Model Reduction for Design and Optimization of Analog Circuits

Brad Bond, Zohaib Mahmood, Alex Megretski, Luca Daniel

1 Overview

Given an analog circuit, a nonlinear system of equations that describes the circuit's dynamics can be constructed by combining conservation laws (e.g. Kirchhoff's current law) with device constitutive relations (e.g. $C\frac{dv}{dt} = i$ for a capacitor). The result is a nonlinear dynamical system of the form

$$\dot{x} = f(x) + bu \qquad\qquad y = C^T x, \tag{1}$$

where x is the state vector (containing, for example, the node voltages), u is a vector of inputs, and y is a vector of outputs. In general, system (1) is computationally expensive to simulate due to the large number of highly nonlinear equations. Our goal is to use model reduction techniques to automatically extract compact models of analog systems that are cheap to simulate and can be used in an optimization process.

Aspects of the analog circuit model reduction problem that make it extremely challenging are the following: systems contain **highly nonlinear functions** resulting from the presence of nonlinear devices such as transistors; parameter dependence must be preserved for a **large number of parameters**, including both device parameters and system geometry; systems are typically comprised of **highly coupled subsystems**, resulting mainly from parasitic coupling between elements such as RF inductors and the chip substrate; and systems may contain a **large number of input and output ports**.

2 Projection Approaches

The standard projection approach is to approximate the state $x \in \mathbb{R}^N$ in a low-dimensional space such that $x \approx Vz$, where $z \in \mathbb{R}^q$ and $q << N$. This approximation combined with a left-projection matrix U constructed to reduce the number of equations results in the reduced model

$$\hat{E}\dot{z} = U^T f(Vz) + \hat{b}u \qquad\qquad y = \hat{C}^T z, \tag{2}$$

where $\hat{E} = U^T V$, $\hat{b} = U^T b$, and $\hat{C} = V^T C$. However, this is not a 'reduced' model in the typical sense, because although the reduced nonlinear function $U^T f(Vz)$ contains only q equations, it still requires $O(N)$ computations to evaluate, meaning simulation of this model will still be extremely computationally expensive. Thus, in addition to the traditional issues associated with linear model reduction, such as subspace selection and stability preservation, we also must ensure that the reduced model is 'cheap' to simulate.

2.1 Function Approximation

There are two common approaches for reducing the complexity of the reduced nonlinear function: approximating the original function $f(x)$ with something that is 'projectable', and approximating the reduced function $\hat{f}(z)$ with something that is cheap to evaluate

$$f(x) \approx \tilde{f}(x) \qquad\qquad U^T f(Vz) \approx \hat{f}(z). \tag{3}$$

Here $\tilde{f}(x)$ is a 'projectable' function, meaning $U^T\tilde{f}(Vz)$ is cheap to evaluate, and $\hat{f}(z)$ is a 'reduced' function, meaning the cost of evaluating $\hat{f}(z)$ is cheap (e.g. $O(q)$ or $O(q^2)$). Two common choices for $\tilde{f}(x)$ are polynomial expansions of $f(x)$ and piece-wise linear (PWL) approximations of $f(x)$. The PWL function and its projection have the form

$$\tilde{f}(x) = \sum_i w_i(x)[A_i x + k_i] \qquad\qquad U^T\tilde{f}(Vz) = \sum_i w_i(z)[\hat{A}_i z + \hat{k}_i]. \tag{4}$$

Here $A_i = \frac{\partial f(x)}{\partial x}$ evaluated at x_i, $k_i = f(x_i) - A_i x_i$, $\hat{A}_i = U^T A_i V$, $\hat{k}_i = U^T k_i$, and w_i are interpolating weighting functions. The complexity of function evaluations is now $O(q^2)$. However, this introduces new questions regarding how to optimally (and efficiently) construct the PWL approximation (e.g. how to select the linearization points x_i).

2.2 Subspace Selection

The second challenge associated with nonlinear model reduction involves constructing the projection matrices U and V. It is possible to obtain projection vectors by applying linear approaches (such as moment matching or balanced truncation) to linearizations of the nonlinear system, but it's not clear that this is the best approach. Another common choice for constructing projection vectors is an SVD-based approach (e.g. POD, PCA, ...), where the projection vectors are obtained from a singular value decomposition on data (a collection of state vectors) obtained from simulation, but this also introduces the question of how to optimally generate this training data.

2.3 Stability

One other key challenge associated with projection-based model reduction for nonlinear systems is guaranteeing stability of the reduced model (meaning that typical bounded inputs do not produce unbounded outputs). This often must be explicitly enforced through the choice of projection matrices U and V, as well as through the choice of the approximation to the reduced function $U^T f(Vz)$.

3 System Identification Approaches

The projection approaches described above require knowledge of the internal dynamics of the system, i.e. the system equations described in (1), but these equations are not always known. For example, we may have access only to physical measurements from a real circuit, or to data generated by some other circuit simulation software. In this case, instead of applying reduction techniques to system (1), we look to identify a compact nonlinear model based on a set of given input-output data pairs $(\tilde{u}, \tilde{y})$. In the most general form, we wish to find a model

$$F(y, \dot{y}, u, \dot{u}) = 0 \tag{5}$$

that accurately produces training outputs $\tilde{y}$ from training inputs $\tilde{u}$, i.e $F(\tilde{y}, \dot{\tilde{y}}, \tilde{u}, \dot{\tilde{u}}) \approx 0$, and is computationally cheap to simulate when solving for $y(t)$. Note that there is no relationship between the nonlinear functions f in (1) and F in (5).

One choice of nonlinear functions that allows us to capture highly nonlinear behavior in a model that is cheap to evaluate without significantly increasing the cost of identification is rational functions. For example, we may define the system as follows to obtain an explicit nonlinear model

$$F = \dot{y}q(y, u, \dot{u}) - p(y, u, \dot{u}) = 0 \qquad \longrightarrow \qquad \dot{y} = \frac{p(y, u, \dot{u})}{q(y, u, \dot{u})}. \tag{6}$$

Here we can capture highly nonlinear behavior even when p and q are simple nonlinear functions, such as polynomials or splines.

From this identification problem we wish to obtain a model that is provably stable, and also has an error bound that will quantify how accurately the model will reproduce training outputs from training inputs. Additionally, we again face the problem of how to efficiently generate a set of optimal training data such that the resulting model is both accurate and robust. That is, the model should be trained such that inputs of interest 'similar' to training inputs, but not exactly contained in the training set, do not drive the model to areas where it will produce inaccurate outputs.