Linearly graded behavioural analogue performance models using support vector machines and VHDL-AMS

Xianqiang Ren, Tom J Kazmierski
School of Electronics and Computer Science
University of Southampton, UK

Abstract
A concept of linearly graded statistical models for analogue performance evaluation is proposed and a suitable technique for automatic generation of analogue performance models using support vector machines is presented. The analogue system’s behaviour is specified using VHDL-AMS descriptions. Practical application of the technique is demonstrated with a case study of automated analogue performance model generation for an analogue filter.

1 Introduction
Analogue performance modelling is an important phase in analogue synthesis environments. The optimization and verification of analogue designs depends on the quality of the models. There are several important features that are expected from a good analogue performance model. The first one is high accuracy. The models should be able to provide accurate predictions of the behaviour of the design within the whole design space. Another expected feature is high generality. An analogue design process is a multi-stage and multi-level procedure and the objectives are complex, hence a method is more preferable if it can be applied onto variable design levels. Finally, the models need to be able to describe complex systems while reducing the computational expenses. Behavioural-level models are of high interest from this point of view with the increment of circuitries’ scales.

In this paper, a new approach of analogue performance modelling is presented. The salient features of the technique are: 1) the use of linearly graded performance models, instead of the traditional “good-bad” models, and 2) the use of behavioural-level descriptions in a hardware description language (HDL) with a set of generic parameters whose space is explored to build a performance model for further regression analysis.

Analogue performance modelling Most extensively studied method for analogue performance modelling is symbolic analysis [WGS98]. CAD tools of this kind extract equation sets for certain circuit figures to tell designers how the figures are influenced by the circuit parameters. The generated symbolic design equation sets constitute a model of the systems’ behaviour that can be used in analogue synthesis tools. Although simplification and decomposition techniques are developed, symbolic analysis still has difficulties to be applied onto large scale circuits [GR00]. Also, it is reported that fuzzy logic method has been applied onto analogue circuit modelling [TCF96a] and analogue synthesis [TCF96b]. The modelling systems using fuzzy logic contain fuzzy interface block to find relationship between the input vectors and the output vectors. Fuzzy logic inference unit is the block that makes decisions based on fuzzy logic rules in a “IF-THEN” format. The system needs to be trained before being used on un-classified data sets.
Another way is to use statistical methods that have been widely used in VLSI design technology. Statistical models for devices and processes are developed for accurate design estimation [McA03]. Because of the statistical nature of the physical models, plus the heuristic characteristics of analogue systems’ behaviour, analogue circuits and system designs are statistical. Interesting development introduces Support Vector Machines (SVMs) for analogue circuits’ performance representation [BJsV03].

**Support Vector Machines** Support vector machines (SVMs) are introduced by Vapnik in 1992 to solve machine learning problems [Bur98]. Problems that SVMs focus on usually consist classification of large amount of inputs spreading over large feasible spaces and the relationship between the classification method and the input data is implicit or potentially very complex. It has been successfully applied to fields like object recognition [PV98], machine learning [LX04], artificial intelligence [BPO05] etc. SVMs approximation function is in the follow form:

\[
F(x) = \text{sign} \left( \sum_{i} (y_i \alpha_i k(x, x_i) + b) \right)
\] (1)

where \(x_i\) are input samples, \(l\) is number of training vectors, \(\alpha_i\) are weighting multipliers, \(b\) is a constant and \(k()\) is called kernel function. The summation in equation 1 is over a specific set of samples that have non-zero corresponding \(\alpha_i\) called support vectors (SVs).

SVMs employ hyperplane classifiers to decide on which side that the input points lie. Hyperplane classifiers provide facilities to do classification in the feature space. SVM employs a mapping function to map points from the input space to the feature space to increase the data separability through hyperplanes. The mapping function does not have to be explicitly known because kernel functions can be used to replace the dot product of mapping functions thus represent hyperplane classifiers and the mapping mechanism.

![Figure 1: The testing error as a function of SVM control parameters C and γ. Thin solid and dashed lines show the change of the testing error and training error rates accordingly.](image)

One of the most popular kernels is Radial Basis Function (RBF) kernel and it is expressed as \(k(x, x') = exp(-\gamma \|\bar{x} - \bar{x'}\|^2)\), \(\gamma > 0\), where \(\gamma\) is the control parameter of the kernel. In this application, the RBF kernel is selected mainly because of its ability to handle nonlinear cases [BJsV03]. Secondly, compared with the polynomial kernel, it has less hyperparameters to decide thus reduces the difficulty of the parameter selection process. Finally, the RBF kernel has less numerical difficulties than other kernels [wHcCjL01].

The modelling process using SVMs is composed of training and testing process. The construction of the classifiers relies on the training process. The training sets populate the hyperplane that can give maximum margin between the “positive” and “negative” classes. The testing procedure is the process to verify the models and check the quality of the models. Once the two processes are done, the model can be used on un-known data.
Although group of samples are used to model the separation hyperplanes in the models in the SVM training procedure, there will still be mis-classified ones. Before the minimum testing error is reached, the SVM is under-trained as the testing accuracy can still increase. Beyond the optimal region, further training can still reduce training error however the testing error starts to increase and the models are over-trained. Figure 1 shows the relationship between the two errors, the SVM control parameters of the RBF kernel ($C$ or $\gamma$).

2 Model structure and the automated modelling process

The data structures of the objective models and the information flow graph are presented in figure 2. The aim is to generate the model matrix $M$. It contains all the sub-models that relate uniformly graded ranges of each performance parameter to all the design parameters. Each sub-model is a collection of model parameters and coefficient weights for all the SVs. The design space matrix $D$ contains design parameter sets. Design parameters are assigned ranges which are randomly sampled with uniform distribution. Then, the performance space matrix $P$ is determined by simulating all the points in the design space to determine the performance parameters observed. The grading routine uses the minimum and maximum values for every performance parameter, divides them into ranges and labels the sampled points with different performance values. The training process uses the design space and the extracted graded performance space as the input and calculates the model matrix $M$.

![Figure 2: Illustration of the data structures and information flow graph of the design.](image)

In figure 2, the design space has $j$ sets of design parameters and each design parameter set is composed of $i$ values of all the design parameters that are $D_1$ to $D_i$. The performance space has the dimension $m$ which is the number of performance parameters specified by the user and the entire space is divided into $q + 1$ uniformly divided ranges. Performance figures are labelled with “1” or “0” according to whether they lie within a given boundary or not. There are $q$ groups
of performance matrices, which are trained to generate $q$ model sets for the $m$ performance parameters to form the model matrix $M$.

The system’s behaviour is described in VHDL-AMS by Differential Algebraic Equations (DAEs). Generic parameters defined in design entities provide a parameter set that can be used for performance exploration. A valid range is assigned to every generic and is sampled. Then the sample points form input parameter sets for the modelling process. Simulations are carried out with every set of the parameters. Performance figures are captured from simulation results for each such generated set of generic values to form a performance data set. Training and testing of the SVM is then carried out on the performance data set. Performance models constructed in this way are then used to classify the unclassified design sets according to the linearly graded ranges. The design flow graph is shown in figure 3.

![Diagram](image)

Figure 3: The automated performance model building process.

### 3 Experiment

The technique has been applied to generate performance models for a second order low-pass analogue filter shown in figure 4. Filters of this kind are used e.g. in $\Sigma\Delta$ modulators.

```vhdl-ams
entity filter is
generic(
a1 : real := -0.8627;
a2 : real := -0.3673;
t1 : real := 10000.0;
t2 : real := 10000.0;
terminal in : electrical;
end filter;
architecture bhv of filter is
terminal m : electrical;
quantity mv across m to op;
quantity pv across p to op;
quantity vin across in to op;
quantity vout across out to gnd;
begn
mv := vin + pv"*a1 + vout"*a2;
pv"*dot := t1*mv;
vout"*dot := t2*pv;
end architecture bhv;
```

Figure 4: Signal flow graph of the 2nd order analogue filter example and the VHDL-AMS description.

Performance of this second order analogue low-pass filter has been analysed. VHDL-AMS
description of the filter contains four generic definitions: \(a_1\), \(a_2\), \(t_1\), and \(t_2\) that represent the amplifier gain and integrator time constants corresponding. Three performance factors are observed: gain (G), cut-off frequency \((F_c)\), and over-shoot (R). 50000 simulations were carried out for performance exploration and generate models to represent the relationship between the 4-dimensional design space and the 3-dimensional performance space.

Figure 5: SVM classification accuracy and SV number for each set of performance models at each range ratio \(\tau\).

When the models are built, classification accuracy is tested and SVs are observed. Figure 5 shows classification accuracies and number of support vectors of the boundaries that divide the performance space into 10 different ranges according to the range ratio factor \(\tau\). With the increment of the number of SVs, the boundaries will be more blurred thus the accuracies are lower than those of the boundaries with less SVs. The maximum classification accuracies shown in figure 5 a, which indicate the robustness of the method in the presence of blurred boundaries between performance ranges and with random concentrations of design points.

Figure 6: A 3D accuracy diagram showing model construction for boundary of \(F_c = 40\%\) and two projections onto 2D planes showing optimal, under-trained and over-trained regions.

The control factor \(C\) and \(\gamma\) in the RBF kernel affects the performance model accuracy as shown in figure 6. It is consistent with the theory introduced in figure 1. In figure 6 the top plot is a 3D accuracy diagram showing model construction for boundary of \(F_c = 40\%\). The two sub-plots below are projections onto 2D planes showing optimal, under-trained and over-trained regions. The sub-plot a) corresponds to the region where \(C\) is constant while \(\gamma\) varies within the whole grid-search range, and the sub-plot b) is a projection onto the plane formed by \(C\) and accuracy.
3.1 Conclusion and Future Work

A linearly graded statistical approach to analogue performance modelling is proposed as well as a corresponding simulation-based SVM modelling method for automatic model generation. The method relies on behavioural descriptions in VHDL-AMS and can be adapted to other standard HDLs. The method is general and the model generation process is fully automatic. The method has been successfully applied to analyse the performance of a second-order low-pass filter. As linearly graded models provide accurate performance evaluation over the entire performance space, the method can potentially lead to improved analogue synthesis techniques.

References


