Performance Modeling of Analog Integrated Circuits using Least-Squares Support Vector Machines

Tholom Kiely and Georges Gielen
Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT-MICAS
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
gielen@esat.kuleuven.ac.be

Abstract

This paper describes the application of Least-Squares Support Vector Machine (LS-SVM) training to analog circuit performance modeling as needed for accelerated or hierarchical analog circuit synthesis. The training is a type of regression, where a function of a special form is fit to experimental performance data derived from analog circuit simulations. The method is contrasted with a feasibility model approach based on the more traditional use of SVMs, namely classification. A Design of Experiments (DOE) strategy is reviewed which forms the basis of an efficient simulation sampling scheme. The results of our functional regression are then compared to two other DOE-based fitting schemes: a simple linear least-squares regression and a regression using posynomial models. The LS-SVM fitting has advantages over these approaches in terms of accuracy of fit to measured data, prediction of intermediate data points and reduction of free model tuning parameters.

1. Introduction

Performance modeling of analog circuits is a topic of major importance in the light of the increasing levels of abstraction that are required for the architectural exploration, design and verification of increasingly complex mixed Analog-Digital integrated circuits and systems. Performance models are mathematical models that relate the achievable performances of an analog circuit to the range of the design variables. These models are essential in architectural exploration, in system definition, in the acceleration of analog circuit synthesis or in hierarchical synthesis of more complex analog blocks.

Earlier work on analog performance modeling has been based either on analytic symbolic techniques [1] or on regression fitting of simulation data [2]. Symbolic techniques are limited to rather linear circuit performance characteristics. Regression techniques can handle any simulatable characteristic, but the designer must first define a regression template (e.g. polynomial, posynomial...).

Earlier work typically fitted one global model for the entire space, sometimes resulting in large errors in data points not part of the initial regression set. In this paper, Support Vector Machines are used as template for the regression fitting. It will be shown that this results in improved accuracy, also in new data points, partly because the support vectors model the performance space more locally.

This paper is organized as follows: section 2 defines performance modeling of analog circuits, with particular emphasis on the distinction with feasibility modeling. Section 3 provides some background on Support Vector Machines, with particular focus on the Least-Squares version introduced by Suykens [3]. We describe how performance modeling of analog ICs may be implemented with LS-SVMs as a function regression problem and contrast the presented approach with the classification approach presented by De Bernardinis et al [4]. In section 4 we present a practical example which illustrates the advantages of the method presented with respect to parametric fitting when combined with a sparse sampling scheme from Design of Experiments (DOE) theory. We present our conclusions in section 5.

2. Performance and feasibility models: levels of abstraction

2.1. Performance Modeling

Performance modeling of an analog circuit may be viewed as an abstraction of behavioral modeling. This is represented schematically in Figure 1. A behavioral model B of a circuit with design parameters X (transistor sizes, voltage and current biases, etc.) describes its input-output relationship: regarding the circuit as a system subject to an excitation E, the output signal Y is represented by the model as a function of the input sig-

1530-1591/04 $20.00 (c) 2004 IEEE

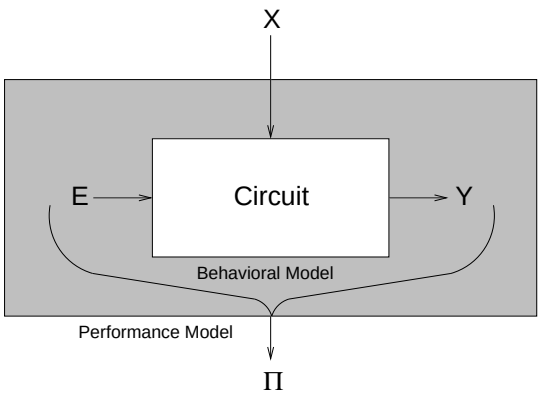

Figure 1. Performance Model of an Analog IC.

nal and the design parameters:

$$Y = B(E,X). \tag{1}$$

At a higher level, however, we may consider some metric that generalizes the relationship between a range of input and output signals: this is called a *performance model* Π:

$$\Pi = P(Y,E). \tag{2}$$

Typical examples are signal gain, gain bandwidth, slew rate, etc. The purpose of performance modeling is to mirror the designer's mode of reasoning by representing the performance as depending only on the design parameters X, that is, to eliminate E and Y from (1) and (2):

$$\Pi = \Pi(X) \tag{3}$$

In a realistic case, a performance model will in general be a non-linear real-valued function over a high-dimensional space of circuit parameters. Clearly, being able to find a representation of $\Pi(X)$ is essential to analog and mixed-signal IC development:

- *architectural evaluation* involves choosing from a range of possible circuit solutions, each characterized by a range of performances $\Pi_{1,i}, \Pi_{2,i}, \ldots$, which depend on the design parameters X_i of that circuit's configuration;
- *optimal circuit design* involves finding $X^\star$ for which the optimal performance $\Pi^\star$ results that satisfies certain constraints;
- for complex designs, *system-level verification* requires as compact a model as possible of each sub-block to make (co-)simulation feasible. This entails being able to replace lower-level abstractions B_i of sub-blocks where possible by their corresponding performance models Π_i. This also allows hierarchical synthesis.

The main focus of this paper is to use Support Vector Machine (SVM)-based function regression to develop a performance model representation $\Pi(X)$ for a particular analog circuit.

2.2. Feasibility Modeling

One other kind of modeling will be considered here: a *feasibility model* is a collection of circuit performances $\{\pi_1,\ldots\}$ and specifications $\{s_1,\ldots\}$ such that a *feasibility region* X_f can be defined, giving the set of design parameters for which *all* performances are 'within spec':

$$X_f = X_1 \cap X_2 \cap \ldots X_n \tag{4}$$

where

$$X_i = \{X : s_i(\pi_i(X)) = 1\} \tag{5}$$

and $s_i(\cdot) = 1$ indicates that the specification is met for that performance over a range X of the design parameters. The item of interest here is not the performances themselves, but the design parameter space over which a $\{1,0\}$-valued function $\mathcal{P}$ can be defined and for which $\mathcal{P}(X_f) = 1$ means that all performances are within specification for the range X_f of design parameters. The problem of representing $\mathcal{P}$ is a *classification* problem for which a Support Vector Machine classifier is a good candidate. De Bernardinis *et al*. [4] used Support Vector Machines (SVMs) to generate feasibility models for analog circuits.

We contrast the latter use of SVMs with their use in the present paper for real-valued function regression applied to performance models as defined above. In the next section SVMs for classifier applications will be introduced first, as the basic concepts are more straightforward and represent the original motivation of SVM theory. We then introduce the SVM-based function regression formulation and apply it in the subsequent section to an analog circuit performance model.

3. Support Vector Machines

3.1. The Classification Problem

SVMs (Vapnik, [5]) were proposed originally in the context of machine learning, for classification problems on (typically large) sets of data which have an unknown dependence on (possibly many) variables. We consider each of N data points $x_k \in \mathbb{R}^n$, $k = 1,\ldots,N$ to be associated with a label $y_k \in \{-1,+1\}$ which classifies the data into one of two sets. In the simplest SVM formalism, the problem of finding a general representation of the classifier $y(x)$ becomes that of the construction of a hyper-plane $w^T x_k + b$ which provides 'maximal separation' $\frac{2}{\|w\|^2}$ between points x_k be-

longing to the two classes. This gives rise to an optimization problem of the form

$$P:\ \min_{w,b} \quad \tfrac{1}{2}w^T w \quad s.t. \quad y_k[w^T x_k + b] \ \geq 1 \qquad (6)$$

where the $\frac{1}{2}w^T w$ term represents a cost function to be minimized in order to maximize separation. The constraints are formulated such that the nearest points x_k with labels +1,-1 are (with appropriate input-space scaling) at least $\frac{1}{\|w\|^2}$ distant from the separating hyper-plane. To solve this 'primal' minimization problem, we construct the dual maximization of (6) using the Lagrangian form:

$$D:\ \max_{\alpha}\ \mathcal{L}(w,b;\alpha) \qquad (7)$$

where

$$\mathcal{L}(w,b;\alpha) = \frac{1}{2}w^T w - \sum_{k=1}^{N} \alpha_k (y_k[w^T x_k + b] - 1) \qquad (8)$$

and α_k are the Lagrange multipliers.

After applying the conditions for optimality[1]

$$\frac{\partial \mathcal{L}}{\partial w} = 0, \quad \frac{\partial \mathcal{L}}{\partial b} = 0, \quad \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \qquad (9)$$

and eliminating w by expressing it in terms of $\alpha = [\alpha_1, \ldots, \alpha_N]$, we arrive at a Quadratic Programming (QP) problem:

$$\min\ (\alpha Q \alpha + B\alpha) \qquad (10)$$

for suitably defined matrices Q, B. Having solved for α, the following classifier representation is obtained:

$$y(x) = \text{sign}\left[\sum_{k=1}^{\#SV} \alpha_k y_k x_k^T x + b\right] \qquad (11)$$

where #SV represents the number of non-zero Lagrange multipliers α_k, called support values, corresponding to the input data x_k. The SVM representation will be *sparse* if only a few of the input data, called support vectors, are 'near' to the separating hyper-plane.

A key feature of Support Vector Machines is the ability to replace the input data by a (non-linear) function $\phi(x)$ operating on the input data. This may be viewed as mapping the input data to a higher (possibly infinite) dimensional space[2], to enable classification of data that is not linearly separable in the original input space. To do this, we formally replace $x_k^T x$ (the dot product between a support vector x_k and any point x of the input space) in equation (11) by $\phi(x_k)^T \phi(x)$ to represent the action of this mapping, obtaining:

$$y(x) = \text{sign}\left[\sum_{k=1}^{\#SV} \alpha_k y_k \phi(x_k)^T \phi(x) + b\right] \qquad (12)$$

In the case where $\phi(\cdot)$ is infinite-dimensional, we invoke the so-called 'kernel trick': the expression $\phi(x_k)^T \phi(x)$ may under certain conditions be replaced by a kernel function $K(x_k, x)$. An equivalent interpretation is that the kernel function is a suitably-defined dot product $< x_k, x >$ replacing $x_k^T x$ in the Hilbert space defined by the mapping ϕ. In this way we avoid ever having to represent the mapping ϕ explicitly. In either case, the use of a kernel function allows the SVM representation to be independent of the dimensionality of the input space. As we shall see in the following section, the model for even a relatively simple circuit can become quite high-dimensional.

The difficulty of fitting a polynomial model is exponential in the number of dimensions in the input space (cf. the discussion of Harjani and Shao in [6]). This 'curse of dimensionality' can be circumvented by choosing as kernel a Radial Basis Function (RBF), of the general form:

$$K(x_k, x) = \kappa(-\|x - x_k\|_2^2 / \sigma^2) \qquad (13)$$

for suitable function κ and where σ is a tuning parameter. In the remainder of this paper, we will invariably use the following exponential RBF Kernel:

$$K(x_k, x) = \exp(-\|x - x_k\|_2^2 / \sigma^2) \qquad (14)$$

which indeed corresponds to an infinite-dimensional feature-space mapping ϕ. This choice of kernel function will be further justified in the following sections in relation to the Least-Squares variant of SVM and its application to analog circuit performance modeling.

3.2. Least-Squares SVMs for Function Regression

In the case of function regression, the labels $y_k \in \{-1, +1\}$ represented by the {+1,-1}-valued function $y(x)$ are replaced by real-valued $y_k \in \mathbb{R}$. We write

$$y(x) = w^T \phi(x) + b \qquad (15)$$

where $x_k \in \mathbb{R}^n$, $y, b \in \mathbb{R}$, $w \in \mathbb{R}^{n_h}$ and $\phi(\cdot) : \mathbb{R}^n \longrightarrow \mathbb{R}^{n_h}$ is the mapping to the high (n_h-, potentially infinite-) dimensional feature space. For the Least-Squares SVM regression, we introduce error variables for the fitting problem as follows:

$$e_k = w^T \phi(x_k) + b - y_k \quad k = 1, \ldots, N \qquad (16)$$

and construct the following primal optimization problem

$$P:\ \min_{w,b,e} J_P(w,e) = \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2 \qquad (17)$$

1 See, for example, Suykens [3], Chap 2.

2 We shall subsequently call this the 'feature space'.

together with the N constraints (16). This formulation involves the trade off between a cost function term (as in equation (6)) and a sum of squared errors governed by the trade-off parameter γ. In the regression formalism the term $\frac{1}{2}w^T w$ is no longer related to hyper-plane separation, but instead determines the 'smoothness' of the resulting model. In fact, the primal problem in the LS-SVM formalism is wholly equivalent to a ridge regression [7] problem formulated in the feature space, with parameter γ performing the role of smoothing parameter. Proceeding to the dual Lagrangian-based formulation

$$D: \max_{\alpha} \mathcal{L}(w,b,e;\alpha) \tag{18}$$

$$\mathcal{L} = J_P(w,e) - \sum_{k=1}^{N} \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\} \tag{19}$$

and applying the kernel trick, we arrive at the model (cf. [3], p. 98-99)

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x) + b \tag{20}$$

Thus the function is modeled as a linear sum of kernel functions weighted by the support values (nonzero Lagrange parameters) α_k. Though (18) is a QP problem, it can be shown that in LS-SVM using the Kernel trick, a linear problem results. We have from the conditions for optimality

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\Rightarrow\; \alpha_k = \gamma e_k,\; k = 1,\ldots,N \tag{21}$$

allowing us to eliminate not only w as before but also e_k, yielding the $(N+1)\times(N+1)$ system of equations:

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N & K + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{22}$$

where α is the N-vector defined for equation (10), y the corresponding vector of y_k-values, 1_N is the unity N-vector $[1;\ldots;1]$ and K is the $N \times N$ 'Kernel matrix':

$$K = K(x_k, x_l) = \phi(x_k)^T \phi(x_l) \;\; k,l = 1,\ldots,N. \tag{23}$$

Another important consequence of (21) is that the model is no longer sparse, because the support values α_k are directly proportional to the fitting errors e_k which are in general non-zero. This means that a LS-SVM model does not have the typical sparseness of the original Vapnik SVM. However, for the purposes of function regression, our experiments suggest that achieving good levels of accuracy with the Vapnik formulation involves tuning parameter settings which tend in any case to reduce sparseness. Various methods are available to reduce the size of LS-SVM models, such as support-vector pruning [8] and incremental training.

The fact that the LS-SVM formulation involves solving the linear system (22) makes it efficient for iterative training. An additional advantage when the exponential RBF kernel (14) is used is the small number of tuning parameters set by the user, namely the regularization term γ and the kernel bandwidth σ^2. This minimizes the cost of iterating to find optimal settings for these parameters to allow accuracy/compactness trade-offs. This is discussed further in the practical example of the next section.

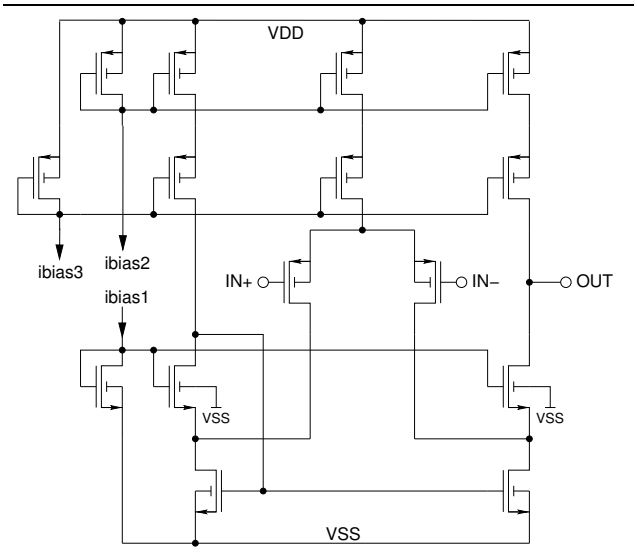

Figure 2. Operational Transconductance Amplifier (OTA).

4. LS-SVM performance model of an analog circuit

4.1. Model specification

In figure 2, a high-speed CMOS operational transconductance amplifier (OTA) circuit is shown. For this circuit we considered the following performance measures: low-frequency gain (AV_{LF}), unity-gain frequency (f_u), phase margin (PM), input-referred offset and positive and negative slew rate (SR_p, SR_n). For nominal supply-voltage, process and temperature design, these may be considered functions of 13 independent voltage and current biases.

An operating-point driven (OPD) sizing scheme [9] is used to specify the device W,L sizes which correspond to the voltage biases (Vds, Vgs of transistors) which are taken, together with three current biases, as independent variables. The input space is therefore $X \subset \mathbb{R}^{13}$.

For each input variable a logarithmic scaling is used such that all scaled variables x_k are real numbers in the $[0,1]$-range. For example, where x_k corresponds to a bias voltage

$v_k \in [lb_k, ub_k]$, we scale as follows:

$$x_k = \frac{\log\left(\frac{v_k}{lb_k}\right)}{\log\left(\frac{ub_k}{lb_k}\right)} \tag{24}$$

This has the advantage of facilitating a good choice of RBF kernel tuning parameter σ^2. With the aforementioned restriction on input variables x_k, the (RBF argument) expression $\|x - x_k\|_2^2$ lies in the range $[0, n^2]$, where n is the input dimensionality (13 in the present case). Allowing each experimentally-obtained performance data point to contribute roughly equally to the overall model suggests using a wide bandwidth. In practice, we found setting $\sigma^2 \simeq \frac{3n^2}{4}$ (i.e., about 127), gave the best results. Setting a narrower bandwidth allows more non-linear behavior to be modeled, but makes it harder to keep the average fitting error low and maintain accuracy when pruning the model.

The second tuning parameter γ can be chosen based on the expected smoothness of the resulting performance model. Setting a large value (say 10^4) strongly penalizes fitting error in the measured points, but can degrade accuracy when using the model predictively. Choosing a low value $\gamma \simeq 1$ is not appropriate when using the exponential RBF (14) to model performances which are often approximately linear or weakly quadratic in most input variables: such behavior is best approximated by allowing the support values α_k to scale strongly with deviations from the points fitted (recall equations (20) and (21)). Setting $\gamma = 100$ was found to be a good choice to keep the fitting error low while maintaining a good generalization (predictive performance) of the model.

4.2. Design of Experiments

A compact sampling scheme is essential for performance modeling based on simulation. While it has been shown above that the LS-SVM approach with RBF kernel has the attractive properties that (a) its solution requires solving a linear system and (b) the solution complexity is independent of the dimensionality of the input space, nevertheless the question arises of sampling the input space with sufficient density to ensure good coverage for the model and therefore good generalization. Techniques from *design of experiments* (DOE) provide a mathematical basis for selecting a limited but 'optimal' set of sample points for the presented modeling problem [10, 11].

A typical response surface for one performance (Phase Margin) is shown in figure 3. It is shown as a plot of two input variables corresponding to a voltage bias and current bias in the OTA circuit of figure 2. This graph illustrates the usefulness of logarithmic input scaling such that performances exhibit weakly non-linear behavior over a wide range of each input variable. On the assumption of weak

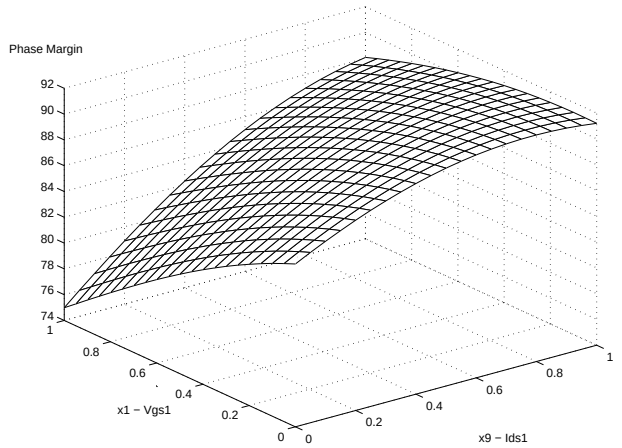

Figure 3. 2-D phase margin performance response.

non-linearity in the resulting performance model, a reasonable choice of sampling scheme is a particular type of fractional-factorial plan called an *orthogonal array* [12]. For purposes of direct accuracy comparison with the posynomial fitting scheme of Daems [2], we use as sampling scheme an orthogonal array of strength 3, which allows uncorrelated estimation of linear, quadratic and interaction terms of a second-order polynomial phenomenon [13].

4.3. Results and Comparison

Using the sampling scheme described in the previous section, 243 SPICE simulations (DC operating point, AC sweep or transient, depending on the performance) were run for each performance parameter and the resulting data were used to train an LS-SVM model for each performance, using the lssvmlab toolbox [14]. To facilitate comparing models, a scaling of performances is applied: for each performance measure p_k a target $p_{k,spec}$ is set, and scaling is done as follows:

$$p_{k,scaled} = \frac{p_k - p_{k,spec}}{1 + p_{k,spec}} \tag{25}$$

for all performances except unity-gain frequency f_u, for which we write:

$$f_{k,scaled} = \frac{\log\left(\frac{f_k}{f_{k,spec}}\right)}{1 + \log\left(f_{k,spec}\right)} \tag{26}$$

To rate models, two accuracy measures were used for each performance: a root-mean-squared (RMS) error on the scaled performances at the fitting points, and the worst-case RMS error for (scaled-performance) prediction of interpolated points. In the latter case, the interpolation was done

RMS error	Linear-LS	Posynomial	LS-SVM
Fit points	0.072	0.041	0.016
Interpolated	0.084	0.049	0.023

Table 1. RMS error for 3 models of OTA low-frequency gain AV_{LF}.

RMS error	Linear-LS	Posynomial	LS-SVM
Fit points	0.151	0.059	0.024
Interpolated	0.327	0.097	0.030

Table 2. RMS error for 3 models of phase margin PM.

using subsets of $9 \times 243 = 2187$ randomly-sampled data from within the $\mathbb{R}^{13}$-hypercube over which each performance model was trained.

In tables 1 and 2 results are presented for 3 prediction schemes: a simple linear least-squares regression, the posynomial fitting scheme of [2], and the LS-SVM model described here. The two performances considered here are at the extremes in terms of achievable fit with a linear least-squares model: low-frequency gain AV_{LF} (linear model gives relatively low-error fit) and phase margin PM (linear model gives poor fit).

It can be seen from both tables that the LS-SVM model outperforms the other models for both performances shown; this trend is maintained for all performance models generated. Of particular note is how the interpolated (i.e. generalization) model accuracy degrades for each of the three models: a gradual increase in error is observed in the LS-SVM model moving from the weakly (Table 1) to strongly nonlinear (Table 2) performance models. A much greater divergence can be seen in the other models – note especially the increase of the error from 0.084 to 0.327 in the linear least-squares case, compared with an increase from 0.049 to 0.097 in the posynomial model, which includes first-order cross-products $x_i x_j$ and quadratic terms x_i^2, whereas the increase in interpolated-data error is only from 0.023 to 0.030. This shows not only the higher fitting accuracy of the LS-SVM model compared to the other approaches, both for linear and nonlinear characteristics, but also its ability to better keep this accuracy for interpolated (i.e. non-training) data points.

5. Conclusions

LS-SVM has been successfully applied to analog circuit performance modeling considered as a non-linear/arbitrary function regression problem. We compared the general performance modeling problem to the feasibility-model approach, for which SVM classifiers have already been used. We considered various training approaches to trade off accuracy for model compactness. We conclude that LS-SVMs with RBF kernels are flexible enough to be applicable to both linear and non-linear models, while retaining the simplicity of having only a single pair of tuning parameters. The accuracies, also for interpolated data points, show considerable improvement over previous analog circuit performance modeling methods.

References

[1] G.G.E. Gielen, H.C.C. Walscharts and W.M.C. Sansen. "ISAAC: A Symbolic Simulator for Analog Integrated Circuits". *IEEE J. Solid-State Circuits*, 24(6):1587–1597, Dec. 1989.

[2] W. Daems, G. Gielen, W. Sansen. "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits". In *Proc. of the 39th Design Automation Conf.*, 2002.

[3] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing, 2002.

[4] F. De Bernardinis, M.I. Jordan, A. Sangiovanni-Vincentelli. "Support Vector Machines for Analog Circuit Performance Representation". In *Proc. of DAC*, 2003.

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

[6] R. Harjani and J. Shao. Feasibility and Performance Region Modeling of Analog and Digital Circuits. *"Analog Integrated Circuits and Signal Processing"*, (10):23–43, 1996.

[7] G.J. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1989.

[8] J.A.K. Suykens, L. Lukas, J. Vandewalle. "Sparse approximation using least squares support vector machines". In *Proc. of the IEEE Int. Symp. on Circuits and Systems*, volume 2, Geneva, 2000.

[9] F. Leyn, G. Gielen, W. Sansen. "An efficient DC root solving algorithm with guaranteed convergence for analog integrated CMOS circuits". In *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, Nov. 1998.

[10] G.E.P. Box, W.G. Hunter, J.S. Hunter. *Statistics for experimenters: an introduction to design, analysis and model building*. John Wiley & Sons, New York, 1978.

[11] G.E.P. Box and N.R. Draper. *Empirical Model Building and Response Surfaces*. John Wiley & Sons, New York, 1987.

[12] A.S. Hedayat, N.J.A. Sloane, J. Stufken. *Orthogonal arrays: theory and applications*. Springer-Verlag, New York, 1999.

[13] J.C. Zhang and M.A. Styblinski. *Yield and Variability Optimization of Integrated Circuits*. Kluwer Academic Publishers, 1995.

[14] K. Pelckmans, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, J. Vandewalle. *LS-SVMlab : a Matlab/C toolbox for Least Squares Support Vector Machines*. `http://www.esat.kuleuven.ac.be/sista/lssvmlab`.