A Method for Semi-Automated Modeling of Analog-Mixed Signal Systems in Automotive Applications based on Transient Simulation Data

H. Mielenz*, R. Dölling* and W. Rosenstiel**

* Robert Bosch Group, AE/EIM - Advanced Methods in IC-Design, Reutlingen, Germany
{holger.mielenz, rolando.doelling}@de.bosch.com
** University of Tübingen, Tübingen, Germany, rosenstiel@uni-tuebingen.de

ABSTRACT

Functional verification by simulation is often based on behavioral models and represents an important step during the development of microelectronic solutions for automotive applications. The present article describes a data-based approach for semi-automated generation of behavioral models for analog mixed-signal (A/MS) systems. The approach is based on support vector machines and a transformation dictionary for extraction of dynamic properties. The application of this method results in highly accurate pin-compatible behavioral models for A/MS systems with a significant reduction in simulation times. Additionally, the generated models can be easily integrated in hardware or system-level description languages. Another benefit of the proposed method consists in its flexibility to model systems of different physical domains. The emphasized properties will be illustrated by the modeling of an analog-digital system belonging to the automotive domain. [1]

Keywords: data-based modeling, analog/mixed-signal systems, support vector machines, mixed-level simulation

1 INTRODUCTION

The microelectronic products for automotive applications are characterized by an ever-increasing level of integration complexity. The stringent requirements on the communication between continuous-valued analog systems and their surrounding lead to a growing need for new methods that improve the design productivity and quality [1]. This paper presents a data-based approach for the generation of behavioral models for time-consuming components of analog mixed-signal (A/MS) systems. These models can substitute the time-critical parts in ASIC-designs and contribute to an important improvement of the functional verification process based on a significant acceleration of A/MS-system's simulation within analog and analog-digital simulators.

The proposed approach is based on a two-stage process referring to a signal preprocessing step and a subsequent modeling part. The signal preprocessing is realized by a transformation dictionary of predefined functions in order to extract the dynamic characteristics for the modeling task. The modeling part is accomplished by a decision tree framework for the piecewise local estimation of the global system, whereas support vector machines (SVM) [2] are used as local function approximation method.

The majority of work on the application of support vector machines (SVM) to electronic design automation considers the modeling of design and performance parameters of analog circuits [3] [4] [5] [6]. The use of SVMs for behavioral modeling of analog and analog-mixed signal circuits is a topic of current research [7].

This paper is organized as follows. Section 2 introduces a decision tree-based framework as a modeling strategy for A/MS systems. Section 3 illustrates the properties of the introduced algorithm concerning the generation of behavioral models for simulation acceleration. Furthermore, the flexibility of the generated models is shown by their embedding into different description language environments. Section 4 is devoted to conclusions and suggestions for future work.

2 THEORETICAL BACKGROUND

In general, A/MS systems pertain to dynamic, time-varying processes and might represent strongly non-linear characteristics. This section briefly presents a regression estimation approach for non-linear data modeling based on decision trees supported by support vector machines as non-linear approximation method.

2.1 A Probabilistic Description

Consider a given finite data set $D_N = \{x(t), y(t)\}_{t=1}^{N}$ of N observed input-output pairs generated by a non-linear system, e.g. an analog-digital circuit. The behavioral modeling of A/MS systems is based on the estimation of a functional mapping f between the predefined data D_N referred to as training data. Thus, the problem of regression estimation is the deduction of an *a priori* unknown function $f(\dots)$

$$f : y(t) = f(x(t), \Theta) \tag{1}$$

[1] This work was supported by the German Federal Ministery of Education and Research (BMBF) under the registration 01 M 3079 B (project 'VeronA') in the scope of the BMBF "Nanoelectronics" sponsorship program.

where Θ is an unknown parameter vector associated with a not yet determined model structure that minimizes a loss function L like

$$L(y_t, f(x_t, \Theta)) = (y_t - f(x_t, \Theta))^2. \quad (2)$$

2.2 The Modeling Architectures

The model structure $f(., \theta)$ for regression and function estimation of A/MS systems used in this work is derived from the decision tree theory. In general, support vector machines (SVM) (see chapter 2.3) are applied on predefined training data D_N in order to find an approximation for f that minimizes L, known as a global regression approach (see figure 1a). Our investigations have pointed to a lower applicability of SVMs to the direct modeling of A/MS systems due to lower representation capabilities of different dynamic properties. The procedure has been improved dramatically by the augmentation or replacement of the raw input vectors with additional variables, corresponding to transformations of the input variables containing dynamic information (see figure 1b and section 2.4). The effectiveness of this approach is further improved by tree-based methods which generally partition the input-space into compact sets of corresponding data pairs. Since the output of each set is fitted by an SVM regression model (see figure 1c), decision trees are among the procedures for partial modeling of a global function. The approximation by partial modeling is motivated by a faster and more efficient training process due to the partition of the system's operation area. The different subjects are discussed in the following sections.

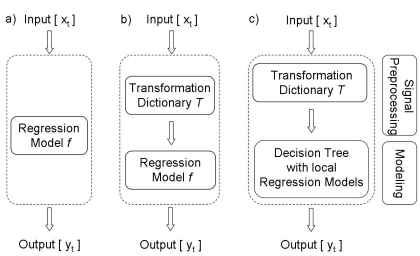

Figure 1: Illustration of different regression modeling techniques. a) regular global regression b) global regression with transformation of input signals c) local regression with transformation of input signals

2.3 Support Vector Machines

The regression approach considered in this work is based on a quadratic programming formulation for regression and function estimation known as support vector regression (SVR) [2]. The SVR approach yields a function estimation of the form

$$f(x_t) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_t, x_i) + b \quad (3)$$

where K is known as the kernel function representing any nonlinear similarity measure obeying Mercer's condition [8]. The algorithm calculates solutions for α_i and α_i^* which are the Lagrange coefficients for the integration of a subset of training vectors x_i into the solution of the quadratic programming object function. Typical kernels used for SVR within the given context are Gaussian or monomial functions [2]. Therefore, the unkown parameter vector Θ for an SVR structure is given by the Lagrange coefficients $\alpha_i^{(*)}$, relevant training vectors x_i and the bias b in addition to the user-predefined kernel function parameters [8].

2.4 The Meaning of the Transformation Dictionary

Figure 1b illustrates a well-known strategy for moving beyond nonlinearity with linear models. This approch can be seen as a dictionary T consisting of a large number of transformation functions M. The purpose of these transformations $t_m(x_t) \in T,\ m \in [1, 2, \ldots M]$ is to expand the scope of the adjacent regression method f. Therefore, the output of such a model is calculated by

$$y(x_t) = f(t_1(x_t), t_2(x_t), \ldots, t_M(x_t)) = f(T(x_t)) \quad (4)$$

where the codomain's dimension is increased according the chosen transformations.

Beside the original usage, dictionary methods can also be utilized for the representation of dynamic properties within the framework of system identification. The application of a dictionary to the modeling of A/MS systems has to consider the different properties of digital and analog input signals. As digital signals are value- and time-discrete, only a few transformations can be applied with valuable outcome. The most important are information about the Shannon entropy and the frequency of the applied clock. The problem-dependent Shannon entropy can be calculated by various integral functions t_S

$$d(x_t) = \int_{\delta t} t_S(x_t) dt \quad (5)$$

for a predefined window size of time δt. The window size represents the memory of the system about the prior clock information.

Since analog signals are value-continuous, more complex functions can be used for the extraction of important information. Beside different filter approaches, like Laguerre and Kautz filters, simple arithmetic operations are applied for the estimation of frequency information. Another class of important dictionary components are signal routing elements like tapped-delay functions with different time constants, also known as autoregressive (AR) or autoregressive moving average (ARMA) functions [9].

All parameters of the digital and analog transformation functions are calcuated on the training data automatically.

2.5 Partial Modeling with Decision Trees

Tree-based methods partition the input space into a set of compact subspaces, allowing the partial modeling of an unknown functionality by the fit of a general regression model to each leaf, as shown in figures 1c and 2. Decision trees are conceptually simple but a powerful solution concerning the obstacles of this work. This is emphasized by a fast determination of the most probable local regressor as the complexity is linear in the depth l of the tree, therefore $O(l)$. Additionally, the partitioning allows the fit of small regressors with less arithmetic operations and faster simulation times. Trees with this property are also known as crisp decision trees [10].

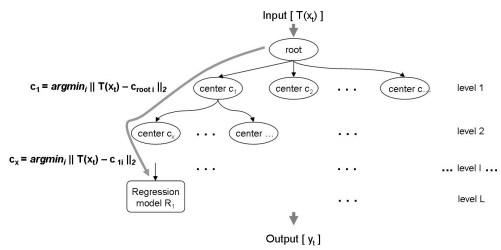

Figure 2: Illustration of a decision tree model

The most probable subregressor is found by the consecutive similarity analysis of the transformed input vector $T(x_t)$ to all centers c_{ji} reachable by the current decision center c_j at level l,

$$c_{j+1} = argmin_{c_{ji}} \parallel T(x_t) - c_{ji} \parallel_2 \tag{6}$$

where $l \in [1, 2, \ldots L]$, $i \in [1, 2, \ldots \mathrm{num}(c_j)]$ and $\mathrm{num}(c_j)$ is the number of centers reachable by c_j. This loop is returned until a final leaf is found. An example is given in figure 2 where the arrow on the left highlights the path taken by an input $T(x_t)$ over the centers c_1 and c_x reaching regression model R_1.

The induction of the decision tree [11] is realized by a probabilistic approach [12]. A cross-validation strategy allows the overfitting of the global model while for each iteration step, the local regressor with the highest error on a predefined testing data set is further split into new submodels. The centers for the new decision nodes are calculated by a k-means algorithm [12].

3 AUTOMOTIVE APPLICATION

This section illustrates the modeling of a charge-pump circuit as an analog-digital system typically used in automotive applications and outlines the influence of the transformation dictionary on model accuracy. Additionally, the embedding of the generated model in different simulators will be highlighted.

A charge-pump is realized by an array of capacitors that generate a voltage larger than the supply voltage from which they operate. In order to describe the functionality of the circuit, seven functional components are outlined in the transistor-level schematic (see figure 3). The system contains six input ports in which two are voltage supplies (V_1, V_2), one represents a current A_1 and another is ground. The remaining ports are digital signals D_1, D_2, where D_1 is the charge-pump clock and D_2 triggers the switching of the connected load. The components highlighted by A) are used as level-shifter between the ground-referenced clock and the voltage source, given by components B). The pump units are implemented as shown in C), while D) represents a diode array for voltage summation. The components emphasized by E), F), and G) act as controller and stabilizer for the output voltage CP.

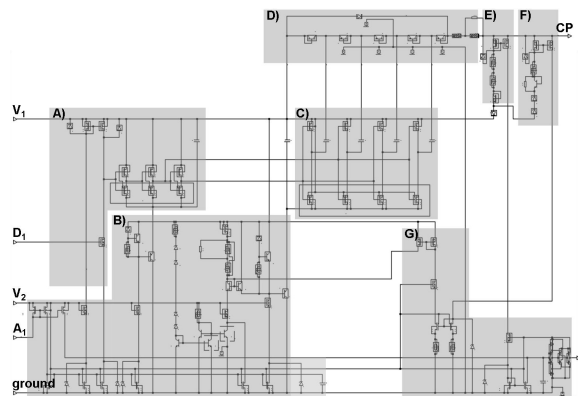

Figure 3: Transistor-level schematic of the charge-pump circuit

Basically, 10 transient simulations were run to create an adequate set of training and testing data. For all simulation runs the signals were designed to produce different stimulation patterns in order to get an extensive characterization of the nonlinear behavior of the original circuit. Therefore, the signal sources were stochastically simulated within the circuit specification.

Based on $4 * 10^5$ simulation vectors $3 * 10^3$ samples were used for the training process in order to achieve an approximation function

$$f : f(D_1, D_2, V_1, V_2, A_1) \rightarrow CP \tag{7}$$

where the other simulation vectors were used as testing data. After integration of the obtained model into the Saber simulation environment, the model was validated by simulation with randomly chosen parameterizations of the signal sources. Afterwards, the model was analyzed for accuracy and simulation speed-up compared to the original circuit. Table 1 shows the results obtained for a set of generated models with an increasing number of submodels. The results refer to highly accurate models with a generalization error of approx. 3.2 % at its best. Additionally, the model generalization properties are increased with the model order while the simulation

speed-up is reduced somewhat. The later observation is given by a longer model runtime for increasing model orders that directly affects the calculation speed-up.

order	model error[%]	*speed-up* factor
global	17.1	172
2	14.7	154
4	6.1	113
8	3.2	106

Table 1: Model error [%] and speed-up factor for the generated CP models within Saber

Since the generated models are represented by a mathematical formula they can be easily embedded into different description languages. The interfaces were realized by C-functions and adapted to each simulator's guidelines. Figure 4 shows the simulation results for a trained model embedded in Simulink, MAST, VHDL-AMS and Verilog-AMS based designs, using the simulators Simulink, Saber, AdvanceMS and AMSDesigner respectively. As given by the envelope of the superposed model errors for the operation in different simulators on a testing signal, the model shows no significant difference in its generalization properties. Besides, the generalization error and speed-up factors correspond with the results shown in table 1.

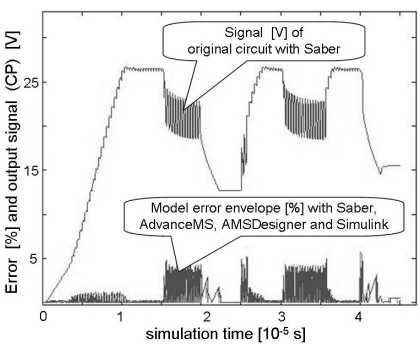

Figure 4: Comparision of the generated model -embedded in different electrical simulators- with the original circuit

The following investigation determines the influence of the transformation library (section 2.4) on the model accuracy. Therefore, the global and decision tree modeling setups were applied on the bare $\vec{x}$ and transformed $T(\vec{x})$ input data. As shown in table 2 the models without dynamic information of the dictionary are not able to reproduce the systems behavior. Although the SVMs are universal nonlinear estimators, this results emphasize the representation of the dynamic information by predefined functions in the regressor space.

4 CONCLUSIONS

The experimental results presented in this work recommend the application of support vector machines for

Transformation Dictionary	global model	order 8
applied	18.1	3.3
not applied	$2E6$	$3.2E6$

Table 2: Model error [%] with and without the transformation dictionary functions

data-based modeling of A/MS systems. The given example demonstrates the efficiency of this approach leading to a simulation speed-up of approximately 100 at its best while model accuracy is close to 95%. In this context the extraction of dynamic properties by a transformation dictionary was shown to possess high influence on model performance.

Additionally, the models can be embedded in different system/hardware description languages like Simulink, MAST, VHDL-AMS and Verilog-AMS. Since the proposed strategy works without previous knowledge of the original circuits its application to other physical domains might be another valuable topic for further industrial research. A further step involves the characterization of the described benefits concerning model error and speed-up compared to a manually generated model.

REFERENCES

[1] MEDEA. *MEDEA+ Design Automation Roadmap - Design Automation Solutions for Europe*, volume 5. MEDEA+ Office, 2005.

[2] A. J. Smola and B. Schölkopf. *A Tutorial on Support Vector Regression.* NeuroCOLT2, Technical Report TR, Royal Holloway College, London, 1998.

[3] R. Vemuri M. Ding. A two-level modeling approach to analog circuit performance macromodeling. In *Proceedings of the Design, Automation and Test in Europe Conference*, 2005.

[4] M. Ding and R. Vemuri. An active learning scheme using support vector machines for analog circuit feasibility classification. In *Proceedings of the International Conference on VLSI Design*, 2005.

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

[6] F. DeBernardinis, M. I. Jordan, and A. SangiovanniVincentelli. Support vector machines for analog circuit performance representation. In *Proceedings of the Design Automation Conference*, pages 964–969, 2003.

[7] V. Ceperic and A. Baric. Modeling of analog circuits by using support vector regression machines. In *Proceedings of the International Conference on Electronics, Circuits and Systems*, pages 391–394, 2004.

[8] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[9] O. Nelles. *Nonlinear System Identification.* Springer-Verlag, Berlin, 2001.

[10] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*, 22(2):198–212, 2003.

[11] J. R. Quinlan. Induction of decision trees. *Journal of Machine Learning*, 1:81–106, 1986.

[12] R. Tibshirani T. Hastie and J. Friedman. *The Elements of Statistical Learning.* Springer - Series in Statistics, New York, 2001.