Functional-Level Analog Macromodeling with Piecewise Linear Signals

Jerzy Dąbrowski
Institute of Electronics
Silesian Technical University
44-101 GLIWICE, Poland
fax: 48-32-372225, E-mail: jdab@boss.iele.polsl.gliwice.pl

Abstract

In this paper an analog network macromodeling technique is presented. It is oriented towards a functional-level simulation of both analog and analog-digital networks. The signals are assumed to be piecewise linear (PWL) waveforms. A class of nonlinear inertial building blocks is introduced for modeling. The proposed macromodels are accurate in sense of a timing behavior and computationally efficient, since an explicit algorithm to obtain the waveforms is used. It is based on a PWL approximation of original smooth timing responses. Practical macromodels of particular functional units are discussed. A functional-level macromodel of analog amplifier is derived in detail and its performance is shown based on SPICE estimates.

1. Introduction

The actual design strategy for complex electronic networks is based on synthesis, analysis and verification performed together at different levels of abstraction [3,4,5]. Modeling and simulation play here important role. An effort has been made to develop efficient techniques of modeling/macromodeling and simulation specific to particular level [3,4,5,6]. Usually simulation accuracy has to be trade for efficiency, reducing thereby the required CPU time. Because of the recent progress in technology of analog and mixed-signal application-specific ICs (ASICs) the need of functional-level analog simulation begins to emerge. Also an attempt of formulating the AHDL standard influences the work towards the efficient high-level macrosimulation tools [2].

In this paper the analog functional-level macromodeling and simulation are addressed. Timing specifications of analog units are incorporated into the models and an efficient explicit algorithm to solve for the timing waveforms is proposed. Piecewise linear (PWL) approximation both for the waveforms and DC characteristics is used [1] and an improved version of the PWL algorithm [11], controlling the range of approximation is given. All the units are assumed to be unidirectional, so that floating elements, like transmission gates or coupling capacitors cannot be represented separately. However, some loading effects can be accounted for. The amplifier macromodel is derived in detail as an example. The proposed macromodeling technique is well suited to the simulation of A/D networks with PWL signals, following at this point the approach presented in [1], where only rather rough analog macromodels have been used. It has been verified by means of a prototype PWL simulator for various A/D networks.

2. Macromodeling technique

The macromodeling technique proposed here is based on a class of first order inertial building blocks. It is to be pointed out that in most cases the analog units consist of subcircuits that exhibit inertial properties. Therefore, any macromodel of an analog unit can usually consist of a few building blocks to mimic the timing behavior, the basic

0-8186-7156-4/95 $4.00

nonlinearities (e.g. saturation) and the output loading effects. The constitutive relation of the basic building block takes the form of:

$$T\frac{dx}{dt} + x = f(x_{inp}) \tag{1}$$

where f(o) denotes its static (DC) characteristic and T its time constant. The eqn.(1) can be generalized into a multiple input case, e.g. for a multiplier or adder. To obtain a pure static block its dynamics has to be removed by letting in eqn.(1): T=0. On the other hand, when capacitive loading effects at the output x must be accounted for, the time constant takes the form of:

$$T = R_s(C_o + C_s) \tag{2}$$

where C_o is charging capacitance at the output and C_s, R_s are respectively the internal output capacitance and resistance. Using a PWL signal $x_{inp}(t)$ and a PWL approximation for f(o), also a PWL signal $u(t)=f[x_{inp}(t)]$ is obtained. Let $x(0)=x_0$ and $u(t)=u_0 + rt$, $t\in[0,t_{max}]$. Hence, solving for (1) we have:

$$x(t) = (x_0 - u_0 + rT)e^{\frac{-t}{T}} + r(t-T) + u_0 \tag{3}$$

Our aim is to get a PWL approximation of (3) to enable further propagation of the signal x in a linearized form. For this purpose first we split the time interval $[0,t_{max}]$ into subintervals $[0,t_1],[t_1,t_2], \ldots [t_n,t_{max}]$. For each subinterval $[t_i,t_{i+1}]$ a segment of PWL approximating signal x_{lin} is defined by its end points, that are assumed to lie on the curve x. Hence we have: $x_{lin}(t_i)=x(t_i)$ and $x_{lin}(t_{i+1})=x(t_{i+1})$. In fact, given t_i, the t_{i+1} has to be found. To control the accuracy of this approximation, the Chebyshev measure has been found the most advantageous. Consequently our objective can be formulated as an optimization task, that is to maximize the distance $d = t_{i+1}-t_i$ with some constraints and given t_i:

$$\underset{t_{i+1}}{Maximize}\ d:\ \{\ d = t_{i+1} - t_i,\ \ t_{i+1} \le t_{max}\} \tag{4a}$$

$$p(t_i, t_{i+1}) = \underset{t\in(t_i,\, t_{i+1})}{Max} |x(t) - x_{lin}(t)| \tag{4b}$$

$$p(t_i, t_{i+1}) \le p_{max} \tag{4c}$$

where $p(t_i,t_{i+1})$ is the performance index and p_{max} is an arbitrarily chosen constant (maximum allowed approximation error). During simulation it is important to calculate the subsequent points $(t_{i+1},x(t_{i+1}))$ in an efficient way and possibly to avoid iterations, that all typical optimization procedures (e.g. Fibonacci search [7]) are based on. Hence, we propose an explicit algorithm to solve this problem.

First, to avoid the transcendental equation we start with a quadratic approximation for x. Invoking the Taylor series expansion at $t_i=0$ we obtain:

$$x_q(t) = \frac{x_0 - u_0 + rT}{2}\left(\frac{t}{T}\right)^2 + (u_0 - x_0)\frac{t}{T} + x_0 \tag{5}$$

The truncation error introduced by omitting in (5) the third order Lagrange rest is:

$$O(\tau^3) = \frac{u_0 - x_0 - rT}{6}\left(\frac{\tau}{T}\right)^3, \quad \tau \in (0,\ t) \tag{6a}$$

Eqn.(6a) can be reformulated to control the range of the quadratic approximation (5). The maximum allowed time t_a for x_q to hold can be found from:

$$t_a = T\sqrt[3]{\frac{6\epsilon_a}{|x_0 - u_0 + rT|}} \tag{6b}$$

where ϵ_a is an estimated value of the maximum allowed truncation error. Consequently the constraint condition given by (4a) has to be slightly modified by putting $t_{i+1}\in(t_i,t_i+t_a]$. In Fig.1 the signals x,x_q and u are shown. In Fig.2 the distance function $\Delta=|x_{lin}-x_q|$ for the quadratic signal x_q is presented. The x_{lin} is defined by the pair of points $(0,x_q(0))$ and $(t_a,x_q(t_a))$. Similarly, the distance function for the original x signal can be formulated by $\Delta_x=|x_{lin}-x|$. Since the Δ function is strictly quadratic, $\Delta(t_a/2)=\Delta_{max}$ and from simple calculations we have:

$$\Delta_{max} = \frac{|x_0 - u_0 + rT|}{8}\left(\frac{t_a}{T}\right)^2 \tag{7}$$

Assuming Δ_{max} to be an estimate of $p(0,t_a)$ we check for the constraint (4c). Suppose, $\Delta_{max} > p_{max}$, so that this constraint is violated and a new value of $t_{i+1}=t_1$ must be found (see Fig.2). The following condition has to be satisfied: $p(0,t_1)=p_{max}$. Again, from simple calculations we obtain:

$$\frac{\Delta_{max1}}{\Delta_{max}} = \left(\frac{t_1}{t_a}\right)^2 \tag{8}$$

Since x is almost quadratic for $t\le t_a$, we have also:

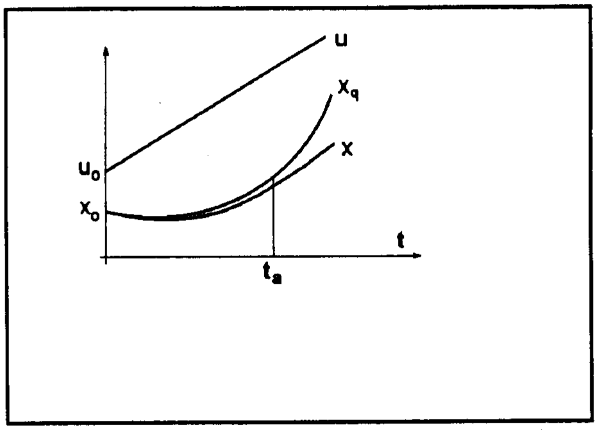

Figure 1. Model time response x versus its quadratic approximation x_q.

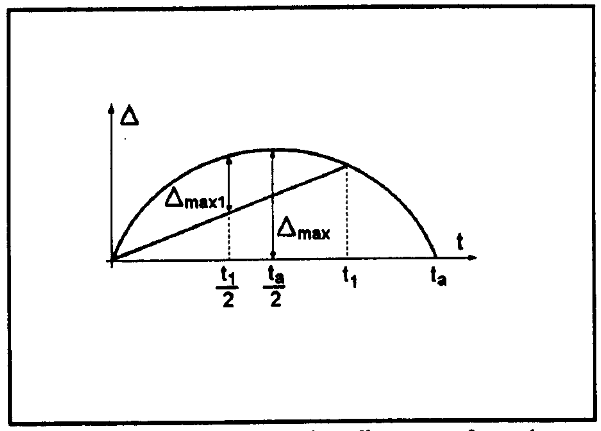

Figure 2. Method of the distance function calculation.

$$\frac{p(0,t_1)}{p(0,t_a)} \approx \left(\frac{t_1}{t_a}\right)^2 \qquad (9)$$

In practice, when the constraint (4c) is violated, t_1 can be computed based on (9) from:

$$t_1 = t_a \sqrt{\frac{p_{max}}{p(0,t_a)}} \qquad (9a)$$

This simple formula is fundamental for an efficient solution of the approximation task (4a,b,c). Hence, a preliminary algorithm for the PWL approximation of x, expressed by (3) is as follows:

repeat
  compute t_a
  if $t_a \le t_{max}$ *then* $t_w \leftarrow t_a$ *else* $t_w \leftarrow t_{max}$
  compute parameters for $\Delta_x = |x - x_{lin}|$
  $p_i \leftarrow \Delta_x(t_w/2)$
  if $p_i > p_{max}$ *then* $t_w \leftarrow t_w * sqrt(p_{max}/p_i)$
  $t_{i+1} \leftarrow t_i + t_w$
  $x_{i+1} \leftarrow x(t_w)$
  $i \leftarrow i+1$
  $t_{max} \leftarrow t_{max} - t_w$
  if $t_{max} > 0$ *then* $x(t) \leftarrow x(t+t_w)$
until $t_{max} > 0$

Observe, that for each linear segment only two computations of the exponential function (present in formula (3)) are required, for $x(t_w)$ and for $x(t_w/2)$. After a segment calculation is completed, the time shift $t \leftarrow t+t_w$ for the x signal is done and new data to compute the next t_a and Δ_x are available.

In Fig.3 an example of the obtained piecewise linear approximation for x is shown. The accuracy controlled by $p_{max}=0.08$ has been chosen. The excitation u consists of two segments.

More detailed analysis of eqns. (3) and (9a) makes an improvement of the presented computation algorithm feasible. It can be shown that the denominator $|x_0-u_0+rT|$ in (6b) is reduced in the next linear segment of x_{lin} by a ratio $\exp(-t_w/T)$, where t_w is the length of the actual time subinterval. Hence, for two subsequent segments of x_{lin} we have:

$$t_a(i+1) = t_a(i)\, e^{\frac{t_w}{3T}} \qquad (10)$$

Clearly, the same holds for the numerator in eqn.(7) and we obtain:

$$\Delta_{max}(i+1) = \frac{|x_0-u_0+rT|\, e^{-\frac{t_w}{T}}}{8}\left(\frac{t_a(i+1)}{T}\right)^2 = \Delta_{max}(i)\, e^{-\frac{t_w}{3T}} \qquad (11)$$

Now, after simple manipulations using (11,8,10), the subintervals $[t_i,t_{i+1}]$ can be found from:

$$t_w(i+1) = t_w(i)\, e^{\frac{t_w(i)}{2T}}, \qquad t_w(i) = t_{i+1} - t_i \qquad (12)$$

However, the latter result is overoptimistic. One could expect to take advantage of formula (12) by calculating from it the subsequent points t_i, t_{i+1}, t_{i+2}, .. without computing the p_i (only p_1 would be required). Unfortunately, the earlier mentioned truncation errors tend in this case to accumulate from segment to segment.

Usually, using the formula (12), the obtained time intervals are shorter than they should be due to the optimization criterion (4a,b,c). In consequence, the values $p(t_i,t_{i+1})$ are much less than p_{max} and the number of segments within $[0,t_{max}]$ becomes unnecessarily too big.

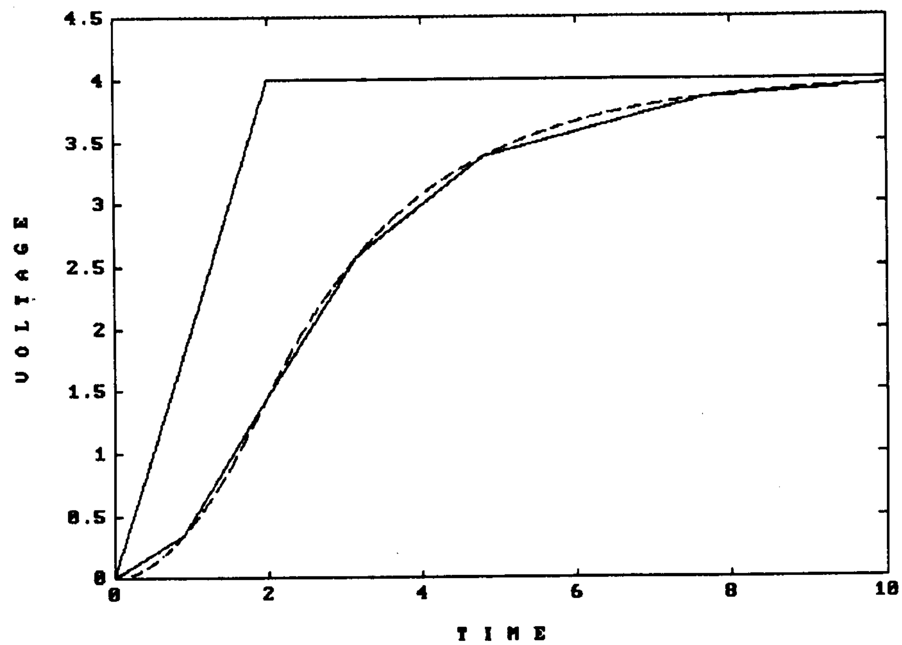

Figure 3. Model time response and its piecewise linear approximation.

3. Analog macromodels

At the functional simulation level we deal with analog units like amplifiers, adders, voltage comparators, multipliers or D/A converters. For the purpose of modeling by means of the proposed technique, their DC characteristics must be represented in a PWL form. Based on the introduced inertial building blocks, basic timing and DC specifications including some nonlinear effects, like saturation, are feasible. Other nonlinear effects arising at a signal multiplication can also be handled, since the respective products are strictly quadratic functions of time and hence the basic algorithm also holds but in this case it can be replaced by a much simpler procedure. Apparently, the t_a need not be computed at all. Moreover, if we assume the quadratic function to be expressed as: $x=\alpha t^2$, $t\in[0,t_{max}]$, the first linear segment crossing x at $t=0$ and $t=t_1$ takes the form of: $x_1=\alpha t_1 t$. For this segment $p(0,t_1)=\alpha(t_1)^2/4$. Hence, for the given accuracy p_{max}, the t_1 should be found from: $t_1=2\mathrm{sqrt}(p_{max}/\alpha)$. It can be also observed that to keep the accuracy for the next segments the same length for each of them should be chosen, i.e. $t_{i+1}-t_i=t_1$. This efficient formula is very useful while dealing with analog multipliers or controlled switches.

Time delays having their origin in saturation effects (e.g. for comparators) can be modelled by means of two inertial building blocks (1) in cascade. The first one should slow down the input signal and control the second stage, putting it from the positive saturation state into the negative one or vice versa. In this way an impact of the initial input polarisation and the overdrive on the output response can be modelled. A careful scaling procedure should be used for particular model specification to obtain the optimal parameter set of it.

Here however, only the macromodel for an amplifier is discussed in detail.

For an amplifier, usually the following specifications must be accounted for: the gain, the dominant pole, the saturation, the output resistance and the slew rate. For small input/output signals a fully linear macromodel would

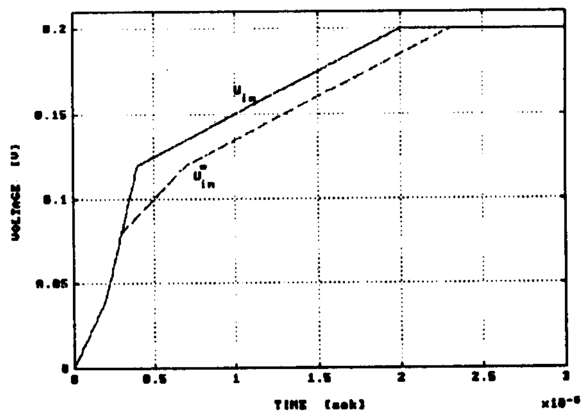

Figure 4. Slope limiting mechanism.

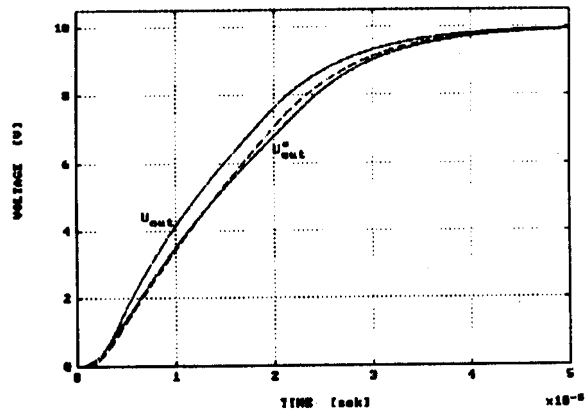

Figure 5. Time responses for input from Fig.4.

be sufficient. However, to cover possibly the full range of input amplitudes a nonlinear function f(o) and a special slope limiting mechanism (SLM) [1] must be used. It begins to act when $|u_{in}| > u_{th}$, where u_{in} is the input signal and u_{th} the threshold voltage, that puts the amplifier input stage into saturation. Then each linear segment of u_{in} with amplitude above this threshold is checked for the slew rate parameter SR. Case the segment slope $|du_{in}/dt| > SR/k$, its value is reduced to the limit SR/k (k is the amplifier gain) and the next PWL segments are shifted appropriately along the time axis to avoid time discontinuities. Beside that, two cascaded building blocks and are required. Denoting by u^*_{in} the signal obtained from the SLM, for the first block we have:

$$T_0 \frac{dx}{dt} + x = k u^*_{in}, \quad when \ |k u^*_{in}| \le U_{os} \tag{13}$$

$$T_0 \frac{dx}{dt} + x = U_{os}, \quad when \ |k u^*_{in}| > U_{os} \tag{14}$$

where T_0 is the inverse of a dominant pole frequency ω_0 and U_{os} is an output saturation voltage. For the second building block we have:

$$T_1 \frac{dy}{dt} + y = x \tag{15}$$

where $y = u_{out}$ is the amplifier output signal and T_1 involves the output resistance R_{out} and output capacitance C_{out} as well as a capacitive load C_l, (see (2)).

In Fig.4 the functioning of the SLM is shown. The u_{in} signal consists of four PWL segments. The SLM limits the slope of the second segment above the threshold voltage that has been assumed to be equal 80 mV. It corresponds to a noninverting amplifier (based on bipolar OPAmp), for which a typical value of u_{th} is approximately 100mV [10] and the SR=0.5 V/μs. The remaining parameters are chosen as follows: the closed loop gain k=50 and ω_0=125600 rad/s. The slopes of the next segments need not be limited since they are less than SR/k. The solid line denotes the original input signal and the dashed line the limited one. In Fig.5 the amplifier output waveforms for u_{in} are given (the PWL approximations are not plotted for the clarity of diagram). The u^*_{out} solid line has been obtained with the SLM, whereas the u_{out} solid line without of it. For comparison the SPICE estimate of the u^*_{out} is also plotted (dashed line). For other values of k ($k\omega_0$=const.) the time responses for various input signals are also good approximation of the SPICE estimates. However, no claim is made regarding the PWL macromodel suitability to mimic ideally the real amplifier behavior for all situations. Using a more natural way to represent the slew rate effect, like for a closed loop OpAmp [6,9,10], leads to the problem of a tight feedback. The nonlinear function f(o) in the first block must depend no more on u_{inp} but on the difference u_{inp}-y/k. In this case the presented PWL approximation algorithm should be supported by the local waveform relaxation. However, because of its very slow convergence for tightly coupled loops [8], it does not seem reasonable to give up this simple technique. Certainly, some accuracy is sacrificed in this way.

4. Summary

A macromodeling technique for the simulation of analog networks has been presented. Merely, the MOS networks have been addressed here. The particular units are assumed to be unidirectional ones, however the loading effects are allowed. The signals are represented in the piecewise linear form and while applied to the model inputs, enable to mimic the real timing behavior. The computation of the PWL waveforms formulated as an approximation task is very efficient. Some of the macromodels have been implemented within a prototype event-driven functional-level simulator, similar to that one presented in [1]. To proceed with A/D networks the basic digital units, like logic gates or registers are included too. Since logic operations AND,OR,INVERT are replaced by the analog functions [1], all the units, both analog and digital ones, are treated in a unified way. However, for complicated units (merely digital ones) their behavioral specification is also necessary to define the macromodels. The simulator is event-driven and subsequent points (t_i, V_i) of the PWL waveforms are defined to be the events

number of events to be processed and the required relaxation based iterations are likely to becoming a serious drawback to overcome.

6. References

[1] G.Ruan, J.Vlach, J.Barby, A.Opal, *"Analog Functional Simulator for Multilevel Systems"*, IEEE Trans.on CAD, vol.10, No.5, May 1991.

[2] A.Vachoux, K.Nolan, *"Analog and Mixed-Level Simulation with Implications to VHDL"* in "Fundamentals and Standarts in Hardware Description Languages" J.P.Mermet,Ed., NATO ASI Series E: Applied Sciences - Vol.249, Kluwer Academic Pub. 1993.

[3] A.E.Ruehli,Ed., *"Circuit Analysis, Simulation and Design"*, Part 1&2, Elsevier Science Pub.1986,1987

[4] R.A.Saleh, A.R.Newton, *"Mixed-Mode Simulation"*, Kluwer Academic Pub. 1990.

[5] A.F.Schwarz, *"Computer-Aided Design of Microelectronic Circuits and Systems"*, vol.2, ch.9.3,Academic Press 1987.

[6] L.O.Chua, P.Lin, *"Computer-Aided Analysis of Electronic Circuits"*, Englewood Cliffs, N.Y.:Prentice Hall, 1975.

[7] W.I.Zangwill, *"Nonlinear Programming"*, Prentice-Hall,