Efficient Transient Simulation of Non-linear Dynamic Networks with Discontinuous Forcing

Thomas Uhle  Günter Elst

Fraunhofer Institute for Integrated Circuits IIS, Design Automation
Division EAS, Zeunerstr. 38, 01069 Dresden, Germany (e-mail: thomas.uhle@eas.iis.fraunhofer.de)

Abstract: New technologies produce highly complex hybrid systems containing analogue components next to digital hardware and software. Simulating those mixed-signal systems exhibits discontinuities in signals due to the nature of the digital components. In this paper, an efficient and accurate method for the transient analysis of non-linear dynamic systems is presented that can handle discontinuous forcing functions which may occur as prescribed voltages or forces, for instance. It is based on a transformation of a model's equation system by substituting the time for a new independent variable. This approach facilitates suitable system-level models that altogether maintain the performance that is needed for an overall system-level simulation of complex application scenarios.

Keywords: differential equations, discontinuities, nonlinear systems, numerical methods, transient analysis

1. MOTIVATION

New technologies produce more and more highly complex hybrid systems containing analogue components next to digital hardware and software. Approximately 80% of today’s Systems on Chip (SoCs) are mixed-signal chips due to ITRS (2007). Most of their input and output signals, e.g., signals from sensors and to actuators, are still analogue because our world and how we interact with it is analogue. However, major parts of the systems are digital due to different reasons. One reason is that circuits for signal processing and controls, for instance, are mostly digital. Another reason is that analogue circuits cannot be scaled as easily to the smaller lithographic dimensions of newer technologies as digital circuits. That is why the number and complexity of analogue components are being reduced. The remaining analogue parts are then “assisted” by digital components as much as possible. In this way, the performance and configurability of a typical analogue front-end is increased through a tight interaction with a digital regulation and control subsystem. Sophisticated signal processing is applied to compensate analogue imperfections. Nowadays, we talk more and more about “digital assisted analogue”, see Rattner (2008) for instance. We are particularly facing those applications in the automotive domain. More and more mechanics are either replaced or assisted by electronics. Consider the brush-less motor for instance: its mechanics are simple and inexpensive, but it therefore requires tightly coupled sophisticated electronics. A second example may be the pulse width modulation stages for the controlling valves of an anti-lock braking system where the non-linear backlash of the valves has a strong impact on the software algorithm that is running on a microcontroller.

The consequence of these design trends is that it is getting ever harder, in some cases even impossible, to design and simulate the analogue components independent of the digital hardware and software or to design any integrated circuit and software algorithm without detailed knowledge of the full system environment. These mutual dependencies make clear paper-based specifications for single parts of a system almost impossible. This is why executable overall system-level models of mixed-signal or hybrid systems have become design-essential for an increasing number of SoCs.

SystemC/SystemCAMS is a C++ based model description language with the focus on abstract modelling of large complex systems with digital, analogue and mixed-signal hardware/software subsystems to facilitate high-performance system-level simulations of “real-time” application scenarios, see OSCI AMS Working Group (2008). In addition to the current SystemC AMS language standard, see OSCI AMS Working Group (2010), a new model of computation (MoC) for the simulation of non-linear dynamic networks with capabilities similar to those of languages like VHDL-AMS and Verilog-A/AMS was developed to enable the modelling of highly detailed models of small analogue parts tightly coupled with models of other parts (e.g., driver stages with pulse width modulation). Please read Uhle and Einwich (2010) for more in detail explanations of the newly proposed language constructs and its capabilities.

Co-simulation of digital and analogue components is normally done by coupling the various parts of a system via directed signals. Some of these signals may be considered as input signals with respect to the analogue parts, the others as output signals. Values of these input signals change at discrete points in time, known as events. The waveforms of prescribed quantities caused by these signals are therefore expected to be discontinuous. Usually, a common approach of most circuit simulators is to replace the steps in the stimulus functions by steep finite slopes.
Then the resulting waveforms are continuous again. These steep slopes are really challenging for every simulator. Huge values in the derivatives of the network variables may occur and so may harm both, the simulation results and the simulation performance. Moreover, during the concept phase of a design process, appropriate rise and fall times of the logic gates’ output voltages, for instance, are not known. Hence, discontinuities in input signals cannot be replaced by steep finite slopes and, thus, are indispensable for a modelling methodology at higher levels of abstraction.

2. NETWORK MODELLING IN BRIEF

It is common knowledge that real circuits can be described by Kirchhoff networks or multipole networks, for instance, in order to simulate their behaviour, see Reibiger (2009), Belevitch (1968) or Desoer and Kuh (1969) for instance. Thus, networks are interconnections of models for circuit elements. Each of these models is described at least by a set of terminals and a constitutive relation that describes its terminal behaviour. The interconnections of these models can be specified by identification of their terminals by means of an equivalence relation. The representatives of its equivalence classes are called nodes.

It is assumed that the reader has preliminary knowledge of how a network equation system is set up. Ho et al. (1975), for instance, describe how to set up a network equation system with respect to the rules of a modified nodal analysis (MNA). Such an equation system

\[ f_1(t, y_a(t), y_d(t), \dot{y}_d(t)) = 0 \]
\[ f_2(t, y_a(t), y_d(t)) = 0 \]  

(1)

generally consists of implicit differential and algebraic equations (DAE) where \( t \) denotes the value of the instantaneous time, \( y_a \) the vector of the algebraic network variables (e.g., terminal voltages), \( y_d \) the vector of the differential network variables (e.g., branch currents, charges, fluxes) and \( \dot{y}_d \) the vector of their derivatives. For this time, \( \partial_t f_1 \) is assumed to be a regular matrix. This DAE system can be discretised by implicit integration methods (e.g., linear multistep methods) and then be solved by non-linear solvers, see Pillage et al. (1994). Notice that also non-electrical networks (e.g., thermal, mechanical, fluidic, magnetic) can be described by (1) using the common analogies, see Reinschke and Schwarz (1976) and Reibiger (2009) for more details.

3. TREATMENT OF DISCONTINUOUS FORCING FUNCTIONS

The determination of consistent initial values after a discontinuity in a forcing function of a prescribed quantity is discussed in a variety of publications. Recent approaches with numerical methods as in Barton and Galán (2000), Reißig et al. (2002), for instance, yield consistent initial values but require that the regarded DAE systems are linear. Then the initial values can be determined by calculating the step response in the Laplace domain. Some approaches as in Bedrosian and Vlach (1992), Opal and Vlach (1990), for instance, also allow some non-linearities in the DAE systems introduced by diodes or transistors. Since these diodes and transistors are treated as ideal switches, the “remaining” system is also linear and the step response can again be calculated in the Laplace domain. Another drawback of the latter method is that there are 2\( n \) different DAE systems for a network with \( n \) of these switches. With the approach presented in this paper, consistent initial values can be determined for arbitrary non-linear dynamic systems, too.

For an efficient transient analysis of a non-linear hybrid system, it is desirable to calculate its solution using an approach with only one numerical method solving one and the same system of equations describing the analogue subsystem. That means that this numerical method shall compute the transient behaviour of the analogue subsystem on time intervals without discontinuities in forcing functions as well as consistent initial values after each event of a discontinuity.

In the following, we consider analogue systems whose differential variables must remain constant across discontinuities in forcing functions to preserve energy conservation. The approach presented in this paper is based on a transformation of the DAE system by substituting the time for a new independent variable. This substitution and a tailored function connecting the left- and right-hand limits of the values of the forcing functions at the discontinuities allow the numerical integration of the non-linear dynamic system between the events of discontinuities as well as the calculation of consistent initial values after each of these events. Note that only one DAE system is therefore needed to be set up.

3.1 Description and Application of the Method

Although the following method can be applied to a wider range of use cases, the probably best way to briefly sketch our ideas might be by means of a simple example circuit as shown in Fig. 1. In section 3.3, generalised approaches are presented.

The DAE system for the rectifier’s network model with MNA may be

\[ \frac{v_C}{R} + i_C - I_{sat} \left( e^{\frac{v_C}{V_T}} - 1 \right) = 0 \]  

(2)

\[ C \dot{i}_C - i_C = 0 \]  

(3)
where $v_C$ and $i_C$ are the network variables and $v_s$ is the prescribed voltage of the voltage source, a sinusoid with the frequency $f = 50$ Hz and the amplitude $a$,

$$a(t) = \begin{cases} 10 \text{ V} & \text{if } t \leq 105 \text{ ms} \\ 3 \text{ V} & \text{if } t > 105 \text{ ms} \end{cases}.$$

(4)

Surely, the network variable $i_C$ can be eliminated by adding (2) and (3), but, as it turns out, it is necessary to keep the differential equation (3) in addition to (2).

To answer the question which network variables should be treated continuously across the discontinuity in $t$ (see Fig. 3), which remains constant on a defined interval $[\tau_1, \tau_2]$, i.e.,

$$t(\tau) = \begin{cases} \tau - \tau_1 & \text{if } \tau \leq \tau_1 \\ 0 & \text{if } \tau_1 < \tau \leq \tau_2 \\ \tau - \tau_2 & \text{if } \tau > \tau_2 \end{cases}.$$

(5)

So we finally get (6, 7) with $t' = 0$ solved on $(0, \tau_1)$ using a common integration method. After that the DAE system (6, 7) with $t' = 0$ is solved on $(\tau_1, \tau_2]$ using the same integration method beginning with the initial value $(\bar{v}_C(\tau_1), i_C(\tau_1))$ until $\lambda = 1$. Its step size control ensures convergence of the applied integration method, especially the convergence of the Newton-Raphson method that solves the non-linear equation (6). Generally, $\lambda$ can be a non-linear function of $\tau$ which is constructed by a continuation method for solving difficult non-linear algebraic equations, see Allgower and Georg (1990). In this general case, $\lambda$ is also a system variable.

So we finally get $(\bar{v}_C(\tau_2), i_C(\tau_2))$ with $\bar{v}_C(\tau_2) = v_C(\tau_1)$ which becomes the initial value for the integration after the discontinuity when $t' = 1$ again. This procedure is analogously repeated for any other discontinuity.

**3.2 Simulation Results**

The results of the complete simulation of the example circuit with an additional discontinuity at $t_s = 305$ ms are shown in Fig. 4. The calculation of the solution on the complete domain $(0, 350$ ms) is sequentially done like it is described in the previous section (resulting waveforms are blue if $t' = 1$ and red if $t' = 0$). There is no difference to solving any other initial value problem because the left-hand limits of the derivatives of the network variables do not equal their right-hand limits. Only the instantaneous values of the network variables are taken as the initial values for the next integration.

**3.3 Generalisations**

As already mentioned before, the presented method can be applied to any other initial value problem whose equation

$$\bar{v}_C + i_C - I_{sat}(v_C - v_s) - 1 = 0 \quad \text{and} \quad C \dot{v}_C - \bar{i}_C = 0.$$

(6)

(7)

with $\lambda: [\tau_1, \tau_2] \rightarrow [0, 1]$. It is important to realize that charge conservation is preserved at the event of a discontinuity due to (7) because

with $t' = 0$ we have $C \dot{v}_C = \dot{q} = 0$,

(1)

and, hence, $\dot{v}_C(\tau) = 0$ on $(\tau_1, \tau_2)$. That is why $\bar{v}_C$ stays constant and, thus, $v_C$ is continuous across a discontinuity in $v_s$, whereas $i_C$ may be discontinuous. Notice that $i_C$ is determined by (6) only.

Similar to the method of source stepping, assume

$$\lambda(\tau) := \frac{\tau - \tau_1}{\tau_2 - \tau_1} \quad \text{ across the discontinuity at } t(\tau) = t_s.$$

(9)

across the discontinuity at $t(\tau) = t_s$. Hence, the waveform of $\bar{a}$ is linear on $(\tau_1, \tau_2]$ like it is illustrated by the dotted line in Fig. 2. So in contrast to $v_s$, $\bar{v}_s$ is continuous, and even smooth on $(\tau_1, \tau_2)$. The results of the complete simulation of the example circuit with an additional discontinuity at $t_s = 305$ ms are shown in Fig. 4. The calculation of the solution on the complete domain $(0, 350$ ms) is sequentially done like it is described in the previous section (resulting waveforms are blue if $t' = 1$ and red if $t' = 0$). There is no difference to solving any other initial value problem because the left-hand limits of the derivatives of the network variables do not equal their right-hand limits. Only the instantaneous values of the network variables are taken as the initial values for the next integration.
system can be written in the implicit form (1). With the substitution
\[ z := \dot{y}_d \]
we obtain an extended DAE system
\[
\begin{align*}
    f_1(t, y_a(t), y_d(t), z(t)) &= 0 \\
    f_2(t, y_a(t), y_d(t)) &= 0 \\
    y_d(t) - z(t) &= 0
\end{align*}
\]
with the system variables \( y_a, y_d, \) and \( z \). After substituting time \( t \) for \( \tau \) similar to (5) we have
\[
\begin{align*}
    f_1(t(\tau), \dot{y}_a(\tau), \dot{y}_d(\tau), \dot{z}(\tau)) &= 0 \\
    f_2(t(\tau), \dot{y}_a(\tau), \dot{y}_d(\tau)) &= 0 \\
    \dot{y}_d(\tau) - \dot{t}(\tau) \cdot \dot{z}(\tau) &= 0
\end{align*}
\]
with the new system variables \( \dot{y}_a, \dot{y}_d, \) and \( \dot{z} \). Although the state space is not changed by the substitution (10), the number of variables of the new DAE system (11) is larger than the number of variables of the original DAE system (1). Possible differences of their numerical properties are to be investigated.

For some special cases of (1), the substitution (10) is not needed, so that the number of variables does not change. At first let \( f_1 \) be a sum of \( f \) and \( \bar{f} \), i.e.,
\[
\begin{align*}
    f_1(t, y_a(t), y_d(t)) &= \tilde{f}(t, y_a(t), y_d(t)) \\
    f_2(t, y_a(t), y_d(t)) &= 0
\end{align*}
\]
with
\[ \tilde{f}(t, x) = 0 \quad \Rightarrow \quad x = 0 \]
for any discontinuities at \( t_n \). Every component of \( \tilde{f} \) shall depend on exactly one differential variable of \( \dot{y}_d \). Then substituting \( t \) for \( \tau \) leads to
\[
\begin{align*}
    \tilde{f}(t, \dot{y}_d(t)) + t' \tilde{f}(t, y_a(t), \dot{y}_d(t)) &= 0 \\
    f_2(t, y_a(t), \dot{y}_d(t)) &= 0
\end{align*}
\]
with every single argument being a function of \( \tau \) which is but not written here and in the following formulae due to readability reasons. If \( \tilde{f} \) is even linear in its second argument, then we have
\[
\begin{align*}
    E(t) \dot{y}_d(t) + t' \tilde{f}(t, y_a(t), \dot{y}_d(t)) &= 0 \\
    f_2(t, y_a(t), \dot{y}_d(t)) &= 0
\end{align*}
\]
with \( E \) being a diagonal matrix or a permutation of it. If the functions \( f \) and \( \tilde{f} \) are linear combinations of states \( x \), inputs \( u \), and outputs \( y \) and \( E(t) = 1 \), then (1) can also be written in matrix notation of a linear state space system with the well-known \( A, B, C, \) and \( D \) matrices. In this special case we have
\[
\begin{align*}
    \dot{x}' &= t'(A(t)\bar{x} + B(t)\bar{u}) \\
    \bar{y} &= C(t)\bar{x} + D(t)\bar{u}
\end{align*}
\]
4. CONCLUSION

An efficient and accurate method for the transient analysis of non-linear dynamic systems with discontinuities in the forcing functions of its prescribed quantities has been presented. In contrast to recent approaches, the DAEs of the systems may be non-linear. Moreover, the numerical method for calculating consistent initial values for the integration after the discontinuities can be the same as the one usually utilised to solve these DAE systems without discontinuous forcing. There is no difference to solving any other initial value problem. For some special cases of DAE systems, generalised formulae for the calculation of the transformed equation system have been derived. Although this approach is not restricted to a specific model description language like SystemC AMS, it facilitates suitable system-level models for an efficient overall system-level simulation of large complex mixed-signal or hybrid systems.

REFERENCES


