Feasibility Region Modeling of Analog Circuits for Hierarchical Circuit Design

Jianfeng Shao and Ramesh Harjani
Department of Electrical Engineering
University of Minnesota, Minneapolis, MN 55455

Abstract—During hierarchical design, it becomes essential at each level of the hierarchy to evaluate if a sub-block design is feasible and if so which design style is the best candidate for the particular problem. We propose a general methodology for evaluating the feasibility and the performance of sub-blocks at all levels of the hierarchy. In this paper we concentrate on techniques to model the feasibility region. The methodology is general and can be used for both analog and digital circuits. Macromodels are developed and verified for analog blocks at different levels of hierarchy.

1 Hierarchical Circuit Design

An increasing percentage of IC’s have analog circuit designs in them. These and other requirements stress the need for automatic synthesis tools for analog circuits. Hierarchy plays a significant role in the design of digital and analog circuits. A large design can be broken up into smaller sub-blocks at the different levels of a hierarchy. An example hierarchy for an A/D converter is shown in Figure 1. Let us consider a typical design flow for the A/D design shown in this figure. The design process is begun by first selecting a particular design style or architecture. At each level of the hierarchy we are presented with a number of candidate design styles for each function block [1]. Each of these design styles provide the same functionality but provide different performance tradeoffs. For example, other design styles for the A/D converter include the flash, successive approximation, and pipelined architectures. However, before we can make a selection we need to know which design styles are feasible, i.e., can be designed to meet the performance specifications. Additionally, once a number of design styles have been determined to be feasible we need to select the best candidate among these design styles, i.e., the one that provides the best performance.

Obviously, one possible solution to the selection problem is to exhaustively try out all the options and then select the best candidate among them. Unfortunately, this means that we need to exhaustively design each design style to the very bottom level. This is important in analog circuit design because the performance of any circuit is highly dependent on the process parameters. For example, an opamp with a gain of 90dB and a unity gain bandwidth of 15MHz that is designable for a 1.2\mu M process of manufacturer A may not be designable for the same feature size process of manufacturer B. Unfortunately, as the number of levels and branches at each level of the hierarchy increase, the design time increases exponentially. The simple generate and test solution doesn’t fit well within a hierarchical framework.

As a solution to this problem, we propose a numerical macromodeling solution that generates macromodels a priori. Two macromodels are generated, a feasibility macromodel and a performance macromodel. The macromodels are built bottom-up. The primary advantage of developing hierarchical macromodels is that the macromodels need only be built once. They can then be used during design time to accommodate a complete top-down design. Therefore, the performance or feasibility of a block at level A can be determined by evaluating the macromodels at that level rather than having to travel to the very bottom of the hierarchy. Once a design style is selected, then translation of the performance specifications down the hierarchy can be performed by other means [2]. OASYS [1, 2] is an example of analog synthesis system that uses similar abstractions to efficiently design analog circuits. We shall use the OASYS system to evaluate our macromodeling approach. However, the methodology being described is general and can be used with any other hierarchical design system.

2 Macromodeling Methodology

To ensure that the methodology is general, i.e., can be applied at all levels of the hierarchy and can be applied to any circuit block, it is essential that the methodology be abstract and not depend on the implementation details of any of the functional blocks. To this effect, we have developed a macromodeling methodology using radial basis functions [3] to obtain the feasibility and the performance macromodels. The feasibility macromodel takes as input the set of performance specifications and generates a binary valued output: feasible or not-feasible. That is to say that for any performance specification that lies within the feasibility region a circuit can be designed
to meet specification. Performance macromodels are generated within each feasible region. Performance macromodels are mappings from the feasible input specification space to the realizable performance space. At each level of the hierarchy each design style has a single feasibility macromodel and a set of performance macromodels. There is a performance macromodel for each performance metric. Examples of performance metrics include area, power, noise, etc. The overall procedure consists of first generating the feasibility macromodel and then generating the performance macromodels. Both model generation procedures share a number of details. They both perform variable screening and grouping. They use static fractional factorial and volume slicing techniques to perform experimental runs. And they use radial basis functions for regression analysis. However, as mentioned earlier, here we shall concentrate on the feasibility macromodel. Details about the performance macromodel can be found in [4].

The feasibility model generation procedure consists of:

1. Variable screening and grouping using a static fractional factorial plan.
2. Boundary point generation using vertical binary search.
3. Regression analysis using radial basis functions.

We will first present some ideas on experimental design and then follow it with a short discussion on variable screening and grouping. Following which we shall talk about regression analysis and then provide more details about the procedure used to generate the boundary points for the feasibility macromodel.

**Experiment design:** Experiment design techniques are employed to build an appropriate experiment plan. The use of an appropriate experiment plan results in substantial savings in the number of experimental runs. In our approach, we use both static and dynamic experimental design techniques. We use a static factorial design technique to measure variable significance and we use a dynamic technique called *volume slicing* to generate the data points for regression. During volume slicing, we dynamically increase the number of experimental runs where the response surface is more complex and decrease the number of experiments where the response surface is smooth. The adaptive volume slicing technique is shown in two dimensions in Figure 2. We start by performing four experimental runs at the corner points, \( (C_1,C_2,C_3,C_4) \), of the domain \( D_0 \). We then generate the macromodel using these regressors. An additional experimental run is performed at the center of the square, \( (C_0) \), and the macromodel is also evaluated at this point. If the difference between the experimental run and the macromodel is sufficiently small then nothing is done. Else, if the error is large then four additional experimental runs are performed such that the original domain is now divided into four equal domains, \( D_1, D_2, D_3, D_4 \). The macromodel is now generated for each of these new domains and the overall process repeats itself until the error is sufficiently small.

**Variable screening and grouping:** Through systematic experimental runs the significance of each variable, i.e., its effect on the output response, is estimated. Experimental runs are performed using the \( 2^{n-p} \) fractional factorial plan [5]. The variables below a certain threshold level are neglected. Selected variables are further grouped into layers with the more significant variables in the upper layers. This classification reflects the varied influences of different variables on the output response and has a significant impact on the complexity of the modeling technique for high dimensions [4].

**Boundary point generation:** The feasibility surface in the input variable space defines the region in which a circuit is designable. Once a feasibility surface for a design style is established then for any set of input specifications we can easily establish if a circuit in that topology can be designed to meet specifications. This provides substantial savings in comparison to a methodology that establishes the feasibility of any topology only by completing the de-
sign to the very lowest level of the hierarchy. Hence, it proves highly beneficial to establish the feasibility region for each topology at each level of the hierarchy.

Ideally, we would like to develop a completely general methodology that works under all circumstances. Unfortunately, since we have no a priori knowledge about the convexness of the surface, this is not feasible. Considering this limitation, we have developed a method that works in the majority of circumstances and produces a good approximation of the feasibility surface. Since one of our primary goals is to reduce the cost of generating these macromodels, we take particular care to reduce the total number of experimental runs.

Though we use a different equation form and a different search algorithm to locate boundary points, the simplicial approximation technique [6, 7] forms the basis of our procedure and will be discussed briefly. For a given region in an n-dimensional space, $R$, and its boundary, $\partial R$, the simplicial approximation is based on approximating the boundary $\partial R$ by a polyhedron, which consists of a set of n-dimensional hyperplanes which lie inside $\partial R$ or on it. The procedure starts with determining a set of $m \geq n+1$ points on the boundary $\partial R$. Usually, $m$ is taken to be either $n+1$ or $2n$. One way to find $2n$ boundary points is to first locate a point inside $R$ and then perform one-dimensional line searches in the positive and negative directions for each coordinate. A convex polyhedron is constructed using the set of the boundary points. This polyhedron is the first approximation of $\partial R$. Next, the largest inscribed hypersphere is found in this polyhedron. Among the hyperplanes, which are tangent to the hypersphere, the largest tangential face is then found. A line search is now performed along the direction from the center of the hypersphere to the tangential point, $P_0$, on the largest face. It results in a boundary point, $P_1$. If the distance between $P_0$ and $P_1$ is small enough, the polyhedron is a good approximation of $\partial R$; otherwise, $P_1$ is added to the set of the boundary points and a new polyhedron is constructed. The above procedure is repeated until a good approximation is obtained.

Figure 3 shows the feasibility surface in two dimensions for a two-stage opamp. The simplicial approximation technique is not well suited for this surface. The primary problem being created by the coordinate axis boundary. It is not possible to find a well inscribed circle inside the polyhedron, which approximates the curve well. This is a fairly typical case because input variables usually have positive values. In addition we note that the curve is not convex.

The key aspect of any methodology is to find a sufficient set of boundary points efficiently. There are two aspects to finding these points; the direction of the search and the method used to perform the search. In simplicial approximation the search direction is generated by drawing a line from the center of the hypersphere to the tangential point on the largest face. The search is performed using a simple line search. To reduce the number of experiments necessary we perform a binary search instead of a line search. And to uncertain the direction of the search we propose two approaches: radial binary search and vertical binary search.

Figure 4 shows the radial binary search and the vertical binary search strategies. For radial binary search, the search direction starts from the origin of the coordinate system. Here, like simplicial approximation, the surface needs to be convex. Additionally, the origin needs to lie inside the surface. The second requirement ensures the existence of a solution (a boundary point) of the search. The first requirement ensures the uniqueness of the solution. As shown on the left-half of Figure 4, there exists at least one solution along any direction from the origin. Furthermore, we observe that, along line 1, there are three solutions because the curve is non-convex in that direction. The radial search strategy is similar to simplicial approximation and therefore the uniqueness of a solution is essential. Furthermore, there is an additional disadvantage to the radial search strategy from a code implementation point of view. Though, the implementation can be visualized in two or three dimensions it is extremely difficult to visualize even the definition of the spherical coordinates in higher dimensions. Most circuit designs have dimensions greater than three. Radial search, though, has the advantage that as long as the two requirements are met then the shape of the surface is irrelevant.

The vertical binary search procedure shown on the right-half of Figure 4 is not without problems either. The problem is illustrated by line 2 in this figure. Any search that is performed along this line will not result in a solution. All the experiment runs along a direction in which there is no solution are completely wasted. Since,
we have no a priori knowledge of the surface, the search procedure is performed on a regularly shaped region. This is also a requirement of the recursive procedure in the case of higher dimensions. However, it is usually the case that the boundary of the feasibility region in the hyperplane perpendicular to the “vertical” direction is irregular. Therefore, we face the problem of either exploring an incomplete surface or performing the search outside the boundary. However, the advantage of the vertical search procedure is that the requirement on convexness is relaxed. The surface need only be convex in at least one dimension.

For our results we use the vertical binary search procedure. We find that it is not unusual that a feasibility surface is non-convex. However, our experience suggests that most feasibility surfaces are convex in at least one direction. Hence, it still allows for a general methodology. Additionally, the vertical search procedure is better suited for high-dimensional situations. We also note that since we use radial basis functions to represent the feasibility surface, it makes it easier to use Cartesian coordinates throughout our entire procedure.

Regression analysis: Having once performed the necessary experimental runs and gathered the data points, we now use these data points to calculate the coefficients of the macromodel. Additionally, the accuracy of the resultant macromodel also needs to be verified. To dynamically collect the data points, we perform the regression analysis at two levels. A local regression analysis procedure is called recursively for a local area until a certain accuracy is obtained. After all the data points have been obtained, a global regression analysis is then performed to obtain an approximation for the entire surface. The adaptive volume slicing technique was developed in conjunction with local regression analysis to dynamically generate the necessary experiments. Radial basis functions are used for macromodeling and is particularly well suited to our methodology because of its locality and linear-in-parameter structure [4].

3 Results and Conclusions

Figure 5 shows the feasibility region for a two-stage opamp using a 2.0μM MOSIS process. And Figure 6 shows the feasibility region for a single-stage OTA opamp using a 1.2μM MOSIS process. Note the substantial difference in the shape of the two graphs. The change of design style is largely responsible for this difference. However, the larger achievable slew-rate and lower gain in the case of the OTA is mostly attributable to the change in the process. Though not explicitly shown here, the validity of our approach has been tested at different hierarchical levels, current-mirror, opamp, and A/D converter [4]. We have found that our methodology works well for analog circuit blocks. Though not explicitly tested, our approach should be directly applicable to digital circuit blocks as well as no knowledge of the underlying circuit is assumed, making the methodology particularly suitable for digital/analog mixed-signal design.

Acknowledgment

This research was supported in part by a grant from NSF under grant number MIP-9110719.

References