
©20xx IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Symbolic Small-Signal Analysis (SSA) Tool

Devrim Aksin  
Department of Electrical Engineering,  
The University of Texas at Dallas  
Dallas, TX, USA  
devrim.aksin@ieee.org

F. Maloberti  
Department of Electrical Engineering  
University of Pavia  
Pavia, Italy  
franco.maloberti@unipv.it

Abstract — the symbolic analysis of analog circuits and systems is made easy by the use of the SSA tool. The input description of the system to be studied is done with a file description or more conveniently, using the Cadence or the Simulink graphic environment. Libraries of basic blocks enable the description of complex systems including high-order sigma-delta modulators. The symbolic results produce various types of output depending on the kind of post-processing required. Examples of use of the tool are provided.

I. INTRODUCTION

Symbolic analysis is important for studying the small signal behavior of any circuit or system. The small-signal equivalent circuit enables stability verification and noise performance estimations. The availability of symbolic equations containing key design parameters is also useful for system optimization. However, the difficulty in obtaining the symbolic description and the complexity of equations to be solved are practical limits that favor a blind optimization of the design based on a try-and-see approach.

The symbolic study becomes unfeasible with complex non-linear components (the description of sub-micron transistors is an example of complex non-linear elements). The number of parameters used in the modeling is so large and not directly linked to design degrees of freedom that the symbolic description is not usable. Instead, high-level behavioral models describing, for example, an operational-amplifier or an integrator can be based on a reasonable number of parameters whose optimum value helps the overall system optimization.

Writing equations and solving linear systems is time-consuming and error prone. Therefore, the behavioral study requires using suitable computational tools that help in extracting the behavioral result [1]. This is the goal of this paper. It uses an existing engine for symbolic solution but it provide a simple and user-friendly way to obtain the final result starting from a block diagram drawn using commercial CAD tools for system. The tool, named SSA Tool, uses the Cadence DFII and the Simulink environment for the graphical interface and Matlab for solving the symbolic equations.

II. TOOL DESCRIPTION

Three main sections characterize the use of the tool. They are: 1. Input section (system description); 2. Netlist parsing and symbolic circuit solver; 3. Post processing

The key features of the three sections are described in details below.

A. Input section (system generation)

The SSA tool permits us to describe systems in three different ways: 1.Cadence schematic; 2.Simulink schematic; 3.Text file description. The circuit netlist information within Cadence and Simulink schematic is converted to a custom intermediate SSA Tool netlist. Text file description should comply with the custom SSA Tool netlist.

The nomenclature of SSA tool netlist line is specified as {definition, parameters}. The definition section contains an acronym related to the content nature (resistor, transistor, analysis type, etc…) of the line and parameter section contains the values of the required data related to that specific definition. Other features are described in the SSA Tool manual [2].

The Simulink input uses a special design library, named sslibray and requires a pre-processing step to obtain the SSA tool netlist file. The sslibrary contains all necessary components to design systems like, for instance, ΣΔ modulators. Among the available components, we have zero order hold, unit delay, discrete time integrator, sum, gain, product, quantizer. The library also enables special functions like polynomial, discrete transfer function. The input to the block can be a constant, a sine wave, a step, etc. The outputs are identified by scopes.

Since the Simulink schematic editor requires one and only one driving block for any node, it is not possible to have bidirectional port required by resistor, capacitor, etc… This kind of elements is possible with Cadence input. The designer can draw the schematic of Fig. 2 to describe a simple first order ΣΔ modulator. The circuit is very simple and is used here just to illustrate the method. The parameter b accounts for the gain error of the integrator. The design
procedure is exactly the one required by the simulation of the system in the time-domain. A Matlab function, named
\texttt{convsimu()}, is used to convert simulink model file to the SSA tool netlist. The function performed is basically a text processing. The routine reads the Simulink model file and extracts block and connection information. The automatically generated SSA tool netlist file for the first order \(\Sigma\Delta\) modulator looks like:

\begin{verbatim}
% Simulink to SSA Conversion Tool Output
% Written By Devrim AKSIN
% Version 1.
% Grounds {g,gn}
% Description {
%     vol,Integ1,Inteq1,gn,Sum*(1-b)/(z-1),
%     xtf,OUT1,Quantizer1,simp},
%     {vol,Quan1,Quan1,gn,Integrator1+epsil_1,}
%     {vol,SineWave,SineWave,gn,Input,}
%     {vol,Sum,Sum,gn,SineWave-Quantizer1,}
end
\end{verbatim}

The Cadence description also uses a design library (\texttt{SSALIB}) and requires two steps of pre-processing for obtaining the input netlist file. The \texttt{SSALIB} library contains 64 components, including all the Matlab Simulink blocks, linear and non-linear passive components, nonlinear active components, independent sources, dependent sources, and global sources. Fig. 2 shows an example of diagram that can be drawn with \texttt{SSALIB}. It is a simple amplifier with a bipolar transistor. First step towards the creation of the SSA tool netlist in Cadence environment involves the generation of the text based spectre(S) netlist. A Cadence Skill function is written for this purpose which add SSA Tool menu item to the banner of the schematic window. When invoked, SSA Tool launches a pop-up window asking for the run directory and desired netlist procedure, i.e. Spectre/SpectreS (Fig. 3). Then, the spectre(S) netlist is automatically created. A Matlab function, i.e. \texttt{comp2netlist()}, is used to convert spectre(S) netlist file to SSA tool netlist. This function is also a simple text processor. It reads the spectre netlist file and extracts block and connection information and finally creates the SSA tool netlist. The resulting netlist looks very similar to the example netlist given previously for the first-order \(\Sigma\Delta\) modulator.

\subsection{Netlist parsing and Symbolic circuit solver}

The SSA tool netlist generated by the input section is then parsed by Matlab function \texttt{main()} to extract requested analysis types and component descriptions. The syntax of the function is simply:

\begin{verbatim}
Out = main(netlist_file_name)
\end{verbatim}

Then, each component within the netlist is replaced by their small-signal equivalent circuit. Symbolic node equations are extracted from this later equivalent circuit netlist. Finally, the symbolic node equation system is solved by symbolic equation solver. The symbolic equation solver is the one provided within Matlab symbolic toolbox which is in turn based upon Maple kernel. Developed system basically extract the node equations of a given network netlist using its small-signal equivalent model and let the Matlab symbolic solver solves the linear symbolic equation system.

As will be detailed in section III, SSA tool is also capable of calculating circuit’s second and third order harmonic response provided that required non-linearity coefficients are given within the circuit netlist. For example the parameters \(K_2\) and \(K_3\) in Fig. 2 are the non-linearity parameters of the non-linear resistor 1297. The Matlab function \texttt{main()} makes necessary netlist modifications for harmonic response calculations and supervises the whole process of non-linear calculation. At the end of this step, the symbolic circuit response is obtained. The following result calculated by Matlab when the bipolar amplifier shown in Fig. 2 is solved by SSA Tool:

\begin{equation}
Out = -\frac{gm}{(G_e + go)} \frac{gb}{(gb + gpi)} V_{\text{IN}}
\end{equation}

Notice that the first term in (1) is the transconductance of the transistor multiplied by the equivalent output resistance and the second term is the base-emitter voltage of the BJT which is the output of the resistive divider made of \(gb\) and input resistance \(gpi\) of BJT.
C. Post Processing

After obtaining the circuit responses the SSA tool provides several different ways to post-process the symbolic solutions. Namely, it is possible to obtain the followings:

- Calculation of signal and noise transfer function of $\Sigma\Delta$ modulator in z-domain. SSA tool exploits the fact that the output signal of a $\Sigma\Delta$ modulator is in the form of SNF,$X+NTF,\varepsilon$ where SNF is signal transfer function, NTF is noise transfer function, $X$ is the input signal and finally $\varepsilon$ is the quantization error, to simplify greatly the final expression. The module calculates the SNF, NTF and quantization noise power in terms of circuit parameters.
- Calculation of real and imaginary parts of the impedance, admittance or transfer function.
- Execution of custom written Matlab codes to further process the circuit solutions.

III. NON-LINEAR ANALYSIS

The SSA Tool is also able to estimate the harmonic responses of weakly non-linear circuits (or low-distortion circuits). The method used is the one described in the book ‘Non-Linearity analysis of Analog Integrated Circuit’, Piet Wamback and Willy Sansen [1]. The weakly non-linear assumption imposes the following conditions: 1) the non-linear circuit is driven by a small enough input signal such that $n^{th}$ order harmonic response can be determined by only circuits $n^{th}$ order nonlinear behavior, 2) the energy of the output signal is mainly concentrated in the lowest harmonics (in other words, the array of amplitudes of the even and odd order harmonics is decreasing). Weakly non-linear behavior is typically due to the curvature of the device characteristics in a given operation condition (Whereas the strongly non-linear behavior is due to the operation region changes of the device such as turn on and off of a transistor.) Due to the weakly non-linear behavior assumption, higher order harmonics to the lower order harmonics are ignored [1].

SSA Tool calculates only second and third order harmonic responses. The reason is that the dominant harmonic terms of a single-ended and a differential circuit are in fact its second order and third order harmonics, respectively. The equivalent circuit of the components used can contain non-linear conductance, resistances, capacitance, trans-conductance and trans-resistance. SSA Tool solves the circuit response using the node equations. Therefore, it represents the non-linear resistance and non-linear trans-resistance in the form of $I=I(V)$ [1]. Finally, the method uses power expansion coefficients for device model calculated around device’s operation point (or biasing point). The assumption of weakly non-linear circuit allows ignoring the terms with order higher than three. SSA tool modifies the circuit netlist using provided power expansion coefficients according to the method defined in [1] and solves the modified netlist to obtain circuits higher order responses. Following second and third order responses of the bipolar amplifier shown in Fig. 2 is calculated by SSA tool[3]:

\[
Out = -\frac{1}{2} \frac{gm}{(G_c + g) (gb + gpi)} V_{IN}^2,
\]

\[
Out = -\frac{1}{4} \frac{gm}{(G_c + g) (gb + gpi - 2K_{gb})} gb^3 V_{IN}^3,
\]

Notice that only the nonlinearities related to $gb$ is defined within the schematic Fig. 2.

IV. EXAMPLE

The use of the SSA tool has been tested in many sampled data systems. Here we give some results of the use of SSA tool to analyze the fourth order $\Sigma\Delta$ modulator shown in Fig. 4. The nominal parameters of the design are given in reference [4]. Using the SSA tool, the modulator transfer function can be immediately obtained, it is

\[
Out = \frac{0.4 \left(2z^3 - 5.2z^2 + 4.5805z - 1.3623\right) IN}{z^4 - 3.2z^3 + 3.92025z^2 - 2.1683z + 0.45533} + \frac{0.25 \left(4z^2 - 8z + 4.001\right) (z - 1)^2}{z^4 - 3.2z^3 + 3.92025z^2 - 2.1683z + 0.45533} \varepsilon.
\]

The plot of the NTF and STF are given in Fig. 4. The STF is flat in the signal band and the NTF shows a transmission zero due to the resonator controlled by the
The variation of the NTF with respect to parameter $g$. It is possible to use SSA to determine the sensitivity of the NTF to the variation of $g$ and, estimate the modification of the NTF. Fig. 6 plots the obtained results.

The tool also enables to determine the STF and NTF as a function of one or more design parameters. For example, the coefficient $a_k$ affects the responses as shown in (5)

$$Out = \frac{0.4(2z^2 - 5.2z^2 + 4.5805z - 1.3803 + 0.012a_k)}{z^4 - 3.2z^3 + 3.92025z^2 - 2.1683z + 0.44813 + 0.0048a_k} IN$$

$$+ \frac{0.25(4z^2 - 8z + 4.001)(z - 1)^3}{z^4 - 3.2z^3 + 3.92025z^2 - 2.1683z + 0.44813 + 0.0048a_k} e$$  \hspace{1cm} (5)$$

Observe that both STF and NTF changes. Designer can assess the effect for a suitable care in the design or the layout.

The use of SSA also makes it possible to analyze the stability of the modulator by using the linear root locus method. The quantizer is replaced by a gain stage to obtain the expression given in (6) ($K$ is the gain of the quantizer.) The post-processing obtains the pole zero plot given in Fig. 7. Thus, for example it is possible to verify that the modulator becomes unstable for $K$ values below 0.567.

It is also possible to analyze with SSA tool, the effect of circuit non-idealities, such as finite DC gain and gain bandwidth product of the amplifiers, capacitive mismatch, finite switch-on resistance, amplifier noise, noise at any given node within the system, to the modulator output. These equations than can used to determine required performances of different sub-block within the system.

V. Conclusion

The small signal analysis tool described in this paper provides many features to help the designer in the symbolic analysis of systems. The tool does not include a specific symbolic equation solver or provide specific graphic input interface. The strategy is to exploit commercially available tools and, by integrating them, optimizing the use for a more effective design activity.

REFERENCES