A Formal Approach to Nonlinear Analog Circuit Verification

Lars Hedrich Erich Barke

Institute of Microelectronic Systems
Department of Electrical Engineering
University of Hanover, D-30167 Hanover, Germany
E-mail: hedrich@ims.uni-hannover.de

Abstract

This contribution presents an approach to nonlinear dynamic analog circuit verification. The input-output behavior of two systems is analyzed to check whether they are functionally similar. The algorithm compares the implicit nonlinear state space descriptions of the two systems on the same or on different levels of abstraction by sampling the state spaces and by building a nonlinear one-to-one mapping of the state spaces. Some examples demonstrate the feasibility of our approach.

I. Introduction

Formal verification is an attractive alternative to simulation. Up to now it is only discussed for digital circuits [3]. In this contribution we propose a formal approach to verification of nonlinear dynamic analog integrated circuits.

In contrast to digital logic analog circuits are continuous in signal values and time. Moreover, it is not useful to check the identity of the circuits, because the parameters of the circuits are continuous and even a very small deviation in one parameter may lead to a negative verification result. Due to this fact formal verification of analog circuits has to be defined in its own fashion:

> Formal verification of analog circuits checks whether two circuits are functionally similar with respect to their input-output behavior. Verification is valid for all input signals.

In this paper we propose an algorithm for formal verification of nonlinear dynamic analog circuits. The circuit representation is given as a system of nonlinear differential equations. Therefore, it is possible to compare circuits on the same level of abstraction as well as on different levels, e.g. SPICE netlists versus analog behavioral models. Verification is done by iterative comparison of implicit state space descriptions. It indicates whether two systems are functionally similar or not. An explicit error measure is given.

In Section II the basic concepts of our approach are described. Section III deals with details of the algorithms. Examples are presented in Section IV, followed by a comparison with transient analysis in Section V. Finally, we conclude in Section VI.

II. Basic concepts

A. State space description

A large class of nonlinear dynamic single input single output (SISO) circuits can be described by a set of implicit nonlinear time-invariant first order differential equations:

$$\begin{aligned} \phi(\mathbf{x}(t),\dot{\mathbf{x}}(t),u(t)) &= \mathbf{0} \\ \gamma(\mathbf{x}(t),y(t),u(t)) &= 0 \end{aligned} \qquad (1)$$

where $\mathbf{x}(t) = \left[x_1(t),...,x_p(t)\right]^T$ is the state vector (a vector is denoted by bold printing) and u(t) and y(t) describe input and output signals, respectively.

The nonlinear functions ϕ,γ can be implicitly represented by a vector field $\dot{\mathbf{x}}$ and a scalar field y in the state space which is extended by the input u. In the following, we will name the state space which is extended by the input extended state space.

We consider two systems in an implicit nonlinear state space description of type (1). The objective of our approach to verification is to determine that the vector fields $\dot{\mathbf{x}}_A(\mathbf{x}_A,u)$, $\dot{\mathbf{x}}_B(\mathbf{x}_B,u)$ and the scalar fields $y_A(\mathbf{x}_A,u)$, $y_B(\mathbf{x}_B,u)$ representing the functions ϕ_A,γ_A and ϕ_B,γ_B of the systems A and B are similar. Note that the vector and scalar fields can numerically be found by evaluating the implicit functions ϕ,γ at discrete points. If two systems have the same internal state variables, a straightforward comparison of their vector fields $\dot{\mathbf{x}}_A, \dot{\mathbf{x}}_B$ and their scalar fields y_A, y_B is sufficient to determine that the systems are equal.

B. Nonlinear mapping of state vectors

In general, two systems do not have the same internal state variables because they represent different implementations on maybe different levels of abstraction. In this case, the method described above is not able to identify systems with similar input-output behavior. We consider the following two systems:

$\phi_A: \dot{x}_A(t)+3(x_A(t)-x_A^{2/3}(t)\cdot u(t))=0$ $\quad \phi_B: \dot{x}_B(t)+x_B(t)-u(t)=0$

$\gamma_A: y_A(t)-x_A(t)=0$ $\quad \gamma_B: y_B(t)-x_B^3(t)=0$

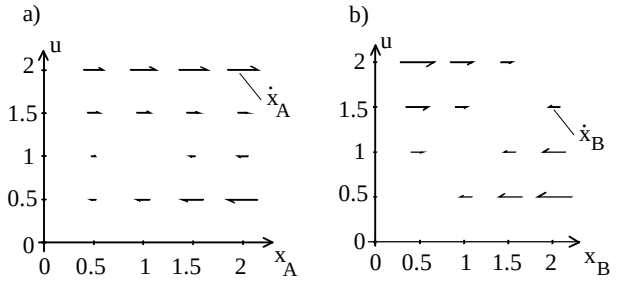

Figure 1: Systems with different state encoding

The two systems A and B are equal with respect to their input-output behavior: ϕ_B, γ_B can be derived from ϕ_A, γ_A using $x_A=x_B^3$. However, the differential equations and the vector and scalar fields are different (see Figure 1). Therefore, a one-to-one mapping $\mathbf{x}'_A=t(\mathbf{x}_A)$ has to be found which uniquely maps the state vector $\mathbf{x}_A$ upon the state vector $\mathbf{x}'_A$. After this mapping the vector field $\dot{\mathbf{x}}'_A$ in the extended state space of system A can be compared to the vector field $\dot{\mathbf{x}}_B$. For this example using $\mathbf{x}'_A=\mathbf{x}_A^{1/3}$ leads to two identical vector fields $\dot{\mathbf{x}}'_A$ and $\dot{\mathbf{x}}_B$ like that in Figure 1b. The same result can be obtained for the scalar fields y_A and y_B.

III. Algorithm

A. Sampling the state space

The vector and scalar fields in the extended state space of the two systems are compared by sampling the extended state space and comparing the fields at every discrete sampling point (see similar sampling method in system identification [1]). The boundaries of the extended state space are determined by the maximum excitation of the state variables and the input. This leads to a finite set of points at which a comparison has to be carried out. The basic verification algorithm reads as follows:

Verification of Nonlinear Dynamic Systems

for every input value in predefined range

begin

do DC analysis in order to get an initial state vector

for every sample point in the state space in predefined ranges

begin

compute linear mapping matrices $\mathbf{T}_A$ and $\mathbf{T}_B$ (procedure explained in Section III B.)

calculate $f_{\dot{\mathbf{x}}}$ (error of the state derivatives vector fields) and

f_y (error of the output scalar field)

adjust x_A (procedure explained in Section III C.)

calculate new state vectors for next sample point (procedure explained below)

end

end

if $f_{\dot{\mathbf{x}}}$, f_y for all sample points < error margin

then systems are functionally similar

Two loops are necessary to sample the extended state space. In the outer loop a DC analysis is required because the operating point is used as an initial state vector. The inner loop starts with the computation of linear mapping matrices $\mathbf{T}_A$ and $\mathbf{T}_B$ for the actual sample point using a linearized state space description, because only for linear systems a canonical representation and a corresponding mapping can be found (see Section III B.). The mapping matrices $\mathbf{T}_A$ and $\mathbf{T}_B$ are used to calculate a state vector for the next sample point by adding a finite distance $\Delta\mathbf{x}'$ multiplied by $\mathbf{T}^{-1}$ to the actual sample point:

$$\mathbf{x}_{A\,new}=\mathbf{T}_A^{-1}\cdot\Delta\boldsymbol{x}'+\mathbf{x}_A,\ \mathbf{x}_{B new}=\mathbf{T}_B^{-1}\cdot\Delta\boldsymbol{x}'+\mathbf{x}_B. \quad (2)$$

Note that we use italics to denote linearized values valid only for the actual sample point. In Figure 2 a computation of $\mathbf{x}_{A\,new}$ for a system having only one state variable is shown.

By iterative evaluation of the equation (2) the complete nonlinear discrete mappings

$$\mathbf{x}'_A=t_A(\mathbf{x}_A), \qquad \mathbf{x}'_B=t_B(\mathbf{x}_B)$$

are constructed point by point while stepping through the state space. This leads to a mapping which has a little deviation from the unknown exact mapping function shown in Figure 2 by a solid line. Therefore, the sample point x_A is adjusted after the error is calculated (described in Section III C.).

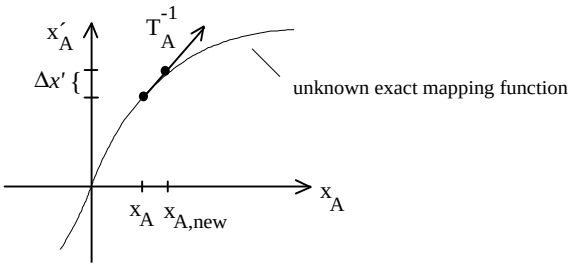

Figure 2: Computation of $\mathbf{x}_{A\text{new}}$ and generation of the mapping of $\mathbf{x}'_A = t_A(\mathbf{x}_A)$

To answer the basic question, whether the two systems are functionally similar at each point the error evaluation is done. The errors are defined as follows:

$$f_{\dot{\mathbf{x}}} = \left\| \mathbf{T}_A \cdot \dot{\mathbf{x}}_A - \mathbf{T}_B \cdot \dot{\mathbf{x}}_B \right\|, \quad f_y = \left\| y_A - y_B \right\|$$

The first error corresponds to the deviation of ϕ_A and ϕ_B, respectively, the second one to γ_A and γ_B at the actual sample point. An overall error can be found by calculating a mean value over all sample points.

B. Obtaining the state vector transformation

It is impossible to calculate the nonlinear mapping $\mathbf{x}'_A = t(\mathbf{x}_A)$ and $\mathbf{x}'_B = t(\mathbf{x}_B)$ directly from the nonlinear system functions ϕ_A and ϕ_B. However, a mapping in a particular sample point can be calculated by linearizing the systems. For a linear system in a state space description a canonical form can be found, e.g. the Jordan form of the system matrix **A**. To obtain a unique mapping of each state variable from one system to the other the Jordan form has to be sorted according to the magnitude of the eigenvalues. Here, only unique eigenvalues are permitted leading to sorted diagonal system matrices.

A transformation matrix **T** can be calculated which transforms the system matrices and the state vectors according to:

$$\mathbf{A}' \cdot \mathbf{T} = \mathbf{T} \cdot \mathbf{A}, \qquad \boldsymbol{x}' = \mathbf{T} \cdot \boldsymbol{x}$$

A sorted Jordan matrix $\mathbf{A}'$ and its corresponding transformation matrix **T** are set up for each system. Different scales of the state vectors are eliminated by the following relation:

$$\mathbf{T}_A \cdot \mathbf{b}_A \stackrel{!}{=} \mathbf{T}_B \cdot \mathbf{b}_B$$

If the resulting systems have different order the n largest negative eigenvalues are skipped which ensures, that the comparison is still possible [2]. These skipped eigenvalues should have no influence on the input-output behavior.

The corresponding algorithm reads as follows:

Determine Linear Mapping Matrices $\mathbf{T}_A$ and $\mathbf{T}_B$
linearize systems A and B at the sample point
calculate the state space descriptions:

$$\dot{\boldsymbol{x}}_A(t) = \mathbf{A}_A \cdot \boldsymbol{x}_A(t) + \mathbf{b}_A \cdot u(t) \quad \dot{\boldsymbol{x}}_B(t) = \mathbf{A}_B \cdot \boldsymbol{x}_B(t) + \mathbf{b}_B \cdot u(t)$$

$$y_A(t) = \mathbf{c}_A^T \cdot \boldsymbol{x}_A(t) + d_A \cdot u(t) \quad y_B(t) = \mathbf{c}_B^T \cdot \boldsymbol{x}_B(t) + d_B \cdot u(t)$$

calculate the sorted Jordan-form: $\mathbf{A}'_A, \mathbf{b}'_A, \mathbf{c}'^T_A, d'_A$;
$\mathbf{A}'_B, \mathbf{b}'_B, \mathbf{c}'^T_B, d'_B$ and the corresponding transformation matrices: $\mathbf{T}_A$ and $\mathbf{T}_B$
if the number of eigenvalues is not the same
then reduce the system having more eigenvalues by skipping the n largest negative eigenvalues
scale transformation matrices according to
$\mathbf{T}_A \cdot \mathbf{b}_A \stackrel{!}{=} \mathbf{T}_B \cdot \mathbf{b}_B$

C. Adjusting the operating point

During the iterative construction of the nonlinear mapping $\mathbf{x}'_A = t_A(\mathbf{x}_A)$ and $\mathbf{x}'_B = t_B(\mathbf{x}_B)$ a small error in the calculated sample points $\mathbf{x}_A$ and $\mathbf{x}_B$ adds up to a large error (see the dots in Figure 3).

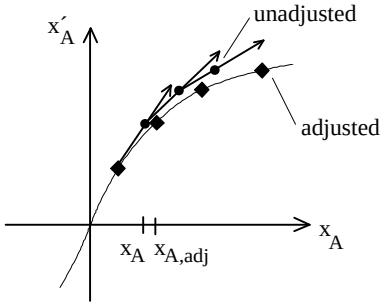

Figure 3: Unadjusted (•) and adjusted (♦) computation of the mapping $\mathbf{x}'_A = t_A(\mathbf{x}_A)$ for a system having only one state variable

An iterative adjustment algorithm avoids this problem using the condition $f_{\dot{\mathbf{x}}} = \left\| \mathbf{T}_A \cdot \dot{\mathbf{x}}_A - \mathbf{T}_B \cdot \dot{\mathbf{x}}_B \right\| \stackrel{!}{=} 0$ to adjust $\mathbf{x}_A$ by an modified Quasi-Newton optimization method [7]. The condition holds, because, assuming that the functions ϕ_A and ϕ_B are equal in the sense of the above definition, a difference between the vectors $\dot{\mathbf{x}}'_A$ and $\dot{\mathbf{x}}'_B$ indicates a difference between the corresponding state vectors $\mathbf{x}_A$ and $\mathbf{x}_B$. The adjusted operating point $\mathbf{x}_{A,\text{adj}}$ meets the exact mapping function marked in Figure 3 by diamonds.

IV. Examples

The algorithm is implemented using the symbolic math package Maple V™ [6]. A netlist translator from SPICE netlists is applied to handle arbitrary analog circuits on transistor level. Additionally, a hardware description language using the syntax of Maple V™ has been developed in order to provide a comparison of behavioral descriptions. Due to the prototype implementation and the exclusive use of Maple V™ the CPU times of the algorithm are still high. A future more efficient implentation will reduce them drastically.

As a first example we consider the two nonlinear differential equations:

$$\phi_A: \dot{x}_A(t) + x_A(t) - u(t) = 0 \qquad \phi_B: \dot{x}_B(t) - (\operatorname{arctanh}(x_B(t)) - u(t)) \cdot (1 - x_B^2(t)) = 0$$

$$\gamma_A: y_A(t) - \tanh(x_A(t)) = 0 \qquad \gamma_B: y_B(t) - x_B(t) = 0$$

Using the substitution $x_A = \operatorname{arctanh}(x_B)$ the equality of these systems can be shown. In Figure 4 the relative errors $f_{\dot{x}}$ and f_y obtained by the verification procedure are shown in the extended state space of system A. The stepsize between the sample points in x and u direction is 0.01 and 0.4, respectively, the boundaries of the extended state space are predefined by $-2 < x < 2$, $0 < u < 2$. The sample points are connected by a grid. With this stepsize the algorithm achieves relative errors below 2 %, further reduced stepsize leads to smaller errors.

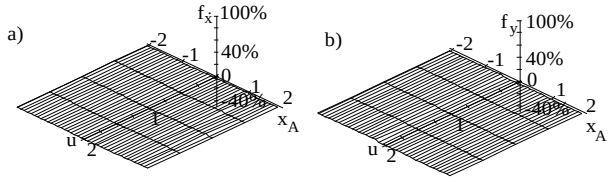

Figure 4: Relative errors a) $f_{\dot{x}}$ and b) f_y of the differential equations

The second example deals with a CMOS inverter circuit shown in Figure 5.

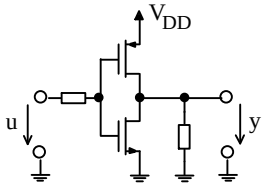

Figure 5: CMOS inverter schematic

In Table 1 the verification results of two equal inverter circuits and two further ones having different parameters are presented. The results show that a difference in the dynamic behavior, e.g. different gate capacities, leads to an error in the state vector derivatives $\bar{f}_{\dot{x}}$ as well as to an error in the output values $\bar{f}_y$.

Circuit description	Relative errors			
	Mean values		Maximum values	
	$\bar{f}_{\dot{x}}$	$\bar{f}_y$	$\hat{f}_{\dot{x}}$	$\hat{f}_y$
Equal	0 %	0 %	0 %	0%
Different gate capacity	18 %	72 %	66 %	200 %
Diff. thresh. voltage	0 %	44 %	0 %	200 %

Table 1: Relative errors of the verification of CMOS inverter circuits

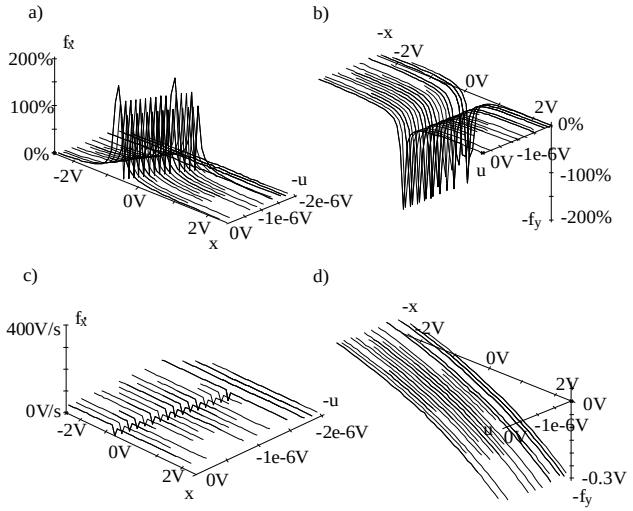

Figure 6: Relative errors a) $f_{\dot{x}}$, b) f_y and absolute errors c) $f_{\dot{x}}$, d) f_y for the verification of the operational amplifiers

Finally, a comparison of a CMOS operational amplifier consisting of eight MOS transistors versus a behavioral description is presented. The behavioral description models slew rate, gain bandwidth product and limitation to the supply voltages. In Figure 6 the relative errors $f_{\dot{x}}$, f_y are shown in the extended state space. The verification is performed in the ranges $-2.5V < x < 2.5V$, $0V < u < 2.0 \cdot 10^{-6}V$.

The large relative error of $f_{\dot{x}}$ in the small area near the DC solution results from the small values of $\dot{x}$ near the dynamic equilibrium (see Figure 6a). In the remaining area the relative error of $f_{\dot{x}}$ is near 0% indicating a very good correspondence in dynamic behavior. The absolute error of $f_{\dot{x}}$ is constant at 40 V/s resulting from a slight difference in pole locations (see Figure 6c). A large relative error of f_y appears near the point $x = 0V$ belonging to an output voltage of $y = 0V$ (see

Figure 6b). It is caused by a small absolute error of f_y near y = 0V indicating a difference in the nonlinear DC transfer curve (see Figure 6d).

Taking these facts into account and considering the mean values of the relative errors: $\overline{f_{\dot{x}}}$ = 16.4% $\overline{f_y}$ = 19.7%, the verification process shows that the operational amplifiers are identical with little deviations in DC behavior and pole locations.

V. Comparison to transient analysis

We consider a transient analysis of a system with one state variable, e.g. a nonlinear R-C circuit.

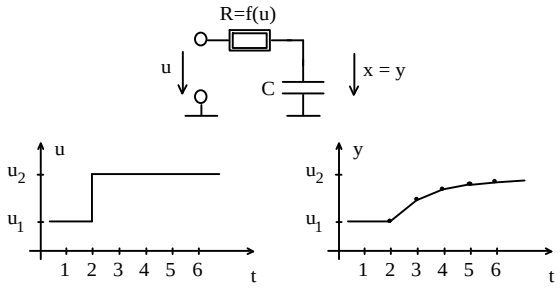

Figure 7: Nonlinear R-C circuit with input and output signal calculated by a transient analysis

The circuit with an input excitation and the corresponding output are shown in Figure 7. Calculated points are marked by dots. The extended state space in Figure 8a is described by the input voltage u and by the capacitor's voltage x which is equal to the output voltage y. The DC solution for varying u is shown by a solid line. The calculated points corresponding to the transient analysis in the extended state space are given in Figure 8a.

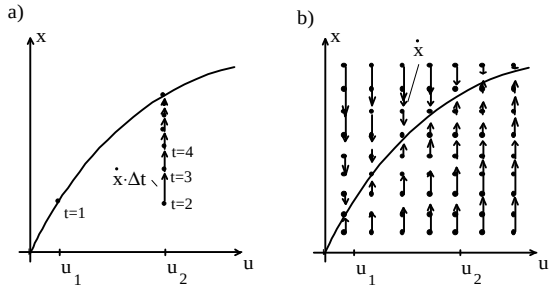

Figure 8: Sampled points in a) transient analysis and b) verification.

Note that by a simulation only a few points of the extended state space are calculated. In order to visit all points of the state space many simulations with different input excitations are necessary. Since all simulations return to DC solution and therefore evaluate the same regions of the state space many points are evaluated several times leading to much wasted simulation time.

In contrast to simulation our approach visits each point only once (see Figure 8b). Additionally, the plane is completely covered. The drastically reduced number of visited points implies that our approach is faster than a transient analysis. The complete coverage leads to more reliable results compared to transient analysis.

VI. Conclusion

In this contribution, a formal approach to nonlinear analog circuit verification has been presented. It is based on comparing the implicit nonlinear state space descriptions of two systems with the aid of a nonlinear one-to-one mapping between the state spaces of the systems. Examples from simple equations up to a verification of a CMOS operational amplifier versus a behavioral model demonstrate the feasibility of the method.

Acknowledgement

The authors would like to thank Robert Bosch Corporation for having stimulating discussions and for providing real world examples.

References

[1] F. H. Bursal, B. H. Tongue, "A New Method of Nonlinear System Identification Using Interpolated Cell Mapping"; Proc. of the American Control Conf., Vol 4, pp. 3160-4, 1992

[2] L. Fortuna, G. Nunnari and A. Gallo, "Model Order Reduction Techniques with Applications in Electrical Engineering", Springer-Verlag London, 1992

[3] M. C. McFarland, "Formal Verification of Sequential Hardware: A Tutorial", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 12, No. 5, May 1993

[4] W. Mathis, "Geometric Theory of Nonlinear Dynamical Networks"; Lecture Notes in Computer Science, Vol. 585, Editors: F. Pichler, R. Moreno Diaz, Springer Verlag, Berlin, 1991

[5] W. Mathis, "Theorie nichtlinearer Netzwerke" (in german); Springer Verlag, Berlin, 1987,

[6] Maple V™, Waterloo Maple Software, University of Waterloo, Version 5.2, 1994

[7] P.E. Gill, W. Murray, M.H. Wright, "Practical Optimization", Academic Press Inc., London, 1981