2009 22nd International Conference on VLSI Design

Systematic Methodology for High-Level Performance Modeling of Analog Systems

Soumya Pandit
Meghnad Saha Institute of Technology
Kolkata, India

Chittaranjan Mandal
Indian Institute of Technology
Kharagpur, India

Amit Patra
Indian Institute of Technology
Kharagpur, India

Abstract— **This paper presents a systematic methodology for construction of high-level performance models using least squares support vector machine. The transistor sizes of the circuit-level implementation of a component block along with a set of geometry constraints applied over them define the sample space. Optimal values of the model hyper parameters are computed using genetic algorithm. The novelty of the methodology is that the models constructed with this methodology are accurate, fast to evaluate with good generalization ability and low construction time. The present methodology has been compared with two other standard methodologies and the novelties are clearly demonstrated with experimental results.**

I. Introduction

An important step of an analog design automation process is analog high-level design. This is defined as the translation of analog system-level specifications into a proper topology of component blocks, in which the specifications of all the component blocks are completely determined so that the overall system meets its desired specifications [1]. There are two broad types of design methodologies available in literature [2] to address the problem of analog high-level design: optimization-based methodology and library-based methodology. An optimization-based analog high-level design methodology has two primary components: a search algorithm and a high-level performance estimator. A high-level performance estimation model is a function that estimates the performance of an analog component block when some high-level design parameters of the block are given as inputs [3].

Analog performance models constructed with regression technique are generally fast to evaluate and the accuracy with respect to real circuit-level simulation results is also good. This technique is therefore, often used for construction of performance models [4], [5], [6]. There are two types of regression-based technique – parametric regression technique and non-parametric regression technique. In parametric regression technique, a parameterised model is first proposed by the model developer and the values of the parameters are then fitted by some least-square error optimisation so that the model response matches closely the response of the real circuit. In non-parametric regression technique, a training network (e.g., support vector machine, artificial neural network) is used that is being trained with SPICE simulation results of the real circuit until the response of the network matches closely enough the response of the real circuit. An important advantage of a non-parametric regression technique over a parametric technique is that it does not require any model template. However, a major limitation of the non-parametric technique is that, the generalization ability of the constructed models is often not good. In addition, the model construction time is generally high which increases the design overhead.

In this work, we have developed a methodology for generation of high-level performance models for analog system using least squares support vector machine (LS-SVM) technique. The novelty of the methodology is that the constructed models are accurate, fast to evaluate with good generalization ability and low construction time. This methodology can be used in conjunction with an optimization procedure to develop a procedure for high-level topology sizing/optimization.

The rest of the paper is arranged as follows. Section II reviews some related works. Section III presents the necessary preliminary concepts. The methodology is described in detail in Section IV. Experimental results are provided in Section V and finally conclusion is drawn in Section VI.

II. Related Work

A fairly complete survey of related works is given in [7]. An analog performance estimation (APE) tool for high-level synthesis of analog integrated circuits is described in [8]. It takes the design parameters of an analog circuit as inputs and determines its performance parameters (e.g., power consumption, thermal noise) along with anticipated sizes of all the circuit elements. A power estimation model for ADC using empirical formulae is described in [3]. The estimators are fast to evaluate. However, the accuracy with respect to real simulation results under all conditions is off by orders of magnitude. The technique for generation of posynomial equation-based performance estimation models for analog circuits like opamps, multistage amplifiers, switch capacitor filters, etc., is described in [9]. An automatic procedure for generation of posynomial models using fitting technique is described in [4]. A neural network based tool for automated power and area estimation is described in [6]. Circuit simulation results are used to train a neural network model, which is subsequently used as an estimator. Fairly recently, support vector machine (SVM) has been used for modeling of performance parameters for RF and analog circuits [10], [5].

1063-9667/09 $25.00 © 2009 IEEE
DOI 10.1109/VLSI.Design.2009.26

III. Preliminaries

A. High-Level Performance Model

Let us consider an analog system defined by a set of specification parameters $\bar{X}$ (e.g., gain, bandwidth of the system) and performance parameters $\bar{\rho}$ (e.g., input referred noise, power consumption). These two parameters are related as

$$\bar{\rho} = \bar{\mathcal{P}}(\bar{X}) \tag{1}$$

where $\bar{\mathcal{P}}$ is referred to as the set of high-level performance models of the system. Note that, a high-level performance model is different from circuit-level performance model in the sense that for high-level model, the input parameters are specification parameters of the component-blocks, whereas for circuit-level performance models, the input parameters are transistor sizes and/or biasing.

The important requirements for a good high-level performance model are: (i) The model needs to be low dimensional. Only those specification parameters are to be considered as inputs which have dominant contributions on a performance parameter to be estimated. (ii) The predicted results need to be accurate. The model accuracy is measured as the deviation of the model predicted value from the true function value. The function value in this case is the performance parameter obtained from transistor level simulation. (iii) The evaluation time must be short. This is measured by the CPU time required to evaluate a model. (iv) The time required to construct an accurate model must be small, so that the design overhead does not become high. This is relatively harder to quantify. This process involves both applying design knowledge to setup testbench circuit and design variable selection and computational time needed to use an algorithm to train a model. As a rough estimate, the construction cost is measured as

$$T_{\text{construction}} = T_{\text{data generation}} + T_{\text{training}} \tag{2}$$

where the terms are self explanatory. There exists a tradeoff between these requirements since a model with lower prediction error generally takes more time for construction and evaluation.

B. Least Squares Support Vector Regression

In recent years, the support vector machine (SVM), as a powerful new tool for data classification and function estimation, has been developed [11]. Suykens and Vandewalle [12] proposed a modified version of SVM called least squares SVM. In this subsection, we briefly outline the theory behind the LS-SVM as function regressor.

Consider a given set of training samples $\{x_k, y_k\}_{k=1,2,...,N_{tr}}$ where x_k is the input value and y_k is the corresponding target value for the k^{th} sample. With a SVR, the relationship between the input vector and the target vector is given as

$$\hat{\bar{y}}(x) = w^T \phi(x) + b \tag{3}$$

where ϕ is the mapping of the vector $\bar{x}$ to some (probably high-dimensional) feature space, b is the bias and w is the weight vector of the same dimension as the feature space. The mapping $\phi(\bar{x})$ is generally non-linear which makes it possible to approximate non-linear functions. The approximation error for the k^{th} sample is defined as

$$e_k = y_k - \hat{y}_k(x_k) \tag{4}$$

The minimization of the error together with the regression is given as

$$\min \ \mathcal{J}(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N_{tr}} e_k^2 \tag{5}$$

with equality constraint

$$y_k = w^T \phi(x_k) + b + e_k, \ \ k = 1, 2, ..., n \tag{6}$$

where N_{tr} denotes the total number of training data sets and the suffix k denotes the index of the training set, i.e., k^{th} training data, γ is the regularization parameter. LS-SVM considers the optimization problem to be a constrained optimization problem and uses dual Lagrangian-based formulation

$$\mathcal{L} = \mathcal{J}(w, e) - \sum_{k=1}^{N_{tr}} \alpha_k \left(w^T \phi(\bar{X}_k) + b + e_k - \rho_k \right) \tag{7}$$

and applying 'kernel trick', we arrive at the final model [12]

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

where $K(x_k, x)$ is the kernel function. The elegance of using the kernel function lies in the fact that one can deal with feature spaces of arbitrary dimensionality without having to compute the map $\phi(\bar{x})$ explicitly. The Gaussian kernel function defined as

$$K(x_k, x) = exp \left(- ||x_k - x| \, |^2 / \sigma^2 \right) \tag{9}$$

is used in the present work, where σ^2 denotes the kernel bandwidth.

IV. High-Level Performance Model Generation

In this section, we describe the various steps of the construction methodology in detail.

A. Sample Space and Design of Experiments

While choosing the set of inputs, only those specification parameters forming a set $\bar{X}' \subseteq \bar{X}$ which have dominant contributions to specific performance parameters $\bar{\rho} = \{\rho_1, \rho_2, ..., \rho_n\}$ are considered as inputs. This choice of inputs relies on the designer's knowledge depending upon the application system and the topology considered. The dominant specification parameters are referred to as the high-level design parameters. For ease of notation, the prime indicating the reduction is omitted in the rest of this paper. Both the inputs and the output of the performance model $\mathcal{P}$ are taken to be functions of a set of geometry parameters $\bar{\alpha}$ (transistor sizes) of a component block, expressed as

$$\bar{X} = \mathcal{R}(\bar{\alpha}) \tag{10}$$

$$\bar{\rho} = \mathcal{Q}(\bar{\alpha}) \tag{11}$$

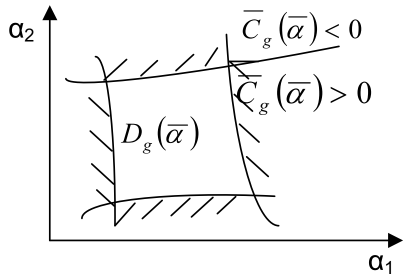

Fig. 1. 2D projection of a four dimensional sample space.

$\mathcal{R}$ and $\mathcal{Q}$ represents the mapping of the geometry parameters to electrical parameters. The multidimensional space spanned by the elements of the set $\bar{\alpha}$ is defined as circuit-level design space $\mathcal{D}_\alpha$.

A set of geometry constraints is applied on the transistor sizes to enclose a region within $\mathcal{D}_\alpha$, from which samples are extracted for training data generation. These geometry constraints include equality constraints as well as inequality constraints. The equality constraints, expressed as algebraic equations directly correlate the transistor sizes. For example, for matching purpose, the sizes of a differential pair transistors are equal. The equality constraints eliminate elements of the set $\bar{\alpha}$ and therefore reduce the dimension of the circuit-level design space $\mathcal{D}_\alpha$. The inequality constraints exclude additional portion of the reduced design space $\mathcal{D}_\alpha$, (correct notation is $\mathcal{D}_{\alpha'}$, which we avoid for ease of notation) without further reducing its dimension. The inequality constraints are usually given as box constraints, i.e., in the form of lower bounds and upper bounds. The lower bounds are determined by the feature size of a technology. The upper bounds are selected such that the transistors are not excessively large. With elementary algebraic transformations, all the geometry constraints are combined into a single non-linear vector inequality, which is interpreted element wise as:

$$\bar{C}_g(\bar{\alpha}) \geq 0 \ \Leftrightarrow \forall_{i\in\{1\ldots q\}} C_{gi}(\bar{\alpha}) \geq 0 \tag{12}$$

These constraints as functions of $\bar{\alpha}$ define a space, which we call as a sample space $\mathcal{D}_g$, defined as

$$\mathcal{D}_g = \{\bar{\alpha}\,|\,\bar{C}_g(\bar{\alpha}) \geq 0\} \tag{13}$$

Clearly $\mathcal{D}_g \subset \mathcal{D}_\alpha$. A two dimensional projection of a four dimensional sample space is illustrated in Fig. 1. Within the sample space, the circuit performance behavior becomes weakly non-linear [13]. Therefore, simple sampling strategies are used to construct models with good generalization ability.

The transistor sizes for generating training data corresponding to $\bar{X}$ and $\bar{\rho}$ are restricted to $\mathcal{D}_g(\bar{\alpha})$. The data generation process is generally an expensive process. Strategies from design of experiments (DOE) provide a mathematical basis to select a limited but optimal set of sample points from the sample space for training data generation. In the present work, these points are generated using a Halton sequence generator. A Halton sequence generator is a quasi-random number generator which generates a set of uniformly distributed random points in the sample space. This ensures a uniform and unbiased representation of the sample space.

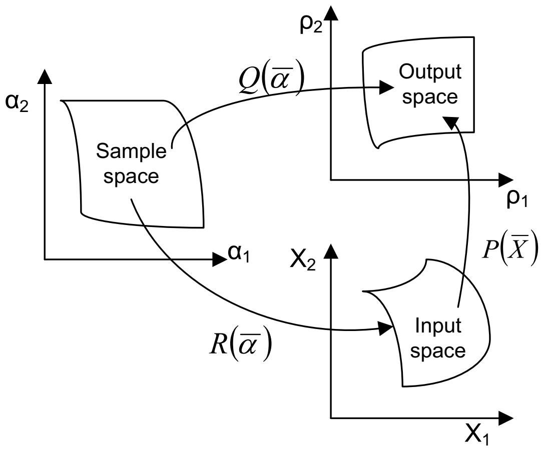

Fig. 2. Non-linear relation between the sample space and the input, output space.

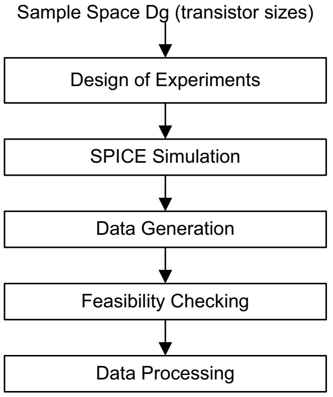

Fig. 3. An outline of the procedure for generation of training data.

B. Training Data Generation

From (10) and (11), we see that the inputs $(\bar{X})$ and output $(\bar{\rho})$ of a high-level performance model $\mathcal{P}$ are functions of transistor-level parameters $\bar{\alpha}$. The inputs and the outputs are electrical parameters, whereas $\bar{\alpha}$ is a set of geometry parameters. The functions $(\mathcal{R}, \mathcal{Q})$ for mapping the geometry parameters to the electrical parameters are complex non-linear functions, considering the deep submicron effects of MOS transistors. In this work, these are achieved element-wise through a circuit simulation process, which is accepted to be the most accurate technique. The relationships are illustrated in Fig. 2. $\mathcal{R}$ and $\mathcal{Q}$ are used for generating the training data and $\mathcal{P}$ is the performance model to be constructed.

The training data generation process is outlined in Fig. 3.

For each input sample (transistor sizes) extracted from the sample space $\mathcal{D}_g$, the chosen circuit topology of a component block is simulated using SPICE through Cadence Spectre tool. The BSIM3v3 model is used for simulation, ensuring that the important deep submicron effects are considered while generating the training set. Depending upon the selected input-output parameters of an estimation function, it is necessary to construct a set of test benches that would provide sufficient data to facilitate automatic extraction of these parameters via postprocessing of SPICE output files. The commonly used SPICE analysis are ac analysis, transient analysis, dc sweep etc. The voltages and currents at the various nodes of the circuit are also measured. A set of constraints, referred to as feasibility constraints is then considered to ensure that only feasible data are taken for training.

The generated input-output data are considered to be feasible, if either they themselves satisfy a set of constraints or the mapping procedures $(\mathcal{R}, \mathcal{Q})$ through which they are generated satisfy a set of constraints. The constraints are as follows [13]:

1) Functionality constraints C_f : These constraints are applied on the measured node voltages and currents. They ensure correct functionality of the circuit and are expressed as

$$C_f = \{f_k(v, i) \geq 0 \;\; k = 1, 2, ..., n_f\} \tag{14}$$

For example, the transistors of a differential pair must work in saturation.

2) Performance constraints C_p : These are applied directly on the input-output parameters, depending upon an application system. These are expressed as

$$C_p = \{f_k(\bar{\rho}) \geq 0 \;\; f_k(\bar{X}) \geq 0 \;\; k = 1, 2, ..., n_p\} \tag{15}$$

For example, the phase margin of an opamp must be greater than 45^0.

The total set of constraints for feasibility checking is thus $C = \{C_f \cup C_p\}$.

Data scaling is an essential step to improve the learning/training process of SVMs. The present methodology employs both linear scaling as well as logarithmic scaling depending upon the parameters chosen.

C. LS-SVM Hyperparameter determination

To obtain good performances, some parameters in the SVM models have to be chosen carefully. These parameters include: (i) the regularization parameter γ, which determines the trade-off between minimizing the training error and minimizing the model complexity and (ii) parameter (σ^2) of the kernel function that implicitly defines the non-linear mapping from the input space to some high-dimensional feature space. These higher level parameters are usually referred to as hyper parameters. In general, in any SVM problem, if the hyper parameters of the model are not well selected, the predicted results will not be good enough and the generalization ability will also be poor. Tuning of these hyper parameters is usually done by minimizing the estimated generalization error. The generalization error is a function that measures the generalization ability of the constructed models, i.e., the ability to predict correctly the performance of an unknown sample. In the present methodology, hold-out technique is used for estimating the generalization error. This is a simple technique for estimating the generalization error. The data set is separated into two sets, called the training set and the testing set. The SVM is constructed using the training set only. Then it is tested using the test data set. The test data are completely unknown to the estimator. The errors it makes are accumulated to give the mean test set error, which is used to evaluate the model.

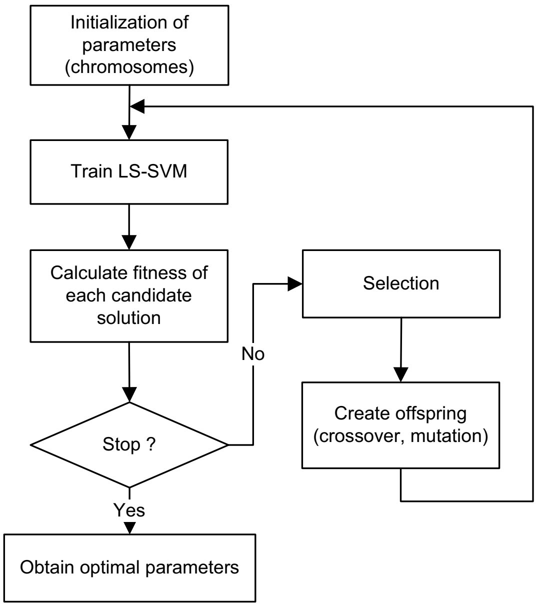
Initialization of parameters (chromosomes)
Train LS-SVM
Calculate fitness of each candidate solution
Selection
No
Create offspring (crossover, mutation)
Stop ?
Yes
Obtain optimal parameters

Fig. 4. Outline of GA-based hyperparameter selection procedure

The present methodology uses genetic algorithm (GA)-based technique for determining optimal values of the model hyperparameters. The task of selection of the hyper parameters is same as an optima searching task, and each point in the search space represents one feasible solution (specific hyper parameters). An outline of the GA-based process is shown in Fig. 4. The chromosomes consist of two parts, $\log_2 \gamma$ and $\log_2 \sigma^2$. During the evolutionary process of GA, a model is trained with the current hyper parameter values. The fitness of the chromosomes depends on the average relative error (ARE) calculated over the test samples. The fitness function is defined as

$$\text{fitness} = \frac{1}{ARE(\gamma, \sigma^2)} \tag{16}$$

Thus, maximizing the fitness value corresponds to minimizing the predicted error. The ARE function is defined as

$$ARE = \frac{1}{N_{te}\rho'} \sum_{1}^{N_{te}} (\rho' - \rho) \tag{17}$$

Here N_{te}, ρ and ρ' are the number of test data, the SVM estimator output and the corresponding SPICE simulated value,

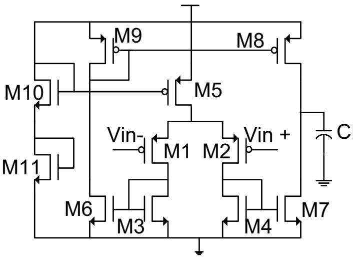

Fig. 5. PMOS OTA circuit

TABLE I

TRANSISTOR SIZES AND FEASIBILITY CONSTRAINTS FOR OTA

	Parameters	Ranges
Transistor Sizes Geometry Constraints	$W_1 = W_2$	$[280nm, 400\mu m]$
	$W_3 = W_4 = W_6 = W_7$	$[1\mu m, 20\mu m]$
	$W_8 = W_9$	$[280nm, 10\mu m]$
	W_5	$[1\mu m, 50\mu m]$
	C_L	$[1pF, 10pF]$
Functional Constraints	Parameters	Range
	$V_{gs} - V_{th}$	$\geq 0.1V$
	V_{op}	$\approx 0.9V$
	V_{off}	$\leq 2mV$
Performance Constraints	Slew rate	$[0.1V/\mu s, 20V/\mu s]$
	Bandwidth	$\geq 2MHz$
	DC Gain	$\geq$ 70 dB
	Phase margin	$[45^0, 60^0]$

respectively. The fitness of each chromosome is taken to be the average of five repetitions. This reduces the stochastic variability of the model training process in GA-based LS-SVM.

D. Quality Measures

Statistical functions are generally used to assess the quality of the generated estimator. The ARE function defined in (17) is one such measure. Another commonly used measure is the correlation coefficient. This is defined as follows:

$$R = \frac{N_{te} \sum \rho\rho' - \sum \rho \sum \rho'}{\sqrt{\left[N_{te} \sum \rho^2 - (\sum \rho)^2\right] \left[N_{te} \sum \rho'^2 - (\sum \rho')^2\right]}} \quad (18)$$

The correlation coefficient is a measure of how closely the LS-SVM outputs fit with the target values. It is a number between 0 and 1. The higher the correlation coefficient, the better it is.

V. EXPERIMENTAL RESULTS

In this section, we provide experimental results demonstrating the methodology described above. The entire methodology has been implemented in Matlab environment and the training of the LS-SVM has been done using Matlab toolbox [14].

A. Experiment 1

A two stage CMOS operational transconductance amplifier (OTA) is shown in Fig. 5. The technology is $0.18\mu m$ CMOS process, with a supply voltage of $1.8V$. The transistor level parameters along with the various feasibility constraints are listed in Table I. We consider the problem of modeling input referred thermal noise (ρ_1), power consumption (ρ_2) and output impedance (ρ_3) as functions of DC gain (X_1), bandwidth (X_2) and slew rate (X_3). From the sample space defined by the transistor sizes, a set of 5000 samples is generated using a Halton sequence generator. These are simulated through ac analysis, operating point analysis, noise analysis and transient analysis using SPICE program. Out of all samples, only 1027 samples are found to satisfy the functional and performance constraints listed in Table I.

TABLE II

HYPER PARAMETER VALUES AND QUALITY MEASURES

Model	σ^2	γ	$ARE(\%)$		R		T_{tr}
			Training	Test	Training	Test	(min)
ρ_1	2.38	250.13	1.82	2.48	0.999	0.998	12.06
ρ_2	5.62	480.19	2.12	3.82	0.994	0.961	10.83
ρ_3	5.19	140.15	1.98	2.90	0.999	0.998	11.56

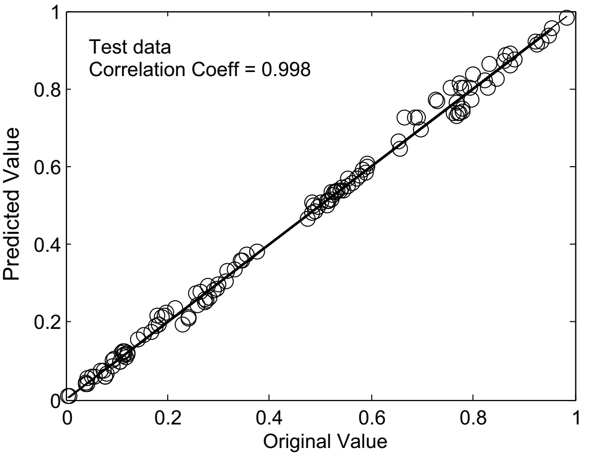

Fig. 6. Scatter plot of estimated and original values for the noise model with normalized test data.

The estimation functions are generated using LS-SVM technique. The generalization errors are estimated through the hold-out method. The hyper parameters are computed through the GA-based technique. The population size is taken to be ten times the number of optimization variables. The crossover probability and the mutation probability are taken as 0.8 and 0.05 respectively. These are determined through a trial and error process. The hyper parameter values and the quality measures of the constructed models are reported in Table II. We find that the constructed models are quite accurate with average relative generalization ability error less than 4%.

The scatter plot of SPICE-simulated and LS-SVM estimated values for normalized test data of the noise model is shown in Fig. 6. The scatter plot illustrates the correlation between the SPICE simulated and the LS-SVM estimated test data. The correlation coefficient is very close to unity. Perfect accuracy would result in the data points forming a straight line along the diagonal axis.

TABLE III
COMPARISON BETWEEN OUR METHOD AND [5]

Model	Method	$ARE(\%)$		R	
		Training	Test	Training	Test
ρ_1	Our	1.82	2.48	0.999	0.998
	[5]	2.86	6.48	0.999	0.875
ρ_2	Our	2.12	3.82	0.994	0.961
	[5]	3.32	7.18	0.980	0.800
ρ_3	Our	1.98	2.90	0.999	0.998
	[5]	2.02	5.14	0.999	0.937

TABLE IV
COMPARISON BETWEEN OUR METHODOLOGY AND ESTEMATE [6]

Method	# Samples		$ARE(\%)$		Generation time	Training time
	Training	Test	Training	Test		
Our	821	206	2.12	3.82	14 min	10.83 min
[6]	2564	641	2.88	6.53	10 hour	21 min

B. Experiment 2

In this experiment, we provide a comparison between our methodology of developing a performance model and that presented in [5]. The model hyper parameters are determined in [5] through heuristic technique. The same performance models are used for comparison purpose. The comparison results with respect to average relative generalization error (ARE), correlation coefficient (R) are reported in Table III. We observe from the comparison results that the generalization ability of the model constructed with our methodology is better than that constructed through [5] technique. This is because of the optimal choice of LS-SVM hyper parameters in our methodology through an optimization process.

C. Experiment 3

Here we present a comparison between our methodology and the EsteMate technique [6]. The power consumption model is reconstructed using the EsteMate technique. A set of 5000 samples is considered. For each selected sample, an optimal sizing is performed with a simulated annealing-based optimization procedure and standard analytical equations. The performances of each configuration is measured within the optimization process and are checked for feasibility. Out of all samples, only 3205 samples are accepted and the rest are rejected. The determination of the training set took 10 hours of CPU time. The training is done through an artificial neural network structure with two hidden layers. The comparative results are shown in Table IV. The generalization ability of our methodology is better than that of [6]. This is becuase of the use of SVM in our methodology. The generalization ability of SVM is found to be better than that of neural network [12]. In EsteMate, for each sample, a complete circuit sizing task using a global optimization algorithm is required for generation of the training data. This is usually prohibitively time consuming. On the other hand in our method, simple circuit simulations using the sampled transistor sizes are required for data generation. Feasibility checking are done afterwards Therefore, the cost of training data generation in our method is much less compared to that in the EsteMate methodology. This is evident from the experimental results also.

VI. CONCLUSION

This paper presents a systematic methodology for generation of analog high-level performance model using LS-SVM. The transistor sizes along with a set of feasibility constraints applied over them define the sample space. The SVM hyper parameters are determined through GA-based optimization technique. The quality of the constructed models is estimated by comparing the predicted performances with actual circuit-level simulation results. The novelty of present methodology is that the models constructed with this methodology are accurate, fast to evaluate with good generalization ability and low construction time. The methodology has been compared with other standard methodologies and the advantages of our methodology have been demonstrated with experimental results. The current methodology can be used in conjunction with an optimization procedure to develop a procedure for high-level topology sizing/optimization.

REFERENCES

[1] Georges.G.E. Gielen. Modeling and Analysis Techniques for System-Level Architectural Design of Telecom Front-Ends. *IEEE Trans. MTT*, Vol.50:pp.360–368, January 2002.

[2] S.Donnay et.al. High-level synthesis of analog sensor interface front-end. In *Proc. of ED&TC*, pages 56–60, 1997.

[3] E. Lauwers and Georges Gielen. Power Estimation Methods for Analog Circuits for Architectural Exploration of Integrated Systems. *IEEE Trans. VLSI Systems*, Vol.10:pp.155–162, April 2002.

[4] W. Daems, G. Gielen, and W. Sansen. Simulation-Based Generation of Posynomial Performance Models for the Sizing of Analog Integrated Circuits. *IEEE Trans. CADICS*, Vol.22:pp.517–534, May 2003.

[5] T. Kiely and G. Gielen. Performance Modeling of Analog Integrated Circuits using Least-Squares Support Vector Machines. In *Proc. of DATE*, pages 448–453, Feb 2004.

[6] G.V. Plas, J. Vandenbussche, G. Gielen, and W. Sansen. EsteMate: A Tool for Automated Power and Area Estimation in Analog Top-down Design and Synthesis. In *Proc. of CICC*, pages 139–142, May 1997.

[7] Rob.A. Rutenbar, Georges.G.E. Gielen, and J.Roychowdhury. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proceedings of the IEEE*, Vol.95:pp.640–669, March 2007.

[8] A. Doboli, N. Dhanwada, A. Nunez-Aldana, and R. Vemuri. A Two-Layer Library-Based Approach to Synthesis of Analog Systems from VHDL-AMS Specifications. *ACM Trans. DAOES*, Vol.9:pp.238–271, April 2004.

[9] P. Mandal and V. Visvanathan. CMOS Op-amp Sizing Using a Geometric Programming Formulation. *IEEE Trans. CADICS*, Vol.20:pp.22–38, January 2001.

[10] X. Ren and T. Kazmierski. Performance Modeling and Optimization of RF Circuits using Support Vector Machines. In *Proc. of MIXDES*, pages 317–321, 2007.

[11] V Vapnik. *Statistical Learning Theory*. Springer, New York, 1998.

[12] J.A.K Suykens, T.V Gestel, J.D Brabanter, B.D Moor, and V. Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[13] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The Sizing Rules Method for Analog Integrated Circuit Design. In *IEEE/ACM ICCAD*, pages 343–349, 2001.

[14] LS-SVM Toolbox. http://www.esat.kuleuven.ac.be/sista/lssvmlab, February 2003.