A Combined Feasibility and Performance Macromodel for Analog Circuits

Mengmeng Ding
ECECS Department
University of Cincinnati
Cincinnati, OH, 45220, USA
dingm@ececs.uc.edu

Ranga Vemuri
ECECS Department
University of Cincinnati
Cincinnati, OH, 45220, USA
ranga@ececs.uc.edu

ABSTRACT

The need to reuse the performance macromodels of an analog circuit topology challenges existing regression based modeling techniques. A model of good reusability should have a number of independent design parameters and each parameter can vary in a large numeric range. On the other hand, these requirements can cause a large percentage of functionally incorrect designs in the design space and thus results in a sparse feasible design space. They also complicate the mathematical relationship between the performance parameters and the design parameters. In order to tackle these challenges, this paper presents a combined feasibility and performance macromodel based on Support Vector Machines (SVMs). The feasibility model identifies the feasible designs that satisfy the design constraints. The performance macromodel is valid for feasible designs. Feasibility macromodeling is formulated as a classification problem while performance macromeling as a regression problem. An active learning scheme [5] has been applied to improve the accuracy of the feasibility model much faster than only using uniformly distributed designs in the entire design space. Our experiment shows that the performance macromodels in the feasible design space are more accurate and faster to construct and evaluate than performance macromodels in the entire design space without functional or performance constraints considered.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Algorithms, Design, Performance

Keywords

Performance macromodeling, feasibility models, active learning

1. INTRODUCTION

Performance macromodels are mathematical models that approximate the relationship between controllable design parameters and performance parameters. One major characteristic of performance macromodels is their short evaluation time compared to a circuit level simulator. Hence, they usually replace the circuit level simulator both to enhance early design space exploration [6], [1], [12] and to accelerate circuit sizing [7], [9]. The other characteristic is reusability. Performance macromodels for a certain circuit topology can be used to size the same circuit for various applications to meet different performance specifications. Short evaluation time and good reusability are also the requirements for performance macromodels.

Previous work in performance macromodeling mainly falls into three categories:

- *Knowledge based approach* relies on manual derivation of mathematical equations by expert designers.
- *Symbolic analysis* techniques generate system transfer functions automatically. Symbolic equations for performance parameters need to be further generated. Either knowledge based approach or symbolic approach requires the numerical values of the small signal parameters that appear in the equations and thus a simple device model such as SPICE Level 1 model or a circuit level simulator is needed. Furthermore, symbolic approach is not readily to be applied in large signal behavior of the circuits.
- *Regression* techniques have gained more and more research interest lately, including artificial neural networks [13], fitting approach to generate symbolic equations [3] and least squares support vector machines [8]. The common strategy involves using a regressor and samples of performance parameters from simulations to fit the regressor. The modeling process is somewhat automated and reusability of the generated models mainly depends on the design space in which these models are applicable. Performance macromodels generated by regression techniques are, however, valid only in the design space defined during model generation. In most previous work [13], [3], designers manually define the design space by selecting the design variables and constraining them between lower and upper bounds.

S. Zazala, J. Eckmuller and H. Grab [14] appear to be the first to define and calculate the feasible design space. They define *feasible*

[0] This work was supported in part by the Defense Advanced Research Projects Agency and the Sensors Directorate of the Air Force Research Laboratory, U.S. Air Force, Wright-Patterson AFB, OH, under Contract F33615-01-C-1977, the Dayton Area Graduate Studies Institution (DAGSI) under the DAGSI/AFRL Research Program Project SN-UC-01-08, and the National Science Foundation under award number CCF-0429717.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

design space as a multidimensional space in which every design satisfies certain design constraints. They calculated the feasible design space by linear approximation. Furthermore, they reduced the complexity of their representation of the feasible design space by eliminating approximately collinear constraints. The major contribution of their work is that they provide a hierarchical way of posing function and robustness constraints on a circuit topology, primarily opamps. It is the first systematic attempt to address the feasible design space of performance macromodels, although the approximation is somewhat poor, resulting in 70% overall accuracy. Stehr et al. [10] further approximated this feasible design space via maximum volume ellipsoid algorithm and applied it as an initial sizing algorithm and used as sizing constraints in later sizing. The approximation might be poorer than the original polytope approximation because it is a further approximation of the original approximation.

This paper will present a combined feasibility and performance macromodel of much higher accuracy. Similar to [14], the feasibility model identifies the feasible design space. The performance models are defined and valid in the feasible design space. In contrast to [14], feasible design space identification problem is treated as a two class *classification* problem and an active learning scheme [5] using Support Vector Machines (SVMs) is used to tackle the sparsity of the feasible designs. Experimental results will show that our feasibility model has an overall accuracy of 99%. We then build performance macromodels using simulation data for feasible designs. Our performance models are highly accurate even in the worst case in terms of the maximum modeling error on the training and validation data set.

The rest of the paper is organized as follows. Section 2 defines the feasible design space identification problem and performance macromodeling. Section 3 gives a brief overview of SVM classifier and regressor. In Section 4, we give a brief overview of the active learning scheme [5] for feasibility modeling. Section 5 shows the experimental results of the proposed methodology. Section 6 and Section 7 discusses the applications and the limitations of the proposed methodology respectively. Section 8 concludes this paper.

2. PROBLEM DEFINITION

2.1 Feasibility Design Space

As mentioned in Section 1, feasible design space is multidimensional and determined by certain design constraints. Given a circuit topology, we can pose three types of constraints [7], [5].

- *Geometry constraints*, C_g, are posed directly on the designer controllable parameters. Only one design variable is assigned to the matched devices. After matching is taken into account, the design parameters are abstracted into a vector of independent design variables $X = \{X1, \cdots, Xn\} \in R^n$. The constraints on the design variables usually are given in the form of lower and upper bounds.

$$C_g = \{lb_i \leq Xi \leq ub_i, \quad i = 1 \cdots n\} \tag{1}$$

- *Functional constraints*, C_f, ensure the functionally correct behavior of the given circuit topology. They are 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.
- *Performance constraints*, C_p, are posed on the performance measurements p based on the applications.

$$C_p = \left\{lb_i \leq p_i \leq ub_i, \quad i = 1 \cdots N_p\right\} \tag{2}$$

While these three types of constraints are widely used in analog circuit sizing, their importance in the scenario of *performance macromodeling* deserves the attention. Device size ranges and functional constraints are useful in defining the feasible design space for they ensure the correct behavior of certain circuit [14]. While performance constraints have never been applied in performance macromodel development, we are enabled to do so by formulating the feasibility modeling problem into a classification problem and thus to handle arbitrary constraints. Take phase margin as an example, in most applications an opamp is to have a minimum phase margin of 45°. We only need to build a macromodel valid for phase margin greater than 45°. Posing performance constraints can further lead to design space reduction and possible complexity and accuracy improvement of the performance macromodels.

The feasible design space $I \subseteq R^n$ is defined in Equation 3. Note that X is a vector of all design variables.

$$I = \{X \,|\, C, X \in \mathbb{R}^n\}, C = C_g \cup C_f \cup C_p \tag{3}$$

Since some constraints are not directly posed on the design variables, I is not in an analytic form but on implicit electrical values. We thus define a feasibility function $y(X)$ whose output only takes two values $\{+1, -1\}$ depending on whether $X \in I$ [5].

$$y(X) = \begin{cases} +1 & \text{if} \quad X \in I \\ -1 & \text{if} \quad x \notin I \end{cases} \tag{4}$$

2.2 Performance Macromodels

We restrict our discussion to regression techniques for performance macromodeling in the following discussion. For this type of technique, it is usually hard to build a performance macromodel valid in the entire design space defined by the geometry constraints. There are several reasons for this. Firstly, the simulator is sometimes unable to measure certain performance parameters when the circuit is not biased correctly. Secondly, even when the simulator is able to measure performance parameters, the circuit can still be biased incorrectly. For example, when Hspice finds the AC performance parameters such as gain or phase margin, the simulation is performed on the linearized circuit instead of the original circuit. We can find a design with good performance parameters but the transistors are in cutoff region! Thirdly, it is possible that the circuit performance parameters are highly nonlinear with respect to the design variables. On the other hand, we want to start with a large design space defined by the geometry constraints. The larger this design space is, the more reusable the performance macromodels will be. We pose constraints to make sure that simulation data of the functionally incorrect designs is not used to build performance models. We also pose constraints on performance parameters to further reduce the design space so that accurate performance models can be built. Our intention is to build *accurate* performance macromodels for a reduced design space rather than have inaccurate models even for a large design space. Our performance macromodels are thus defined in the feasible design space I.

In this paper, we use Vapnik [11] proposed SVMs as our regressor. The original SVM regressor uses a ε-insensitive loss function which is equivalent to minimizing the maximum error. Often the loss function of sum of squares fails to detect large errors. We want to make sure the models will perform well even in the worst case. Thus minimizing the maximum error is favored here. Another advantage of SVM regressor is that it does not require a regular data format required by many Design of Experiments (DoE) techniques. The shape of feasible design space is unknown in the design parameter space to our knowledge. Thus techniques from DoE are not applicable here.

3. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) were initially proposed by Vapnik [11] to solve the learning problem of "finding a desired dependence using a limited number of observations"in 1992 . Let x denote the vector in the input space R^n and y denoted the output value generated by the unknown dependence, a set of l observations $(x_1, y_1), \ldots, (x_l, y_l)$ is called training data. Two main learning problems are classification and regression. In the classification problem, y takes only two values $y = \{+1, -1\}$. In the regression problem, y takes a real value.

The following brief overview of SVMs as classifiers is adapted from [11].

3.1 SVM Classifier

The SVMs map the training input vectors x_i into a high dimensional feature space via nonlinear mapping and construct the generalized optimal hyperplane in that feature space, chosen a prior. The generalized optimal hyperplane is constructed by solving the optimization problem in Equation 5.

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}(w \cdot w) + C\left(\sum_{i=1}^{l} \xi_i\right) \qquad (5) \\ \text{subject to} \quad & y_i\left[(x_i \cdot w) + b\right] \geq 1 - \xi_i, \quad i = 1, \ldots, l \\ & \xi_i \geq 0 \end{aligned}$$

C is the penalty term for the nonseparable vectors.

This mapping does not need to be considered in explicit form since only the inner product in this feature space has to be calculated. Given the fact that the inner product in the feature space can be directly calculated as $(z_i \cdot z_j) = K(x_i, x_j)$, it is sufficient to solve the optimization problem of Equation 6, where the α_i are Lagrange multipliers.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (6) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, l \\ & \sum_{i=1}^{l} y_i \alpha_i = 0 \end{aligned}$$

Only support vectors can have nonzero coefficients α_i.The decision function is

$$f(x) = \operatorname{sign}\left(\sum_{sv} y_i \alpha_i K(x_i, x) - b\right) \qquad (7)$$

Mostly used learning machines include polynomial learning machines, two layer neural networks and radial basis functions machines. We select radial basis function defined in Equation 8 in our experiment because of the little knowledge we have on the properties of the design space.

$$K(x, x_i) = e^{-\gamma |x - x_i|^2} \qquad (8)$$

3.2 SVM Regressor

The support vector type approximation to regression finds the best approximation of the form

$$f(x; v, \beta) = \sum_{i=1}^{N} \beta_i K(x, v_i) + b \qquad (9)$$

where $K(\cdot,\cdot)$ is a kernel function, scalar β_i is the weight for each kernel function and v_i is a parameter of the kernel function, by minimizing the ε-insensitive loss function in Equation 10.

$$|y - f(x)|_{\varepsilon} = \begin{cases} \varepsilon, & \text{if } |y - f(x)|_{\varepsilon} \leq \varepsilon \\ |y - f(x)|, & \text{otherwise} \end{cases} \qquad (10)$$

When ε is set to 0, it is equivalent to minimize the maximum training error.

4. ACTIVE LEARNING SCHEME

Our experimental results show that the feasible designs are very sparse in the entire design space defined by the geometry constraints. If the designs are uniformly distributed, the feasible designs are only 2% of all the sampled designs for the OTA opamp shown in Section 5. The sparsity of the feasible design space poses a challenge in feasible design space modeling in the sense that even if the model has a good overall accuracy, the approximated feasible design space can deviate from the true feasible design space to an unacceptable extent. The error metrics defined in [5] are used here.

- Overall accuracy P_t

$$P_t = \frac{\text{Number of correctly classified samples}}{\text{Number of samples in the validation set}} \qquad (11)$$

- Percentage of false negative samples relative to all the positive samples.

$$P_{fn} = \frac{\text{Number of false negatives}}{\text{Number of positives in the validation set}} \qquad (12)$$

- Percentage of false positive samples relative to all the positive samples predicted by the feasibility classifier.

$$P_{fp} = \frac{\text{Number of false positives}}{\text{Number of predicted positives}} \qquad (13)$$

The active learning scheme [5] is able to improve these three accuracy metrics with much fewer samples compared to uniform random sampling. The idea of the active learning scheme is to find the feasible design space and its neighboring space and only sample this space. A *committee* of classifiers is used to accomplish this: we predict the classes of a large number of designs using each classifier in the committee and apply a voting mechanism such that instances voted as positive by at least one of the classifiers are sampled and instances unanimously voted as negative are not sampled. We add the new sampled instances to the training data set and repeat the same process until some stopping criteria is met. The number of classifiers in the committee is chosen such that training data set can be split into equal-sized sets.

5. EXPERIMENTAL RESULTS

We show two opamps as our illustrative examples. Our experiments are conducted on a Sun Blade 1000 machine and computational time is CPU time. The technology is AMI 0.5μm CMOS process and supply voltage is 5V. We use Hspice to simulate the circuit and extract performance parameters.

We will show the accuracy improvement of the feasibility classifiers constructed by the proposed active learning scheme compared to those constructed by a passive learning scheme. In the passive learning scheme, the training data set is generated by uniform random sampling. We construct the feasibility classifiers using the two learning schemes. The classifiers are both trained with 100% accuracy on the training set using the software called libsvm [2] on equal-sized training data sets. We then calculate the three accuracy metrics defined in Section 4 using a validation data set of 30,000 uniform random samples.

5.1 OTA Opamp

The OTA opamp is shown in Figure 1. We fix the lengths of all the transistors to 1.2μm and set the load capacitor to 1pF. It results in six free design variables by further considering transistor matching. The design variables and geometry constraints are shown in Table 1. Other constraints are shown in Table 2. The functional constraints ensure all the transistors are on and in saturation region with some margin. We also make sure that input offset voltage V_{os} is sufficiently small.

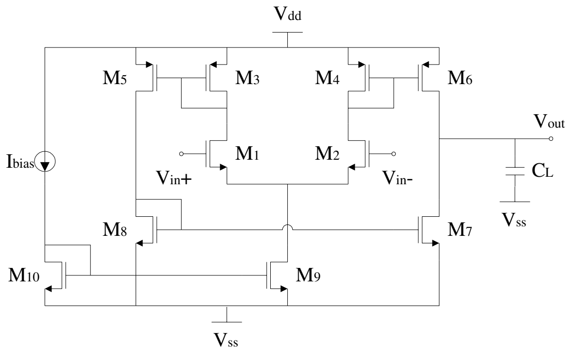

Figure 1: OTA opamp

Table 1: Design variables of OTA op-amp

Design variables	Geometry constraints
$W_1 = W_2$	[6μm, 200μm]
$W_3 = W_4$	[6μm, 200μm]
$W_5 = W_6$	[6μm, 200μm]
$W_7 = W_8$	[6μm, 200μm]
$W_9 = W_{10}$	[6μm, 100μm]
I_{bias}	[6μA, 100μA]

5.1.1 Feasibility Model

We set the number of classifiers in the committee to 16 in all iterations. Figure 2 shows the three accuracy metrics of feasibility classifiers constructed by two learning schemes. We constructed a series of classifiers using training data sets of various sizes to show that the active learning scheme is consistently better than the passive learning scheme. Each marker in these figures corresponds to one classifier constructed by all the sampled data when that iteration is completed and its accuracy metric. The results show that the active learning scheme improves the accuracy metrics at a much faster rate compared to the passive learning scheme.

The active learning scheme starts from 10,000 uniform random samples. With 8,000 more samples, the active learner is able to build a more accurate feasibility model than a passive learner that uses 50,000 random samples. The computational time for the active learning scheme to generate the feasibility model is 2862.9s while it takes the passive learning scheme 6445.6s to generate a less accurate feasibility model. The active learning scheme saves computational time by more than 56% compared to the passive learning scheme. It takes several milliseconds to evaluate the feasibility model.

5.1.2 Performance Macromodels

We further generate the performance macromodels of three performance parameters: open loop gain (Gain), unity gain frequency (UGF) and phase margin (PM). They can be obtained by running one AC analysis. During the feasibility model generation, the active learning scheme has to call Hspice and post processing the simulation data to get performance parameters and check if performance constraints are met. This data is reused here to generate and validate the performance macromodels. The active learning scheme has sampled 14,101 feasible designs, among which 7,000 are used as *training data* and the rest of 7,101 are used as *validation data*. The feasible designs are essentially randomly sampled by the active learning scheme.

Table 2: Design constraints of OTA opamp

Functional constraints	$V_{\text{gs}} - V_{\text{th}} \geq 0.1V$
	$V_{\text{ds}} \geq V_{\text{gs}} - V_{\text{th}} + 0.1V$
	$V_{\text{os}} \leq 0.01V$
Performance constraints	Phase Margin $\geq 45°$

Let $p\prime$ be the estimated performance parameter and p be the actual performance parameter, we calculate the error of models for Gain and PM as

$$e = p' - p \tag{14}$$

and the error of UGF as

$$e = \frac{p' - p}{p} \tag{15}$$

that has a unit percentage (%).

We also define MAX, the maximum of the absolute values of training errors or validation errors, in Equation 16.

$$\text{MAX} = \max\{|e|\} \tag{16}$$

MAX denotes the worst case performance of the generated performance model. Table 3 shows root mean squared error (RMS) and MAX (Equation 16) on the training data set and validation data set. Parameter ε is defined in Equation 10 and γ is the parameter of radial basis function defined in Equation 8. Cost C, defined in Equation 5, is set to 100 times γ.

The experimental results show that even MAX is considerably small. Hence, we are able to build very accurate performance macromodels only using simulation data for feasible designs. It takes a reasonable time to train the models and the models are very fast to evaluate.

Table 3: Statistics of performance macromodels of OTA opamp in feasible design space

	ε	γ	Training set		Validation set		T_{eval}	T_{train}
			RMS	MAX	RMS	MAX	(s)	(s)
Gain(db)	0.1	0.7	0.048	0.215	0.049	0.227	1.0e-4	27.9
UGF(%)	1.16	0.6	0.55	1.27	0.57	2.41	1.2e-4	48.2
PM(°)	0.1	3.0	0.062	0.357	0.080	0.917	6.6e-4	2929

We also generate performance models for Gain, UGF and PM using 7,000 samples and validate it using 7,101 samples but *don't* pose functional or performance constraints. As shown in Table 4, these models are less accurate, take much longer time to train and evaluate.

Table 4: Statistics of performance macromodels of OTA opamp in the entire design space (no functional or performance constraints

	ε	γ	Training set		Validation set		T_{eval}	T_{train}
			RMS	MAX	RMS	MAX	(s)	(s)
Gain(db)	0.1	1.2	0.053	0.777	0.058	0.618	2.6e-4	101.1
UGF(%)	1.16	0.6	0.87	5.1	0.94	6.8	9.8e-4	5688
PM(°)	0.1	3.0	0.37	2.41	0.53	2.89	3.1e-3	6hours

5.2 Two Stage Opamp

We show a two stage opamp as our second example. The circuit diagram is shown in Figure 3. Load capacitor C_L, is set to 1pF. We only apply basic matching knowledge, such as same width and length for transistors in the differential pair, same transistor length for transistors in a current mirror or current mirror bank. This results in 12 independent variables, as shown in Table 5. The design constraints for this two stage opamp are listed in Table 6.

5.2.1 Feasibility Model

In this example, we use two different strategies to choose v: one is to fix v at 20; the other is to decrease v in an annealing like manner, with v set to 20, 20, 10, 10, 10, 5, 5, 5, 4, 4 in each iteration. While we can not tell which strategy is more efficient, they both beat the passive learning scheme as shown in Figure 4.

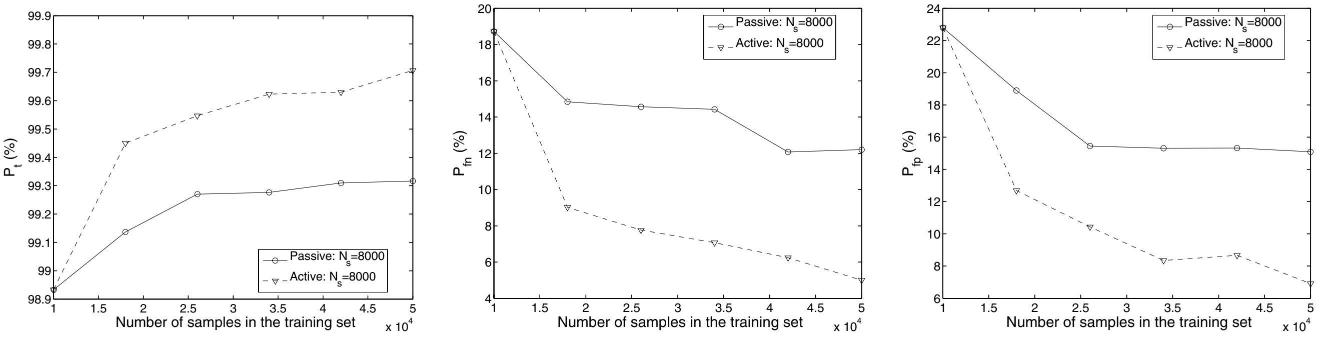

Figure 2: Accuracy metrics of OTA opamp feasibility models using two learning schemes

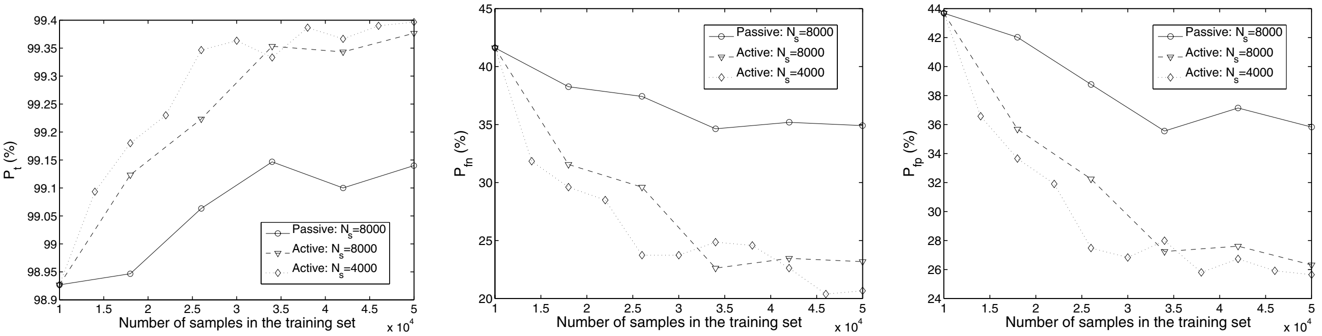

Figure 4: Accuracy metrics of two stage opamp feasibility models using two learning schemes

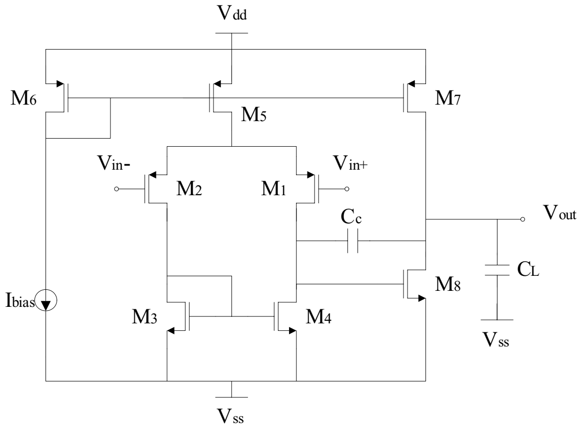

Figure 3: Two stage opamp

Table 5: Design variables of two stage opamp

Design variables	Geometry constraints
$L_1 = L_2$	[1.2μm, 20μm]
$L_3 = L_4$	[1.2μm, 20μm]
$L_5 = L_6 = L_7$	[1.2μm, 20μm]
L_8	[1.2μm, 20μm]
$W_1 = W_2$	[6μm, 200μm]
$W_3 = W_4$	[6μm, 200μm]
W_5	[6μm, 200μm]
W_6	[6μm, 200μm]
W_7	[6μm, 200μm]
I_{bias}	[1μA, 100μA]
C_c	[0.1pF, 10pF]

Table 6: Design constraints of two stage opamp

Functional constraints	$V_{\mathrm{gs}} - V_{\mathrm{th}} \geq 0.1V$ $V_{\mathrm{ds}} \geq V_{\mathrm{gs}} - V_{\mathrm{th}} + 0.1V$ $V_{\mathrm{os}} \leq 0.001V$
Performance constraints	Phase Margin $\geq 45°$

5.2.2 *Performance Macromodels*

We use the samples generated by the active learning scheme in which v is decreased in each iteration to train and validate three performance macromodels (this strategy generates more feasible designs in total). The training set has 10,000 samples and the validation set has 3,756 samples.

Table 7: Statistics of performance macromodels of two stage opamp in feasible design space

	ε	γ	Training set		Validation set		T_{eval}	T_{train}
			RMS	MAX	RMS	MAX	(s)	(s)
Gain(db)	0.50	0.52	0.269	1.64	0.290	1.66	8.6e-4	144.0
UGF(%)	4.71	0.10	2.12	10.2	2.18	10.1	4.0e-4	100.2
PM(°)	0.05	1.80	0.200	2.41	0.911	5.52	8.8e-3	7625

The experimental results show that even in this relatively difficult example, we are still able to build relatively accurate performance macromodels, shown by MAX in Table 7.

6. APPLICATIONS AND RESTRICTIONS

6.1 Design Space Reduction

The feasibility models can reduce the design space, such that performance macromodels in the feasible design space can be highly accurate. We show the design space reduction by posing design constraints in Table 8. Let the design space determined by the geometry constraints C_g be 1, the third column shows the size of the

Table 8: Reduced design space with design constraints

Circuit	C_g	$C_g \cup C_f$	C
OTA opamp	1	12.9%	2.32%
Two stage opamp	1	11.1%	1.24%

reduced design space relative to the original design space after posing the geometry and functional constraints and the fourth column shows the size of reduced design space relative to the original design space after posing all the constraints.

6.2 Analog Synthesis

The combined model can be used to replace the performance parameter acquisition through simulation to speed up analog circuit sizing. The sizing tool will first query if a design is feasible. The performance parameter estimation is only needed by the sizer when the design is predicted feasible. The cost function can be designed with a large penalty term to indicate infeasibility.

6.3 Topology Selection And Design Space Exploration

Since analog sizing can be sped up by a factor of at least one hundred due to the short evaluation time of the performance macromodels compared to transistor level simulation, topology selection is much easier. We can simply size various topologies and pick the most suitable one.

To approximate the design space boundary, numerous designs have to be sized [4]. With the aid of analog performance macromodels the design space boundary of a circuit can be found much more efficiently.

7. LIMITATIONS

The feasibility model could mislead a sizing tool in two possible ways: false convergence caused by the predicted feasible designs being actually infeasible and no convergence caused by the predicted infeasible designs being feasible. The two possible situations have to be studied by using the models in actual sizing.

Another drawback is that the computational time is relative long especially when the input design space is in higher dimension. the improvement of the feasibility model accuracy needs to be further accelerated by using few samples.

This active learning algorithm also needs a set of random samples to start with and requires at least some feasible designs in this set. This might be problematic as the dimension of the design space increases. However, it works well for a medium range of dimension, between six and twelve in our experiments.

8. CONCLUSION

We present a combined feasibility and performance macromodel for analog circuits. The generated model is composed of one feasibility model and a set of performance macromodels. The feasibility model identifies the feasible designs in the entire design space while the performance macromodels approximate the performance parameters. We treated he feasible design space identification problem as a two class classification problem so that advanced classification algorithm SVMs can be applied. We also applied an active learning scheme to tackle the sparsity of the feasible designs. Thus we are able to build accurate performance macromodels. Experimental results show that it is an efficient approach to analog circuit performance macromodeling. The applications of analog performance macromodels include fast analog sizing, design space exploration and topology selection. The limitations of the proposed methodology is also discussed.

9. REFERENCES

[1] O. Bajdechi, J. H. Huijsing, and G. Gielen. Optimal design of delta-sigma ADCs by design space exploration. In *Proceedings of the 39th conference on Design automation*, pages 443–448. ACM Press, 2002.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] W. Daems, G. Gielen, and W. Sansen. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(5):517–534, 2003.

[4] 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*, 22(2):213–224, 2003.

[5] M. Ding and R. Vemuri. An active learning scheme using support vector machines for analog circuit feasibility classification. In *Proceedings of the 18th international conference on VLSI design*, 2005.

[6] A. Doboli, A. Nunez-Aldana, N. Dhanwada, S. Ganesan, and R. Vemuri. Behavioral synthesis of analog systems using two-layered design space exploration. In *Proceedings of the 36th ACM/IEEE conference on Design automation*, pages 951–957. ACM Press, 1999.

[7] 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*, 20(1):1–21, 2001.

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

[9] A. Nunez-Aldana and R. Vemuri. An analog performance estimator for improving the effectiveness of CMOS analog systems circuit synthesis. In *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pages 406–411, 2001.

[10] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich. Initial sizing of analog integrated circuits by centering within topology-given implicit specifications. In *International Conference on Computer Aided Design*, pages 241 – 246, 2003.

[11] V. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.

[12] M. Vogels and G. Gielen. Architectural selection of A/D converters. In *Proceedings of the 40th conference on Design automation*, pages 974–977. ACM Press, 2003.

[13] 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*, 22(2):198–212, 2003.

[14] S. Zizala, J. Eckmueller, and H. Grab. Fast calculation of analog circuits' feasibility regions by low level functional measures. In *IEEE International Conference on Electronics, Circuits and Systems*, pages 85–88, 1998.