^{-\gamma |x - x_k|^2} \tag{16}$$

2.3 Related Work

Performance macro modeling mainly falls into three categories- knowledge based approaches, symbolic analysis and various regression techniques. Knowledge based methods rely on manual derivation of mathematical equations by expert designers. BLADES [?] is an expert system which uses a custom-designed rule set in order to make design decisions during synthesis. Specific to the design of operational amplifiers, OPASYN [?] relies on decision trees and analytical circuit models designed specifically for each topology considered. Knowledge based system work well, but the effort of encoding the knowledge used is often very high and must be repeated for any new circuit topologies. Symbolic techniques discussed in [?]-[?] and [?] use linearized circuit model equations to perform performance parameter estimation. Symbolic techniques can be automated but the equation can become very large and are usually restricted to purely linear formulation. The equations can be simplified by removing terms that have little effect on accuracy in order to reduce their prohibitive size. The generated equations can provide analog designers with valuable insight into the interaction of design trade offs. However, one drawback is that the performance parameters are not necessarily the direct function of controllable design parameters, rather they are dependent 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 [?], fitting approach to generate symbolic equations [?] and least squares support vector machines [?]. The neural network models provide a great deal of time savings in situations where a fixed topology must be reused and re-synthesized many times. The neural network model are also robust. Numerical instability in SPICE and other circuit simulators can prohibit the acquisition of performance parameters for some of the circuit configurations in the sample space. The neural network models can give estimates of values that the simulator failed to provide. However, there are never any guarantees of absolute accuracy when approximating unknown functions.

Authors in [?] have proposed SVM based performance macromodeling of analog circuits. Modified kernels have been used in SVM algorithm that makes these macromodels more efficient and accurate than those that utilize conventional kernels like RBF. In [?], new kernel has been proposed which has been named as Log kernel. Log kernel along with power kernel have shown to be suitable for SVM algorithm.The authors have compared these kernels with RBF and Laplace kernels for image recognition problem. The log and power kernels have provided good results for pattern classification problems. In Ref [1] the interpolation and extrapolation capabilities of the two main type kernels, namely local and global kernels, were investigated. A mixture of local and global kernels that is convex combination of RBF kernel and polynomial kernel was introduced, resulting in having both good interpolation and good extrapolation abilities. These modifications have resulted in kernel not only with desirable characteristics for learning but also good characteristic for generalization. Further it is reported that SVM using a mixture of kernels will be able not only to learn from the data but also take into account the behavior of a process in the limit. However the mixture of only two kernels have been considered and nonuniform data density in the input space have not been explored. Ref [?] empirically analyzes the robustness of support vector regression(SVR) with different kernels. The robustness of SVR with two typical kernels, polynomial kernel and RBF kernel, and the hybrid is assessed. It is shown that SVR using polynomial kernels with lower degree are more robust, and that using RBF kernels with wider parameters is easily influenced by the noise. The hybrid of different types or scales of kernels can improve the robustness to some degree.

3 Proposed Work

In this section we compare the new kernels described in the Section 2 along-with the standard kernels like RBF and polynomial. The modification shown in equation 12 was done on RBF kernel and equation 13 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 for performance parameters of various circuits was trained using the data generated from HSPICE.

Least Square Support Vector Machine Toolbox [**?**] interfaced with MATLAB was used for function estimation. Toolbox was trained using the data generated from HSPICE. The toolbox provides the values of optimized α and bias as output. These values can be use to estimate the function using equation 8. We can see from equation 8 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 models with different kernels 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. Two stage op amp: The circuit diagram is shown in Figure 1. 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

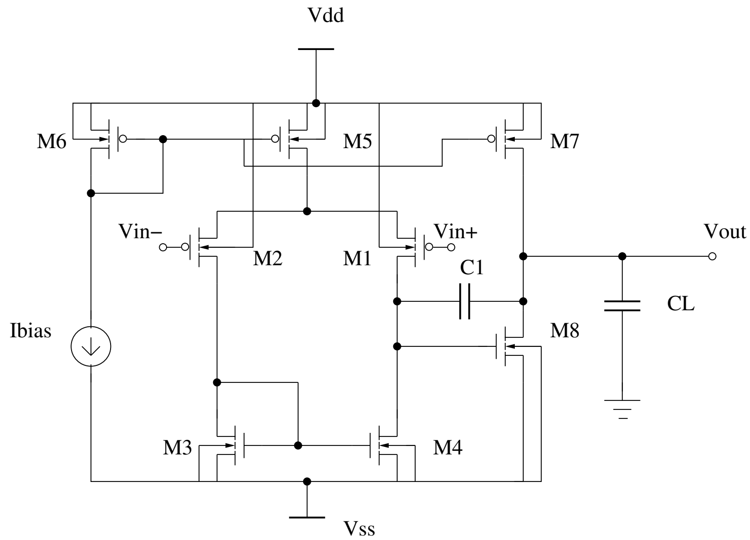

Fig. 1 Two Stage operational amplifier

Table 1 Design Variables and Performance constraints of Two stage Op amp

Design parameters	Geometric constraints	Performance constraints
$W_1 = W_2$	[1μm,100μm]	CMRR$\geq 80dB$
$W_3 = W_4$	[1μm,50μm]	PSRR$\geq 150dB$
W_5	[1μm,100μm]	Phase Margin$\geq 65°$
W_6	[50μm]	Open Loop Gain$\geq$ 20000
W_7	[1μm,100μm]	Unity Gain Frequency$\geq 5 \times 10^6 Hz$
W_8	$2 * W_3 * W_7/W_5$	Slew rate$\geq 6 \times 10^6 V/sec$
$L_1 - L_8$	[1μm]	
Cc	[5pF,20pF]	

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 performance parameters with constraints are shown in Table 1.

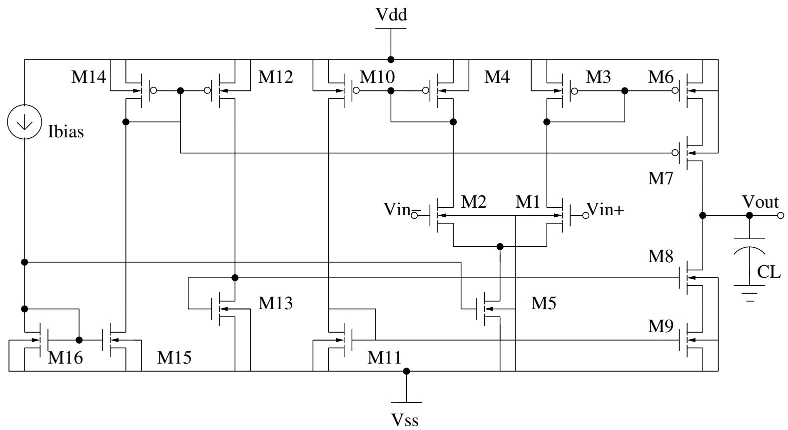

Fig. 2 Cascode op amp

2. Cascode op amp: The circuit of cascode op-amp is shown in Figure 2 We fix the lengths of all transistors to 1μm. Imposing sizing rules [**?**] 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 as well as performance constraints are shown in Table 2. Similar to two stage op amp, SVM models were built for different performance parameters like CMRR, PSRR, phase margin, slew rate and unity gain frequency.

Table 2 Design Variables and Performance Constraints of Cascode op-amp

Design parameters	**Geometric constraints**	**Performance constraints**
$W_1 = W_2$	[1μm, 100μm]	CMRR$\geq 100dB$
$W_3 = W_4$	[1μm, 100μm]	PSRR$\geq 150dB$
$W_5 = W_6$	[1μm,100μm]	Phase Margin$\geq$75°
$Ibias$	[$2\mu A$, $20\mu A$]	Unity Gain Frequency$\geq 1 \times 10^6 Hz$
C_L	[1pF, 10pF]	Slew rate$\geq 1.5 \times 10^6 V/sec$

3. Continuous Time Comparator: The circuit diagram is shown in Figure 3. The width of M1, M2, M3, M4 were taken as design variables. 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 3.

4. Differential Operational Amplifier: The circuit diagram is shown in Figure 4. The design variables chosen are the width of the transistors M1, M2, M3, M4. The slew rate of the differential op amp has been taken as the performance parameter. The design variables are shown in Table 4.

Table 3 Design Variables of Comparator

Design Variables	***Range***
$W_1 = W_2 = W_3 = W_4$	40μm to 160μm
$W_5 = W_6 = W_7 = W_8 = W_9 = W_{14}$	200μm
$W_{10} = W_{12}$	100μm
$W_{11} = W_{13}$	350μm
$L_1 - L_{14}$	1.2μm

Table 4 Design Variables of Differential Op amp

Design Variables	***Range***
$W_1 = W_2 = W_3 = W_4$	30μm to 120μm
$W_5 = W_{10}$	100μm
$W_6 = W_7 = W_8 = W_9$	50μm
$L_1 - L_{10}$	0.2μm

5. Voltage Controlled Oscillator: The circuit diagram is shown in Figure 5. Design variables are taken as widths of transistor M1, M3, M5, M7, M8, M9 and M10. The swing of VCO output and power consumption has been taken as performance parameters. The design variables are shown in Table 5.

Table 5 Design Variables of Voltage Controlled Oscillator

Design Variables	**Geometric constraints**
$W_1 = W_2$	[300μm,400μm]
$W_3 = W_4$	[160μm,300μm]
$W_5 = W_6$	[160μm,300μm]
W_7	[160μm,300μm]
W_8	[160μm,300μm]
$W_9 = W_{11}$	[160μm,300μm]
$W_{10} - W_{12}$	[200μm,300μm]
$L_1 = L_2 = L_7 = L_8 = L_9$	[2μm]
$L_3 = L_4 = L_{11}$	[1μm]
$L_5 = L_6$	[12μm]
$L_{10} = L_{12} = L_{13}$	[6μm]
W_{13}	[1500μm]

4 Results

In this section, we present comparison of performance and accuracy of the kernels in terms of root mean square error in performance parameters and computation time. Table 6 shows the comparison of the performance of the kernels for two stage operational amplifier. We observe substantial 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 accuracy of the kernels considering all the performance parameters. The modifications effected while composing the kernels have resulted in decreased error for all the performance parameters.

Table 7 shows comparison of computational time of HSPICE and the proposed Macromodels. Similar results are shown in Table 8, wherein comparison of accuracy of the kernel is shown for the slew rate of comparator and differential op amp circuit. The results show a better accuracy with the use of multiplied and bias kernel. The error in case of such kernels is much smaller compared to RBF kernel, as RBF kernel fails to estimate an exact function as shown in Figure 6. These plots in Figure 6-8show 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 be seen from the Figure 7 and Figure 8 that log and multiplied kernels follow the data more closely than RBF kernel. However from Table 8, we observe that 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.

However, from Table 9 and Table 10 for different performance parameters of cascode op amp, we observe that Log, Power and Multiplied kernel not only exhibits small root mean square error but also smaller computation time as compared to RBF kernel. Similar results are observed for performance parameters of Voltage controlled oscillator from Table 11.

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 model is concerned. Further extending these efficient kernels approach to the classification problem in analog domain is also being pursued.

References

1. Smits, G.F. and Jordaan, E.M., "Improved SVM regression using mixtures of kernels", In proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 3, 2002, Pages-2785-2790.

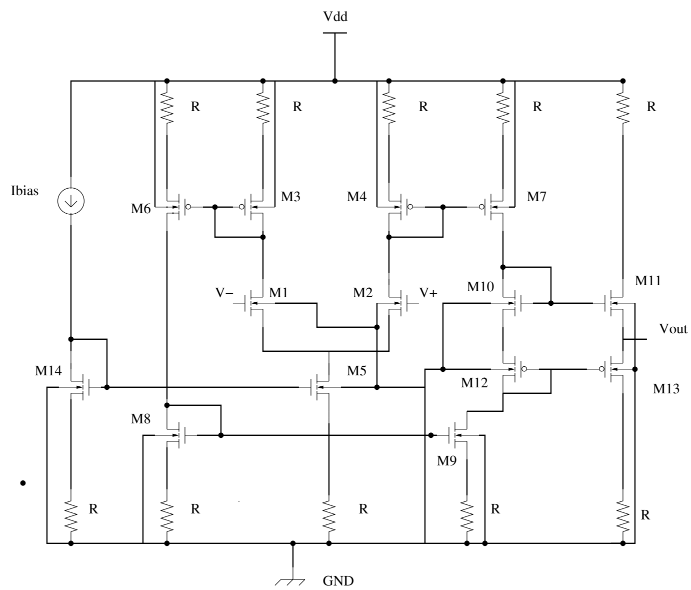

Fig. 3 Continuous Time Comparator

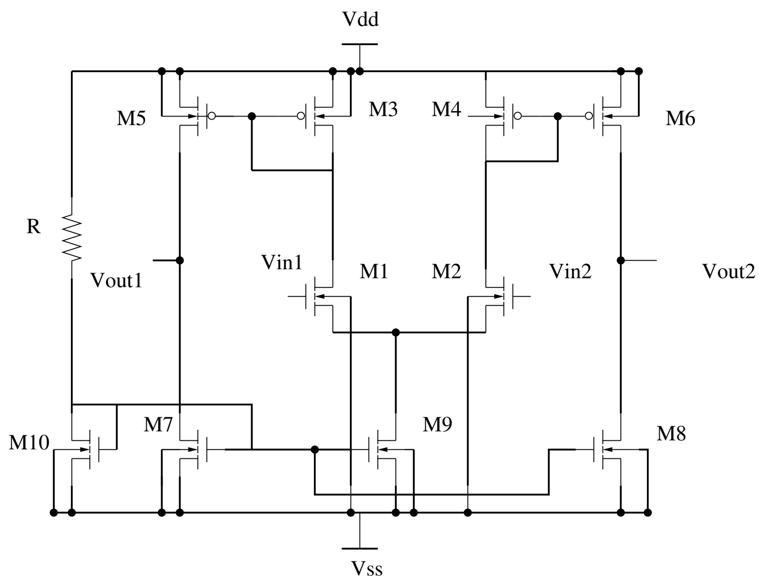

Fig. 4 Differential operational Amplifier

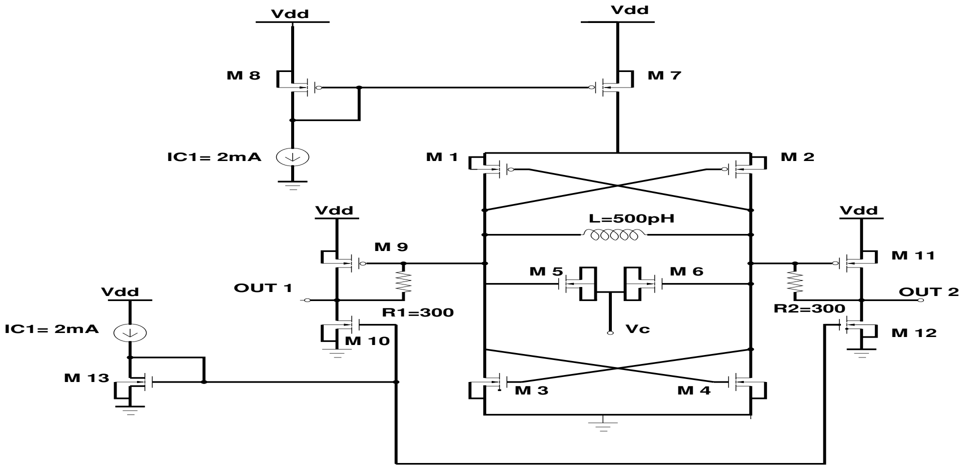

Fig. 5 Voltage Controlled Oscillator

Table 6 Mean Square Error of Performance Parameters for Two Stage Op amp

Kernels	*CMRR*	*OLG*	*Phase Margin*	*PSRR*	*Slew Rate*	*UGF*
RBF	0.013	0.089	0.010	0.007	0.105	0.029
Log	0.009	0.061	0.009	0.003	0.067	0.028
Power	0.009	0.051	0.008	0.003	0.066	0.028
Multiplied	0.008	0.047	0.003	0.055	0.028	0.008
Polynomial	0.013	0.089	0.007	0.105	0.029	.010
Bias	0.011	0.089	0.010	0.007	0.101	0.029

Table 7 Comparing computation times of HSPICE and Macromodels

Performance	*HSPICE (sec)*	*Macromodel (sec)*	*Speed-up*
CMRR	46.89	0.61	76.89
PSRR	12.61	0.65	19.40
PHM	12.51	0.61	20.51
OLG	13.60	0.61	22.29
UGF	14.16	0.65	21.78
SLEW	12.89	0.63	20.46

Table 8 Mean Square Error(MSE) & Computation time for Slew Rate of Comparator and Differential op amp

Kernels	*Accuracy in terms of MSE*		*Computation time in (sec)*	
	Comparator	*Differential Op amp*	*Comparator*	*Differential Op amp*
RBF	0.343	1.060	0.145	0.139
Log	0.217	0.702	0.209	0.199
Power	0.208	0.675	0.172	0.177
Multiplied	0.072	0.061	0.148	0.142
Polynomial	0.343	1.060	0.587	0.539
Bias	0.104	0.165	0.595	0.562

Table 9 Mean Square Error for different Performance Parameters of Cascode op amp

Kernels	*CMRR*	*Phase Margin*	*PSRR*	*Slew Rate*	*UGF*
RBF	0.033	0.107	0.067	0.212	0.156
Log	0.019	0.070	0.060	0.119	0.093
Power	0.016	0.063	0.060	0.103	0.082
Multiplied	0.032	0.102	0.070	0.186	0.150

Table 10 Computation time(in sec) for different Performance Parameters of Cascode op amp

Kernels	*CMRR*	*Phase Margin*	*PSRR*	*Slew Rate*	*UGF*
RBF	12.51	12.02	12.37	11.95	11.99
Log	8.38	8.25	8.47	8.43	8.36
Power	9.44	9.03	8.96	8.88	8.85
Multiplied	4.11	5.63	3.31	5.56	4.36

Table 11 Mean Square Error & Computation time (in sec) for different Performance Parameters of VCO

Kernels	*Accuracy in*	*terms of MSE*	*Computation*	*time in (sec)*
	Voltage Swing	*Power Dissipation*	*Voltage Swing*	*Power Dissipation*
RBF	0.033	0.107	2.52	2.57
Log	0.019	0.070	2.25	2.10
Power	0.016	0.063	2.17	2.12
Multiplied	0.032	0.102	0.95	1.06

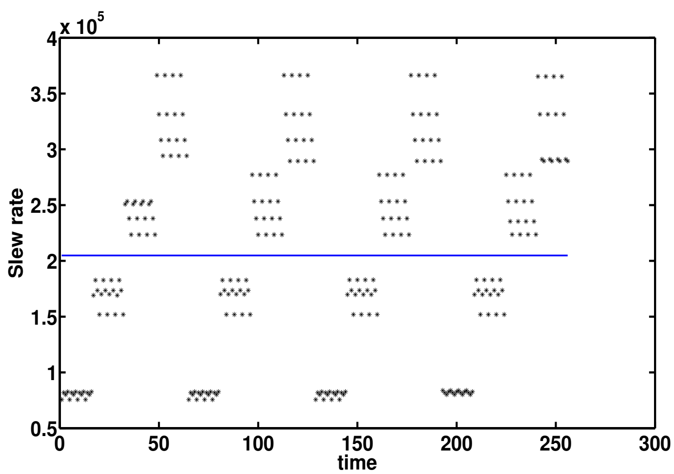

Fig. 6 Function estimation using RBF kernel data points (black *) and estimation (blue line)

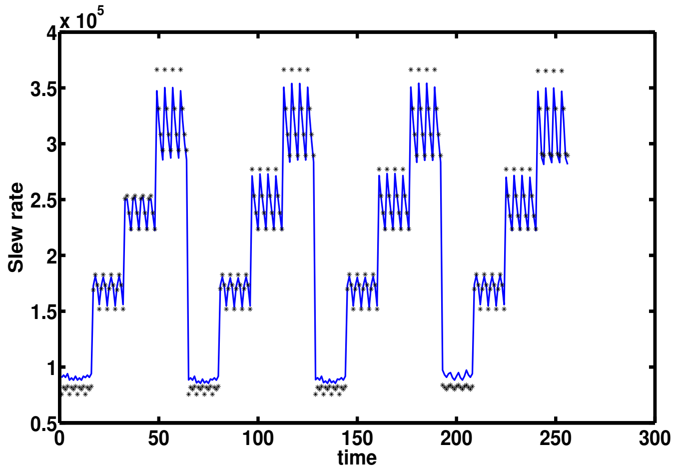

Fig. 7 Function estimation using Log kernel data points (black *) and estimation (blue line)

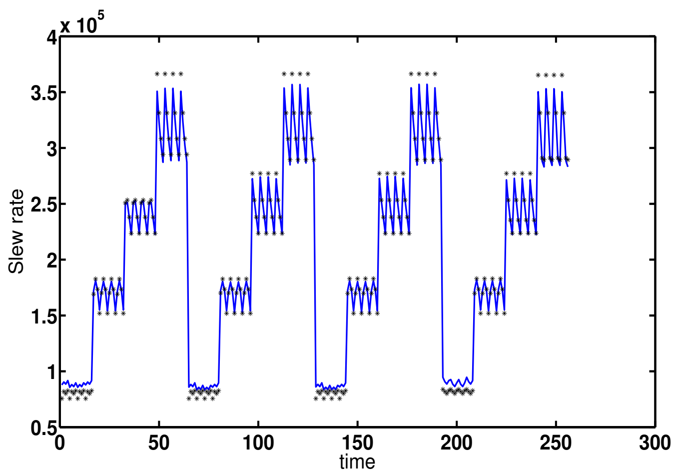

Fig. 8 Function estimation using Multiplied kernel data points (black *) and estimation (blue line)