Behavioral-level performance modeling of analog and mixed-signal systems using support vector machines

X. Ren and T. J. Kazmierski
School of Electronics and Computer Science
University of Southampton, UK
{xr03r,tjk}@ecs.soton.ac.uk

ABSTRACT

This paper presents a novel behavioral-level analog and mixed-signal (AMS) system performance modeling methodology using support vector machines (SVM). The method relies on linearly graded sub-spaces to model complex multi-dimensional performance spaces. A detailed evaluation of the method has been carried out for the purpose of potential use for AMS synthesis. The method has been applied to a complex non-ideal 2^{nd} order Sigma-Delta modulator (SDM) and results show good accuracy of performance modeling and numerical efficiency.

1. INTRODUCTION

One of the key elements of hardware synthesis is to employ a modeling method that evaluates the system's performance. At system design level, performance modeling outcomes are extremely important because the earlier in the design process a hardware decision is committed to fabrication, the more costly it is to change it later. Current AMS IC synthesis methodologies lag behind their digital counterparts. One of the main problems is that different system-level solutions cannot be distinguished effectively because of the absence of efficient performance estimation methods. Moreover, the productivity of existing AMS synthesis tools is limited by the nature of analog designs, which feature multi-dimensional design space and various blocks with different sets of benchmarks. Furthermore, performance parameters of analog systems usually have a cross-domain trait and frequency domain parameters may need to be extracted using complex mathematical methods. Because of the emergence and popularity of modern system-on-chip (SoC), the need to develop new AMS CAD tools is even more important as analog parts, which usually represent only a small portion of a typical SoC, involve disproportionally large design time and require specialized skills [6]. To overcome these challenges, many landmark engineering methods have been applied to create systems at either system-level or circuit-level or both. In recent years, fuzzy logic [10] and neural network methods [12] have been applied to supersede manual knowledge extraction and thus increase the design process efficiency. These systems construct learning rules or artificial intelligent networks to represent and reproduce the behavior of target systems. Another extensively studied methodology is symbolic analysis approach and outstanding systems have been accomplished[3, 4]. Typical applications are mainly circuit-level analog design cases. For an AMS system which includes both analog and digital blocks, the highly non-linear and implicit relationship between design and performance parameters may restrict the effectiveness of the method. The third approach is to use optimization techniques directly on simulation results rather than focus on system modeling [5, 9]. However, the efficiency of the performance evaluation process is severely affected by the simulation cost and the method sacrifices reusability in exchange for generality and flexibility.

SVM-based methods have recently been introduced to the analog performance modeling field [1, 8]. As a newly suggested approach, these powerful learning algorithms need to be studied further in the context of AMS systems. Currently SVM-based performance classification is restricted to the classical "good-bad" problem [1]. In this paper, a novel linearly graded performance modeling process is presented that uses SVMs for behavioral-level AMS systems. The new method can produce a more detailed assessment of the system's performance with an accurate and computationally efficient modeling technique. Section 3 shows how it has been successfully applied to a complex AMS case study.

2. BEHAVIORAL-LEVEL LINEARLY GRADED AMS PERFORMANCE MODELS

SVM uses structured risk minimization (SRM) strategy to train models instead of the empirical risk minimization (ERM) strategy used by the traditional neural network method. It is proved that SRM is superior than ERM strategy [7] with better generalization ability. SVM uses dot products in the feature space to measure similarity between samples and translate the input-space classification problem to optimum hyperplane construction in the feature space. The method is simplified by using inexplicit kernel mapping, which represents the feature space hyperplane using linear combination of weighted kernel functions and biases that centralized only on a subset of the entire data set called support vectors (SVs). LibSVM [11] is used as the SVM simulator and there are four commonly used kernel function included:

1: Linear Kernel: $k(x,x') = \vec{x} \cdot \vec{x}'$
2: Polynomial Kernel: $k(x,x') = (\gamma\vec{x} \cdot \vec{x}' + r)^d$
3: Radial Basis Function Kernel (RBF):
$k(x,x') = exp(-\gamma||\vec{x} - \vec{x}'||^2), \gamma > 0$
4: Sigmoid Kernel: $k(x,x') = tanh(\gamma\vec{x} \cdot \vec{x}' + r)$

where k represents the kernel function, γ, r, d are parameters of the kernel functions that control the characteristics of the kernels and $x, \vec{x}$ are samples in the input and feature space separately.

Support vector classification (SVC) problem is to construct the hyperplane that provides maximum separating margin between different classes, which is equal to the following quadratic optimization problem:

$$\Phi(w, \xi) = \frac{1}{2}||\omega||^2 + C \sum_{i=1}^{m} \xi_i \qquad (1)$$

$$subject\ to\ y_i[< \omega, x_i > +b] \geq 1 - \xi_i \ \ i = 1, \cdots, m \qquad (2)$$

where ω is the normal vector of the hyperplane, x are the samples in the input space and y is the corresponding label vector, b is the bias vector of the hyperplane, ξ is the slack vector corresponding to the errors happens on misclassified samples and C is the penalty parameter that determines the maximum margin and the minimum error cost or equally saying the trade-off control parameter between the complex of the hyperplane and the accuracy of the classification. The SRM principle is used for support vector regression (SVR) to solve regression problems. A loss function [7] is added to enable the distance measurement between the hyperplane and the sample data. The function can be directly used in the optimization problem to replace the targeted labels in SVC. There are two phases to generate classification and regression models: training and testing, which correspond to the creation and verification of the models separately.

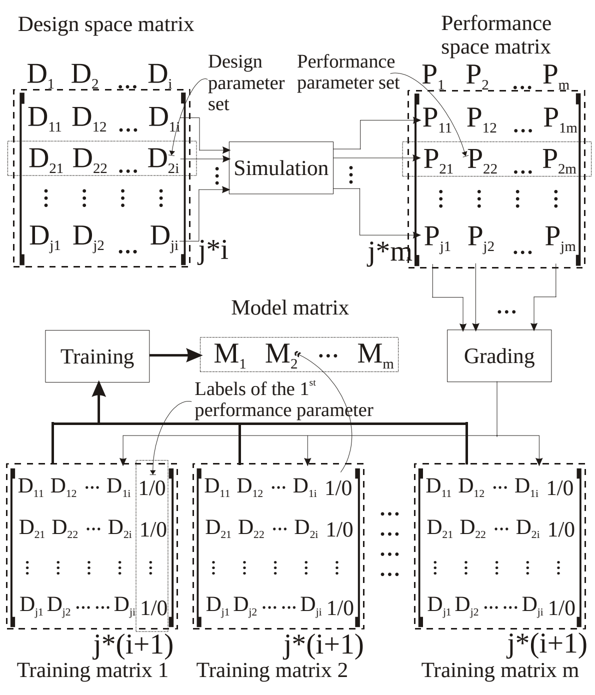


Figure 1: The design structure and information flow of the method from the design space until the generation of the models where $1/0$ **means 1 or 0.**

The proposed approach is to build knowledge database models on the direct relationship, which can avoid adjustment of any of the internal variables, between the design and performance parameters illustrated by the following equation:

$$P(p_1, \cdots, p_m) = f(S, d_1, \cdots, d_n) \qquad (3)$$

where P is the performance space and p_i is the i^{th} performance parameter; d_j is the j^{th} design parameter; f is the direct relationship function and S represents the function of the AMS system. The models are expected to provide accurate modeling and good insight of the behavior of the system.

2.1 Linearly graded performance modeling using SVC and SVR

The scale of equation (3) is usually massive because of the nature of the relationship between the design and performance spaces. To reduce the complexity of the problem, a linearly graded SVM-based approach is proposed. The idea is to divide the performance space into sub-spaces of various performance levels. This not only makes the solution process more efficient but also provides a better performance estimation than the traditional 'good-bad' approach. The boundaries between the sub-spaces are constructed using SVC models and details inside sub-spaces are represented by SVR models based on the corresponding design space data. In this combination, SVC provides a rough framework for target systems and SVR provide detailed modeling within sub-spaces.

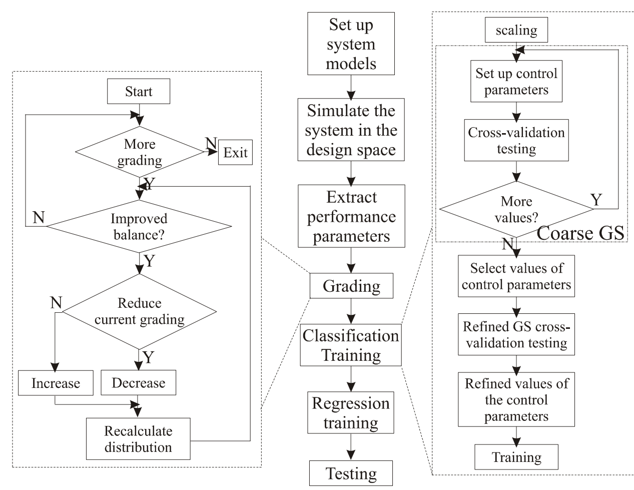


Figure 2: The modeling algorithm including the BDG and the training algorithm details.

Figure 1 shows the data organization and the design flow graph (DFG) of the training process. The first phase is to run simulations to generate performance data. Each design set and its corresponding performance set are combined to form one data set. The second phase is grading, which operates on the performance figures. As each performance parameter corresponds to one dimension in the performance space, when the grading is carried out on each performance parameter, the resulting grading planes separate the entire space into sub-spaces contained by hypersurfaces. Each grading plane is constructed using SVC with samples labeled as '0' or '1'[1] indicating whether or not the sample belongs to the given sub-space. The third phase is to take the design sets and their labels to train the SVM performance models.

[1] For multi-class classification, the label value can be other integrators. For regression process, the labels are replaced by real-valued numbers.

An implementation of the modeling process is shown as the DFG in figure 2 including a balanced data grading (BDG) algorithm on the left and a practical parameter determination algorithm the right in dashed boxes. The grading vectors of the performance parameters are originally specified by designers without considering the distribution of samples in the performance space. The BDG algorithm is a heuristic process that takes the designer provided grading vectors as the starting point and recalculates elements in the grading vectors to obtain approximate uniform grading of the samples for every performance parameter. The idea is to avoid sparse representation of some of the sub-spaces, which can later cause problems such as over-fitting. The algorithm treats all the sub-spaces equally and applies a similar amount of material to the representation of every sub-space. Also, the algorithm contributes the decomposition of the problem because it avoids a congregation of the samples. A disadvantage of the algorithm is that it is not optimal in the sense of finding the most suitable boundary. It may grade samples with better similarity into different groups with compelling separation. However, results show that the generalization ability of the SVM compensates for this disadvantage.

The SVM control parameter determination algorithm in figure 2 is general and effective for all types of kernels. Two techniques are involved to accomplish the parameter determination task: the first one is the grid search (GS) method to scan the control parameters; the second one is the cross-validation technique, which is common in traditional artificial intelligence technology like neural network. Both the GS and cross-validation aim to obtain the optimal values for the control parameters that are used to construct optimal models. Because the SVM cost parameter C and some of the kernel parameters change exponentially, it is not practical to do a GS with high grid resolution on the entire scan range. The problem is overcome by scanning the control parameter space in two successive sub-phases. In the coarse grid search (CGS) sub-phase, the parameters are scanned with exponentially increased grid resolution; while in the refined grid search (RGS) sub-phase, the parameters are scanned in much better resolution only within the optimum region found in the CGS sub-phase. This enhances the efficiency of the modeling process. The cross-validation technique is employed to improve the accuracy performance of the SVM.

3. CASE STUDY

3.1 Design and performance spaces

A 2^{nd} order SDM, which has been developed in SystemC, is used as a case study to demonstrate the effectiveness of the performance modeling approach outlined above. With its inherent and integrated technology dependent non-ideal effects, this example is a typical representative of moderately complex AMS systems found in practice for which there is an urgent need to develop automated synthesis methods. Non-ideal effects include: the clock jitter, switch thermal noise, OpAmp thermal noise, OpAmp slew rate and finite bandwidth, OpAmp finite and non-linear DC gain as well as quantizer hysteresis and offset. These non-idealities make the SDM drift away from the ideal design with a degradation of up to 10 dB in the SNR figure. Therefore, the design's ideal model can not provide practical predictions of the behavior. Such predictions are heavily dependent on specialized expert knowledge. Behavioral-level analytical models developed with the proposed approach attempt to find the relationship between the design and performance parameters including all the imperfections in the non-ideal models.

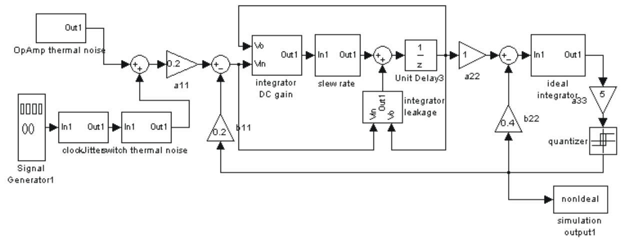


Figure 3: The design example: a 2^{nd} order SDM with integrated non-ideal effects.

As shown in figure 3, there are five amplifiers controlling the feed forward and feedback coefficients: a_1, b_1, a_2, b_2 and b_3. The first four are selected to construct the design space making the design space to be four dimensional. The effect of b_3 is absorbed by the quantizer immediately and is not propagated further so it is not selected. The two most important performance parameters of SDM systems are the signal-to-noise ratio (SNR) and dynamic range (DR). Both are calculated using Fast-Fourier Transform on the simulation results. To make it meaningful, the peak SNR is selected as a performance parameter instead of a complete SNR curve. DR is defined as a ratio of the output power at the frequency of the input sinusoid with a full-scale input over the output power of a small input for which the SNR is 0dB. The third is the stability, which is a binary classification parameter, i.e. the SDM is either stable or unstable. The SDM is said to be unstable when internal signals are detected that exceed a predefined threshold. The fourth performance parameter is the dynamic signal range. This requirement represents a severe problem in circuit technologies such as CMOS VLSI, where the dynamic range of the technology itself is limited [2]. In the SDM, this parameter is controlled by the outputs of the two integrators (represented as INT1 and INT2 symbolically). In total, there are five performance parameters. To sum up, the performance model of the SDM system is a direct relationship between the 4 dimensional design space and the 5 dimensional performance space.

3.2 SVM performance analysis

The two criteria used in the SVM performance analysis are the accuracy (effectiveness) and the computational cost (efficiency). Totally, 4725 design sets have been scanned in the design space. Hence, correspondingly, there are 4725 performance sets distributed in the performance space used for the training.

3.2.1 Kernel comparison

Using the initial grading vectors, the performance of SVM with different kernels is compared and surveyed in table 1 on major performance parameters. From the accuracy point of view, classification accuracy (A) is the measurement of the SVC performance and the mean squared error (MSE) is calculated for SVR. The MSE is defined as the standard deviation between the predicted and simulated values of the

Table 1: Performance of the SVC and SVR with linear, RBF and sigmoid kernels.

SVM method	Compared parameter	Linear	RBF	Sigmoid
SVC	$A_{stability}$	82.98%	99.58%	99.05%
	$T_{stability}$	16:48:09	00:34:17	08:22:00
	A_{SNR}	82.01%	98.77%	98.22%
	T_{SNR}	22:30:40	00:53:54	09:16:12
SVR	MSE_{SNR}	786.39	57.6721	686.048
	T_{SNR}	02:48:31	31:49:49	17:13:37

performance parameters. The SVC section in table 1 shows that the accuracies of the SVM with the RBF kernel and the sigmoid kernel are comparable while that of the linear kernel is worse. The CPU time consumption for the classification problems shows that the RBF kernel is much more efficient than the other two.

Also the MSE performance for regression problems in the SVR section in table 1 shows that the RBF kernel is the best option, even though the regression time in this case is the longest. However, the time consumption can be reduced by improved data grading[2]. The polynomial kernel is not listed in table 1 because the computational cost reaches a level of almost 5 days for parameter selection for SNR's regression modeling. Considering the results above, the RBF kernel was selected for further experiments.

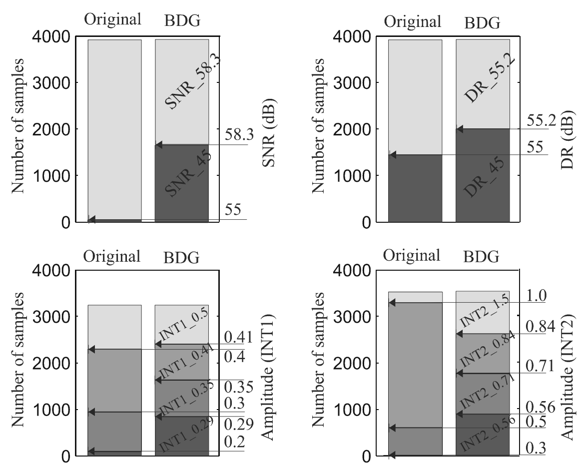


Figure 4: The distribution of the performance sets on every performance parameter dimension in dual-y axis plots. The left y axes are the numbers of samples and the right y axes are performance parameter values.

3.2.2 Generation of grading vectors using the BDG algorithm

Figure 4 shows the distribution of the performance sets with different grading vectors for each performance parameter. The left bars in the subplots are the performance sets distribution with the original grading vectors. They reveal the disadvantage of the original grading vectors because sparse distribution can generate over-fitted models and over-congregated training data increase the difficulty of regression. The right bars in the subplots are the ones calculated with the BDG algorithm. A new set of grading vectors is generated and non-equilibria are avoided. Each segment in the BDG bars in figure 4 corresponds to one class of the performance parameter and is labeled by the inclined text.

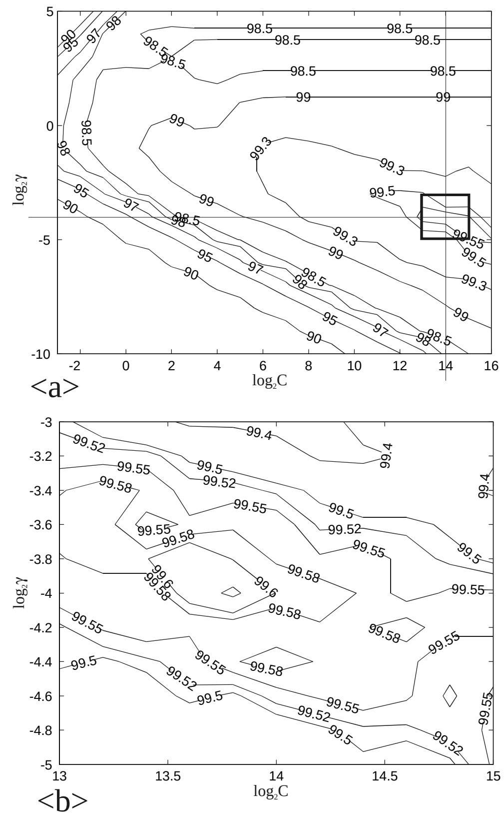


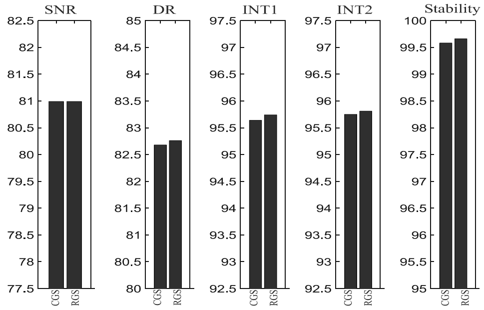
Figure 5: The classification accuracy contour of the stability. a) accuracy contour in the CGS phase; b) accuracy contour in the RGS phase.

3.2.3 SVM training performance

Figure 5 a) shows the CGS accuracy contour of the stability with the scanning ranges of $C \in [2^{-3}, 2^{15}]$ and $\gamma \in [2^{-10}, 2^{5}]$. The expected optimal region is boxed by a thick frame and centered at $C = 2^{14}, \gamma = 2^{-3}$. This feature is observed in all the CGS phase of SVC experiments of all the performance parameters. After finding the coarse optimal region, the next phase is to apply the RGS phase within the region. The accuracy contours shown in figure 5 b) show the result of RGS scanning for stability. As in the figure, the classification accuracy contour of the stability is in the region of $C \in [2^{13}, 2^{15}]$ and $\gamma \in [2^{-5}, 2^{-3}]$, where a set of C and γ is found that can give improved classification accuracy performance. The summary of the CGS and RGS SVC results are compared in figure 6.

Figure 8 a) and b) shows the CGS and RGS MSE contour of INT1_0.35, which is a sub range of INT1. As the clas-

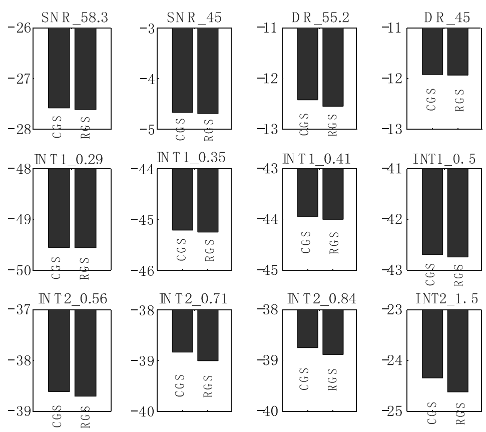
[2] The regression data includes all the samples even the unstable cases, however the unstable samples are dismissed for all the performance parameters in later experiments except for the stability.



	CGS	RGS
Total time	47:50:46	02:18:04

Figure 6: The classification accuracies of all the performance parameters with the corresponding run time of the CGS and RGS phases.

sification accuracy contours for SVC model generation, the diagram shows the same feature that there is an optimal region confined in the coarse scanning, where it is the RGS phase that searches the region in detail. Results of the CGS and RGS phase in the process of SVR models' generation are summarized and compared in figure 7. The MSE (dB) performance of the SVR models in this experiment is at a level of 0.1% compared with the performance parameters being modeled.



	CGS	RGS
Total time	49:24:57	66:52:47

Figure 7: The MSE performance for all the graded ranges of all the performance parameters with the corresponding run time for the CGS and RGS phase.

SVM parameters found in the RGS phase improve the performance of the models both for the classification and the regression problems. These parameters have been used to train the models. The training process of these SVM models of this linearly graded approach is compared with the full-space analysis approach. The top plots in figure 9 show the MSE performance comparison. The full-space bars show that in most cases, the MSE performance of the linearly graded approach is better than that of the full-space analysis approach. The bottom table presents the summary of the computational cost. It shows that grading models lead to more than 50% saving in the computational cost. In fact,

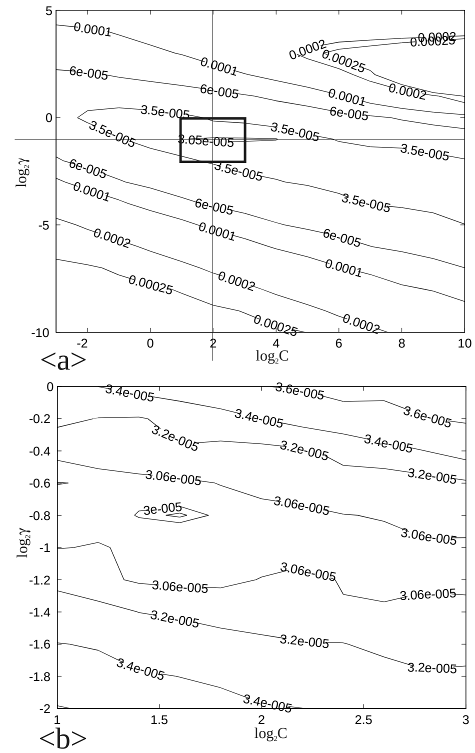
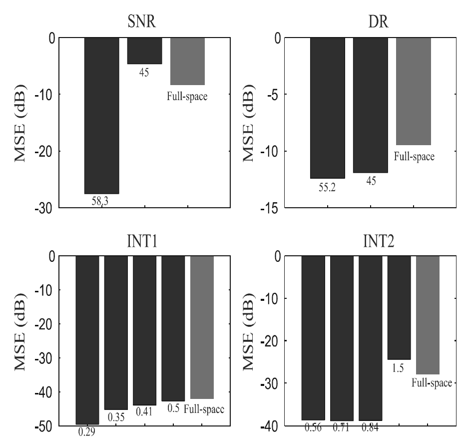


Figure 8: The MSE contour of grading boundary INT1_0.35 in the SVR. a) is the CGS MSE contour and b) is the RGS MSE contour.



Approach	Computational cost
Linearly graded	50:08:50 (classification) 49:27:57 (regression CGS)
Full-space analysis	190:44:11 (CGS only)

Figure 9: A comparison of the regression model MSE performance between the linearly graded approach and the full-space analysis approach. The corresponding grading values are shown at the bottom of each bar.

the full-space analysis approach is not practical for applications of this kind because of the huge time consumption.

3.2.4 Testing of the models

The sampling resolution in the design space used for testing is 5 times denser than that for the training data. This assures that firstly all the testing data are unknown to the models; secondly, the testing data are rigorous and the trained models must have good generality to preserve their accuracy. Using the high sampling resolution, two testing data sets are generated. Testing data set 1 (Testing1), with 1323 samples, spreads over large regions in the design space covering both the stable and unstable designs cases to test the overall quality of the models. Testing data set 2 (Testing2), with 4725 samples, is centralized on a small stable region with mainly preferable design cases but different trade-off between the performance parameters. It is used to verify the models with very congregated data. The testing procedure is to use the classification models to predict the classification of the testing data first then according to the predictions, corresponding regression models are used to calculate numerical results. The performance of the models in the testing phase is summarized in table 2. The overall prediction performance confirms the training performance. Blank cells in the table mean that the corresponding operations were not carried out either because the performance space covered by the testing data set does not include that region or because of misclassifications. Misclassifications happen during the classification testing. However misclassified samples are always very close to the boundaries. The MSE performance of the regression testing shows that very accurate numerical predictions are calculated, so misclassifications do not degrade the performance of the method significantly. All in all, the testing results show great robustness and reliability of the approach.

Table 2: Testing results for the classification and regression models.

	Parameter	Testing1	Testing2
Classification Accuracy	SNR	66.72%	78.73%
	DR	74.5%	62.7%
	INT1	87.7%	97.1%
	INT2	78%	99.6%
	stability	98.4%	100%
Regression MSE(dB)	SNR_58.3	-5	-26
	SNR_45	-	-
	DR_55.2	-11	-6
	DR_45	-	-5
	INT1_0.29	-	-29
	INT1_0.35	-43	-47
	INT1_0.41	-37	-50
	INT1_0.5	-38	-
	INT2_0.56	-28	-20
	INT2_0.71	-28	-29
	INT2_0.84	-26	-
	INT2_1.5	-14	-

4. CONCLUSIONS

This paper presents a new concept of linearly graded knowledge database model for performance modeling of AMS systems. A corresponding modeling process using SVMs has been developed. The efficiency of the method was demonstrated using a complex mixed-signal case study of a non-ideal 2^{nd} order SDM. As the proposed approach provides accurate performance evaluation over the entire performance space, the method can potentially lead to improved AMS synthesis techniques. Testing results show the robustness of the method in the presence of challenging relationships between the design and performance spaces. The accuracy and computational cost of SVMs with different kernels have been compared. It was found that the RBF kernel is superior in terms of the trade-off between the accuracy and the computational cost over other kernels.

5. REFERENCES

[1] F. D. Bernardinis, M. I. Jordan, and A. sangiovanni Vincentelli. *Support vector machines for analog circuit performance representation*, pages 964 – 969. Design Automation Conference, Jun 2-7 2003.

[2] B. Boser and B. Wooley. *The design of sigma-delta modulation analog-to-digital converters*, volume 23, issue 6, pages 1298 – 1308. Solid-State Circuits, IEEE Journal of, Dec 1988.

[3] G. Casinovi and J.-M. Yang. Load-independent behavioural model for cmos operational amplifiers. *Circuits, Devices and Systems, IEE Proceedings [see also IEE Proceedings G- Circuits, Devices and Systems]*, 142, Issue 6:399 – 405, Dec 1995.

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

[5] A. Doboli and R. Vemuri. Behavioral modeling for high-level synthesis of analog and mixed-signal systems from vhdl-ams. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22, Issue 11:1504 – 1520, Nov 2003.

[6] G. G. E. Gielen and R. A. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, 88(12):1825–54, 12/ 2000. M1: Copyright 2001, IEE.

[7] S. Gunn. Support vector machines for classification and regression. Technical report, University of Southampton, May 1998.

[8] T. Kiely and G. Gielen. *Performance modeling of analog integrated circuits using least-squares support vector machines*, volume 1, pages 448 – 453 Vol.1. Design, Automation and Test in Europe Conference and Exhibition, Proceedings, Feb 16-20 2004.

[9] E. Ochotta, R. Rutenbar, and L. Carley. Synthesis of high-performance analog circuits in astrx/oblx. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15, Issue 3:273 – 294, Mar 1996.

[10] A. Torralba, J. Chavez, and L. Franquelo. Fasy: a fuzzy-logic based tool for analog synthesis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15 Issue 7:705 – 715, July 1996.

[11] C. wei. Hsu, C. chung. Chang, and C. jen Lin. *A practical guide to support vector classification*. National Taiwan University, http://www.csie.ntu.edu.tw/cjlin/libsvm, 2001.

[12] G. Wolfe and R. Vemuri. Extraction and use of neural network models in automated synthesis of operational amplifiers. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22, Issue 2:198 – 212, Feb 2003.