Automating the Use of State-Space Representations in Mixed-Signal IC Design and Verification

Francesco Stilgenbauer[1][5], Matteo De Ferrari[1],
Cristiano Meroni[5], Giuseppe Ridinò[3], Cristian Macario[4],
Paolo Stefano Crovetti[2], Edoardo Bonizzoni[1], and Piero Malcovati[1]
[1]University of Pavia, Pavia, Italy, [2]Polytechnic University of Turin, Turin, Italy
[3]MathWorks, Turin, Italy, [4]MathWorks, Munich, Germany
[5]STMicroelectronics, Cornaredo, Italy

***Abstract*—This paper proposes a methodology based on the modified nodal analysis (MNA) automating the extraction of symbolic State-Space (STSP) models directly from electrical circuit netlists, uniquely defining outputs within the schematic itself. The high level of automation achieved significantly reduces the modeling effort and enables rapid adaptation to changes in specifications or circuit topology. Furthermore, this paper details the process by which the extracted models can be utilized to generate behavioral components for use within both analog and digital simulation environments. This ensures a consistent design flow throughout the integrated circuit (IC) design process and markedly enhancing workflow efficiency.**

***Index Terms*—State-Space, Electrical Circuits, Code Generation, DPI-C, Verilog-A, Mixed-Signal Systems.**

I. Introduction

The growing complexity of ICs, coupled with the pressing need for architectural optimization and faster time-to-market, requires advanced methodologies for architectural modelling. In the context of mixed-signal designs, involving both analog and digital teams, the necessity for effective collaboration is particularly pronounced due to the extensive variety of potential participants, including system-architecture designers. Although each team has its established methodologies, model sharing frequently relies on documentation and imagery, necessitating redundant adaptation efforts by all teams. This inefficiency in reuse often stems from the different requirements of each team. A novel methodology must carefully address the unique aspects of every design phase while maintaining the overarching objective of maximizing reuse. The parallelization of design activities is another critical factor. It is crucial to allow all teams to begin their work early in the design process by providing them with high-level models of the components from other teams. This study, which applies the STSP approach [1] to electrical circuits, exploits conventional IC design tools. It ensures that these circuits, along with their parametrization, are integrated throughout the subsequent design stages. In recent decades, other research has also moved in this direction. The study in [2] introduces a systematic method for obtaining the STSP formulation. Work in [3] underscores the need to transition from the integral-differential equation approach to an STSP framework, proposing a method to derive state equations for degenerate and coupling circuits. While the STSP is proficiently used in [4], it does present certain drawbacks, particularly when changes in circuit topology decelerate the process. Overcoming these limitations is crucial, and extensive automation serves as the solution, facilitated by the symbolic approach acknowledged in [5]–[7].

The paper is organized as follows. Section II lays the foundation for symbolic STSP use. Section III presents a consistent and automated methodology for symbolic generation. Section IV shows how the solution can be utilized in fully digital verification testbenches. Section V leverages the analysis results in the context of analog simulation environments. Section VI draws the conclusions.

II. Background

A. Classic approach: the transfer-function

Both continuous-time (CT) and discrete-time (DT) linear time-invariant (LTI) circuits admit the so called transfer-function (TF) representation [8]. TF is suited to analyze the input-output relation and obtain some insights about the system (e.g., impulse response by anti-transformation or zero-pole location by solving the numerator and denominator equations). Unfortunately, this representation has some limits:

- The TF approach is applicable only to Single-Input-Single-Output (SISO) systems, while Multiple-Input-Multiple-Output (MIMO) systems are not supported.
- Internal system states are not described, only the input-output relation is provided.
- The TF represents the forced response of a system. Therefore, initial conditions are inherently not included.
- It is only adapted for LTI modelling.

B. Alternative approach: the STSP

To overcome all the TF approach limits in the context of LTI systems, it is possible to use the STSP representation that changes the standpoint from a single *n*-th order differential equation to a set of *n* first-order equations. For the CT case, the linear STSP form is described by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) \tag{1a}$$

$$\mathbf{y}(t) = \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{u}(t) \tag{1b}$$

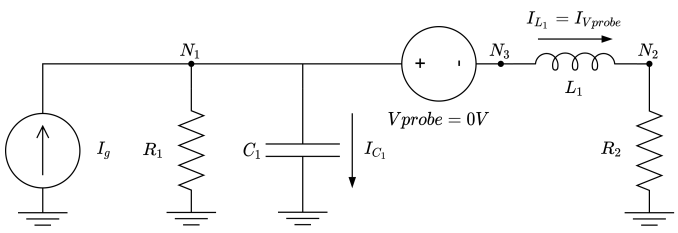

Fig. 1. RLC parallel resonator

where $\mathbf{u}$ is the input vector with size p, $\mathbf{x}$ is the state vector with size n, and $\mathbf{y}$ is the output vector with size q. $\dot{\mathbf{x}}(t)$ represents the state time derivative according to the Newton notation. The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ sizes will be defined according to (1a) and (1b).

By discretization of the time variable with $t = kT$, where T represents the sample time and $k \in \mathbb{N}$, it is possible to express the DT STSP equations in a form similar to the CT case

$$\mathbf{x}\left((k+1)\,T\right) = \mathbf{A_d}\,\mathbf{x}(kT) + \mathbf{B_d}\,\mathbf{u}(kT) \tag{2a}$$

$$\mathbf{y}(kT) = \mathbf{C_d}\,\mathbf{x}(kT) + \mathbf{D_d}\,\mathbf{u}(kT). \tag{2b}$$

The initial condition evolution uses the same system involving the input with no modifications for both CT and DT cases.

C. Electrical circuit STSP example

STSP representation is suitable for linear lumped elements circuits. In this context, the state vector is related to the reactive components: capacitors and inductors. In the simplest case where no degeneration occurs, the state vector is formed by voltages across capacitors and current through inductors. Reactive elements constitutive equations are the basis for the formulation of (1a) and (1b), where

$$\dot{V}_C(t) = I_C(t)/C \tag{3}$$

for capacitors and

$$\dot{I}_L(t) = V_L(t)/L \tag{4}$$

for inductors.

The procedure of extracting STSP models from circuit is deeply described in [2]. For example, Fig. 1 shows a parallel RLC circuit in which the component $Vprobe$ is disregarded and treated as a short-circuit. Consequently, node N_3 is eliminated because it coincides with node N_1. Starting from the Kirchhoff current law, the analysis of the two nodes yields

$$I_g - I_{L_1} - I_{C_1} = V_1/R_1 \tag{5a}$$

$$I_{L_1} = V_2/R_3 \tag{5b}$$

where V_1 and V_2 are the voltage of nodes N_1 and N_2, respectively.

The capacitor also provides two equations: its constitutive relation (3), which is directly substituted in (5a), and the voltage equation, which is in direct relation with nodes voltages $V_{C_1} = V_1$. Inductor current I_{L_1} is already present in (5b), while its time-derivative provides a new equation, $V_1 - V_2 = L_1 \dot{I}_{L_1}(t)$, which is based on (4). The previous equations constitute a non-redundant system that can be solved by substitution providing the STSP description for state evolution

$$\begin{pmatrix} \dot{V}_{C_1} \\ \dot{I}_{L_1} \end{pmatrix} = \begin{pmatrix} -1/R_1C_1 & -1/C_1 \\ 1/L_1 & -R_2/L_1 \end{pmatrix} \begin{pmatrix} V_{C_1} \\ I_{L_1} \end{pmatrix} + \begin{pmatrix} 1/C_1 \\ 0 \end{pmatrix} I_g \tag{6}$$

and for the output

$$I_{out} = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} V_{C_1} \\ I_{L_1} \end{pmatrix} \tag{7}$$

where the D matrix is null since there is no direct path between input and output.

D. Three reasons for using symbolic computations

LTI circuits can be solved using nodal analysis (NA) or, eventually, MNA. Here, substituting the numerical values directly in electrical components before the MNA may lead to the limitation described in [9], where the example of a very simple but critical circuit is reported. It consists in a single mesh formed by an independent current source and the series of two resistances R_1 and R_2. The conductance matrix obtained from the NA in case of very heterogeneous values of the two resistances may suffer from numerical truncation due to finite precision of the computer, leading to an ill-conditioned problem. This problem may be overcome by performing numerical substitution at the last step of the process.

Often in circuit design the impact of component variability due to the fabrication process is evaluated via Monte Carlo simulations [10]–[13]. For example, a given parameter values spread may impact the stability of a closed-loop system in which the circuit is inserted. If the stability analysis is performed via Bode plots considering gain margin and phase margin, the Monte Carlo approach relies on a statistical basis where a given number of diagrams is derived by properly randomizing the values of the components. The limit of this approach is intrinsic in its statistical-based analysis: there is always a certain residual probability that the cases analyzed are not considering a possible critical corner of component values. This limit is overcome by robust control methodologies, which leverage closed form solutions [14], wherein a symbolic approach is required.

Manual calculations for circuit response and sensitivity analysis are time consuming and prone to errors, especially when the complexity increases. The necessity of shifting to an automated symbolic process has already been recognized in [5]–[7].

III. Automating the generation process

In [4], the problem of modifications in a circuit modelled via STSP and consequent equation rewriting is highlighted. Automation is key to minimize the time spent during the generation process. In previous works, the necessity of automation is addressed, as in [15]. This approach is founded on symbolic solution in the Laplace domain. A step in the STSP direction

is in [16], where the symbolic A, B, C, and D CT matrices are generated from circuit netlist.

Our solution, developed in the MATLAB® environment, is named Spice2StateSpace and aims to minimize user intervention by limiting it to the schematic design phase. Multi-dimensional inputs are supported, both independent voltage and current sources. Provided with the netlist, the identification of inputs can be determined uniquely, whereas it is not possible for the outputs. To circumvent the problem, manual data-entry can be executed after the netlist call, which leads to slowing down the procedure. Our proposal for voltage outputs is to read them by means of a null current independent source connected across the desired nodes. Vice-versa, a current output is read with a null voltage independent source connected in a branch. The null value renders the probing element irrelevant in the original circuit. The concept is reported in Fig. 1 as the $Vprobe$ component.

Spice2StateSpace is based on MNA, described by

$$\mathbf{GV} = \mathbf{I} \tag{8a}$$

$$\mathbf{UV} = \mathbf{V_g} \tag{8b}$$

where $\mathbf{G}$ is the conductance matrix, $\mathbf{V}$ is the node voltages vector, $\mathbf{I}$ are the branches current, $\mathbf{U}$ is formed only by the $(1, 0, -1)$ set of values, and $\mathbf{V_g}$ is related to independent voltage sources. The further step performed in Spice2StateSpace is manipulating (8a) and inserting extra equations that complete the additional system provided by (8b).

Capacitors provide a contribution in the column $\mathbf{I}$ of (8a) by substituting the current term in (3). The result will contain the voltage derivative term that will be useful for the state evolution. It also provides an additional equation by means of its voltage $V_C(t)$ and the two connected nodes. Inductors contribute to the column $\mathbf{I}$ of (8a) directly with their current $I_L(t)$ and with an additional equation which links the node voltages to the relation (4) rearranged to remove the $V_L(t)$. This second equation will provide the current derivative term for the state evolution.

All kinds of controlled sources are supported. The voltage-controlled voltage source (VCVS) contributes with a current term in the interested positions of $\mathbf{I}$ of (8a) and an additional equation in which the output nodes voltage drop is the result of the input nodes voltage drop multiplied by the voltage gain. The voltage-controlled current source (VCCS) contributes only to $\mathbf{I}$ of (8a) with a relation depending on a transconductance gain applied between two voltage nodes. The current-controlled voltage source (CCVS) requires a special component insertion: a null independent voltage source with a dedicated naming convention is used during the schematic entry. The current through this source is used as the control variable. An additional equation that bounds the current reference with the controlled node voltage using the transresistance gain is needed. The current-controlled current source (CCCS) requires, as the CCVS, a null independent voltage source. In the interested parts of the $\mathbf{I}$ of (8a), the current terms relative to the controlled current are present as a result of the control current multiplied by the current gain factor.

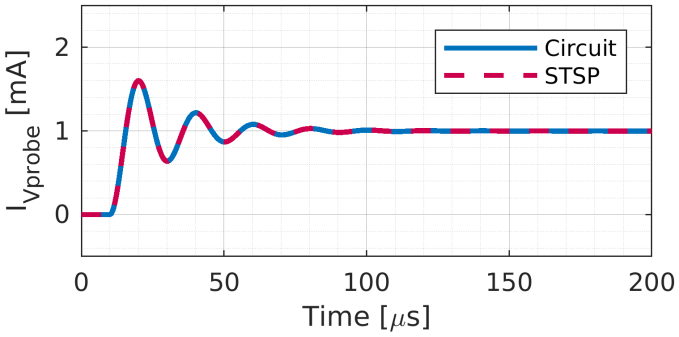

Fig. 2. Resonator step responses

Voltage output measurement introduces an additional equation related to the voltage drop across two selected nodes. Current output measurement affects two rows of $\mathbf{I}$ in (8a) that represent the desired current being sunk from one node and sourced to another. It also yields an additional equation that ensures the voltage drop across the two specified nodes is nullified.

The netlist is parsed one line at a time progressively forming the equations set. The obtained system includes an equation relative to the ground. This is eliminated only at this stage of the flow to maintain a uniform approach during netlist parsing. The obtained system is formally constituted by the following unknowns: $\dot{V}_C$, $\dot{I}_L$, the currents through the output voltage measurement current sources (they will be null, but is formally mandatory), the currents through the output current measurement null voltage sources, the currents through the VCVSs and CCVSs, the currents through the independent input voltage sources and the null voltage sources dedicated to CCVS and CCCS, and voltages of all nodes. It forms a non-redundant squared system that can be solved by the Cramer rule. Since we are interested in the STSP representation, only a subset of unknowns will actually be solved: $\dot{V}_C$, $\dot{I}_L$, output voltages, and output currents. All the operations are performed symbolically.

To verify the functionality of Spice2StateSpace, the step responses of the reference circuit in Fig. 1 and the generated CT STSP have been simulated in the Simulink® environment. The numerical values used are $R_1 = 100.0\Omega$, $R_2 = 100.0m\Omega$, $L_1 = 1.0\mu H$, and $C_1 = 10.0\mu F$ which are substituted into the symbolic variables only after the Spice2StateSpace execution finishes, then passed to the STSP model. The step input rise occurs at $t = 10\mu s$ with an amplitude of $1mA$. The results are reported in Fig. 2, showing the perfect correspondence of the two responses.

A. Ideal Operational Amplifiers modelling approach

Real operational amplifiers (OPAMP) are modelled by progressively including their non-idealities, where the first two considered are the finite gain and finite bandwidth. Finite gain may be modelled with a VCVS with a proper value and the finite bandwidth may be included by cascading a resistor-capacitor [17], representing a single-pole approximation which

may be followed by a unitary gain VCVS for decoupling. Since we rely on a symbolic approach, ideal OPAMP (as nullator/norator pair) is not supported as a primitive component. Instead, it is obtained starting from the finite-gain OPAMP model and, after the symbolic generation, applying the limit for voltage-gain which goes to infinity in the VCVS symbolic gain value.

B. Status degenerations prevention

A circuit may degenerate when it contains a mesh entirely formed by capacitors and voltage sources. In such cases, one capacitor voltage becomes a dependent variable. Similarly, the degeneration occurs when there is a node connected only to inductors and current sources. This kind of problem can be solved by adding at right side of (1a) and (1b) the terms $\mathbf{E}\dot{\mathbf{u}}(t)$ and $\mathbf{F}\dot{\mathbf{u}}(t)$, respectively. Dependence on the input derivative is introduced and is removable only in the first equation by proper manipulation but remains in the second. Since we consider dealing with non-ideal circuits, this problem can be circumvented by considering the parasitic resistances of capacitors and inductors to prevent the degeneration.

IV. Reconfigurable STSP within digital verification environments

Digital verification is driven by a high level of automation: ideally, the developed testbench requires human intervention only in case of bugs or unexpected behaviours. This is achieved with a self-checking approach where the developed software can recognize when IC behaviours exceed defined limits. The inclusion of models of the analog part is crucial to simulate the analog/digital interactions and mixed loops. Often, SystemVerilog hand-written models are developed, which is time-consuming. When the digital verification activities are conducted in event-driven simulation environments, the notion of time is emulated by a sequence of events that trigger the computation of internal states and outputs in the analyzed digital circuit. Simplifying the scenario, a unique clock is present with a fixed period ts. When the import of a CT circuit is needed in a digital verification testbench, a time-discretization must be performed as in [18]. Some methods of transformation are available, including but not limited to bilinear transform, impulse invariance, zero-pole matching, least-squares, zero-order hold, and first-order hold methods.

This section is focused on the zero-order hold discretization method (ZOH) [19] without loss of generality. Its derivation is based on the CT STSP time-domain solution in two subsequent sampling instants, producing a perfect match between the original CT and the derived DT systems with a step input.

A methodology is now shown to address the problem of model generation starting from an LTI electrical circuit and deriving a reconfigurable model suitable for a digital verification testbench. Considering the context of audio Class-D amplifiers, the differential LC filter in Fig. 3 can be used as an example [20]. Voltage sources are the linearized version of the power stages, including equivalent average resistors. With the load resistance, the LC resonator forms the dominant second order filtering contribution. Snubber and damping networks are considered. The null current generators are the voltage output measurement components, as described in III. Spice2StateSpace is deployed to transform the circuit in Fig. 3 into a symbolic STSP form. From this point, an equivalent of the symbolic STSP matrices is available for C code generation, which can be used in SystemVerilog via Direct Programming Interface (DPI).

```
%Generate a .m function from symbolic variables
matlabFunction(A,B,C,D,'file','FiltDiscrete_CT');
%Custom code generation of a wrapping function
varlist_base= string(symvar([A B;C D]));
varlist= ["ts" varlist_base];
fID= fopen(['FiltDiscrete' '.m'],'w');
fprintf(fID, 'function [Ad,Bd,Cd,Dd]= %s(%s)\n',
    'FiltDiscrete',join(varlist,","));
fprintf(fID, ' [A,B,C,D]= %s(%s);\n',
    'FiltDiscrete_CT',join(varlist_base,","));
fprintf(fID, ' Ad= expm(A*ts); Bd=
    (A^-1)*(Ad-eye(size(A)))*B; Cd= C; Dd=
    D;\nend');
fclose(fID);
%DPI coder call
cfg= coder.config('dll', 'ecoder', true);
cfg.FilePartitionMethod= 'SingleFile';
cfg.Toolchain= 'Cadence Xcelium (64-bit Linux)';
cfg.MultiInstanceCode= true;
cfg.BuildConfiguration= 'Faster Builds';
dpigen('FiltDiscrete','-config',cfg,'-args',...
   num2cell(NaN*ones(1,numel(varlist))));
```

Listing 1. Wrapping and DPI-C generation

The generation steps are performed in the MATLAB® environment and are reported in Listing 1. As a first stage, symbolic variables are reconverted in a MATLAB® function. This removes the symbolic representation, maintaining the reconfigurability. Subsequently, a custom wrapping function is generated. It allows performing the discretization for which a new variable for sampling time must be included. This wrapping is needed to avoid symbolical exponentiation and inversion required by the selected ZOH. Finally, the DPI-C code generation is executed. A special mention is needed on the *-args* value of *dpigen* command. All variables in *varlist* are set to a dummy floating-point number to force the data type for the generation: this is selected to NaN. Once the DPI-C files are available, they are imported in the SystemVerilog testbench.

```
import FiltDiscrete_dpi_pkg::*;
initial begin
   objhandle=
      DPI_FiltDiscrete_initialize(objhandle);
   //suppressed: components value generation
   DPI_FiltDiscrete(objhandle,ts,C_damp_m,
      C_damp_p,C_reson_m,C_reson_p,C_snubber_m,
      C_snubber_p,L_reson_m,L_reson_p,R_damp_m,
      R_damp_p,R_load,R_on_avg_m,R_on_avg_p,
      R_reson_m,R_reson_p,R_snubber_m,
      R_snubber_p,Ad,Bd,Cd,Dd);
   //suppressed: DT State-Space driver
end
```

Listing 2. Import and usage in SystemVerilog

Listing 2 reports the usage of DPI-C code in SystemVerilog.

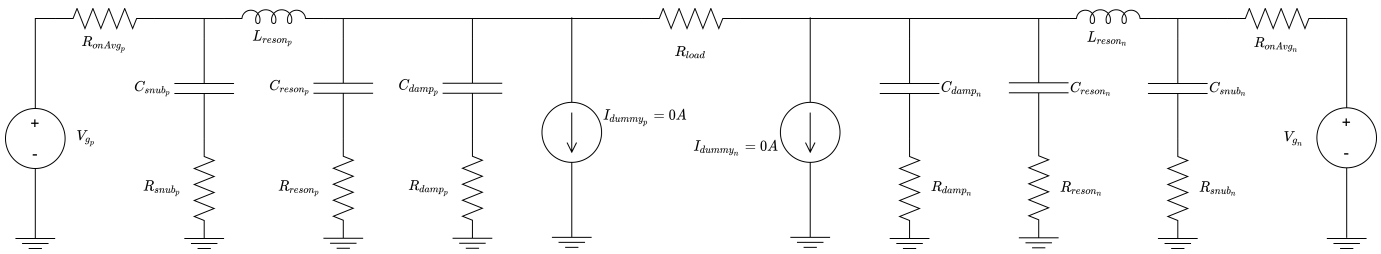

Fig. 3. Differential audio LC filter

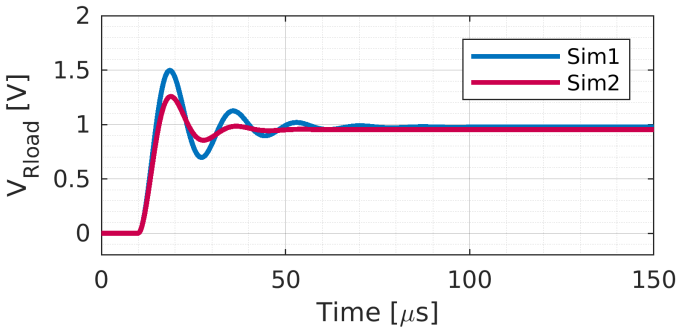

Fig. 4. DT simulation - step responses

Once the package *FiltDiscrete_dpi_pkg* is imported, it can be used in the *initial* section, where the first required step is an *initialize* call.

Component values generation can be performed via constrained randomization of a dedicated SystemVerilog object. Once the components values are available, the discretization can be performed via *DPI_FiltDiscrete*. The result is the DT STSP numerical matrices that are passed as a configuration argument to a dedicated driver implementing the (2a) and (2b).

To validate the strategy, the DT version of the system shown in Fig. 3 has been simulated in the Cadence® environment, focusing on the differential mode response of the MIMO system, which is converted into a SISO system directly within the SystemVerilog testbench. The two voltage source inputs are controlled by a single *real* value resembling a voltage V_{in}: the left side generator is driven by $0.5V_{in}$, while the right side is driven by $-0.5V_{in}$. The two voltage outputs are used to calculate, directly in the testbench, the *real* difference named V_{load} across the R_{load}. In Fig. 4, it is reported the V_{load} response to a $1V$ V_{in} step applied at $10\mu s$ in two different simulations where the R_{load} has been changed: $R_{load_1} < R_{load_2}$ where R_{load_1} and R_{load_2} are related to the first and the second simulation, respectively. As expected, the resulting V_{load} dynamic is damped more in the second simulation.

V. Digital LTI components within analog simulation environments

After the system architecture definition (the first step in the IC design process) the analog and digital design activities can start. Mixed-signal simulations are conducted throughout the analog design process. These simulations are essential for validating intermediate releases of analog blocks, ensuring that they function correctly within the overall system. A methodology is proposed that enables analog design activities to be conducted independently from the completion of the digital LTI parts within an analog simulation environment that supports Verilog-A. The LC filter in Fig. 3 can be considered representative of a function to be compensated. This circuit is imported in the MATLAB® environment via Spice2StateSpace, as in IV. The obtained MIMO STSP model is converted into SISO form by analyzing the input-to-output differential mode. After substituting the component values, it is possible to start with the compensator design that unfolds in a frequency equalization in the band of interest BW, which is considered from $0Hz$ up to $100kHz$. It is recognized that the dominant state variables in the band of interest BW are associated with the LC resonators, which are halved by the SISO transformation; this results in a size value of 2 for the state. The order reduction is executed using the *balred* function with a maximum absolute error target of $0.01dB$ within the band of interest. The pole-zero excess, defined as the difference between the number of poles and zeros, is non-zero. To render the CT system invertible, it is necessary to add sufficient zeros to counterbalance the pole-zero excess. These zeros can be placed at a frequency beyond the bandwidth of interest. Subsequently, discretization is performed using the zero-pole matching method [19]. This method is selected because equalization is required, and the main interest is in the cancellation of the analog CT filter with the zeros in the digital DT compensator.

```
analog begin
   @(initial_step) begin
      xp[1]=0.0; xp[2]=0.0; y[1]=0.0;
   end
   @(cross(V(clk)-vth,1)) begin
      u[1] =V(input1);
      y[1] =c11*xp[1]+c12*xp[2]+d11*u[1];
      xn[1]=a11*xp[1]+a12*xp[2]+b11*u[1];
      xn[2]=a21*xp[1];
      xp[1]=xn[1]; xp[2]=xn[2];
   end
   V(output1)<+y[1];
end
```

Listing 3. Verilog-A code generated

In this work, a custom MATLAB® generator has been developed to produce the Verilog-A code of (2a) and (2b), facilitat-

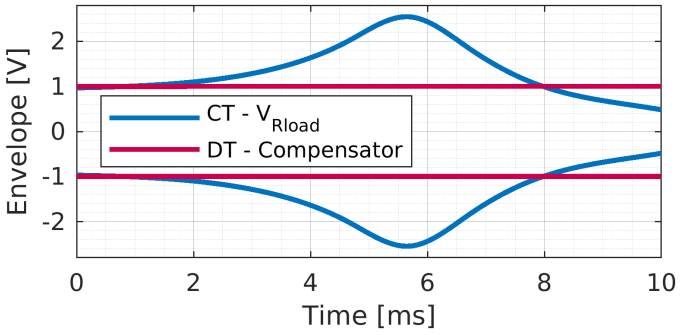

Fig. 5. CT simulation - chirp response compensation

ing the implementation of DT STSP in analog environments. In Listing 3, a portion of the generated code is reported. The *initial_step* directive is used to set the initial states values xp (representing the present state) and the intermediate output y. The STSP coefficients were previously defined as real numbers. When a positive edge of the voltage clock signal clk crosses the threshold vth, the main process is activated. Voltage input is loaded in u, then y and the next state xn are calculated by means of xp and u. At the end, xp is updated with the value calculated in xn, making xp ready for the next clk positive edge. Some coefficients in STSP are often null, which is related to the topological connections (e.g., some states in (2a) are not directly connected with the inputs). The code generator recognizes this phenomenon and automatically removes the unused variables, as done in Listing 3, where $a22$ and $b21$ are suppressed. The purpose of this optimization is to reduce the number of multiply-and-accumulate in high dimension STSP.

To test the developed methodology, the cascade of the CT circuit described in Fig. 3 and the Verilog-A DT STSP compensator has been simulated in the Cadence® environment, using a chirp input signal. This signal has an amplitude of $1V$ and frequency sweeps from $1kHz$ to $100kHz$ in $10ms$. The result is reported in Fig. 5, where the LC output envelope is completely flattened at the DT compensator output.

VI. Conclusions

A fully automated technique for symbolic STSP generation, founded on an algorithmic extension of the MNA, is presented. Univocal definition of circuit outputs directly on the schematic is a key result since it requires user intervention only during the schematic definition. The problem of model parameters reconfigurability in the context of event-driven simulation is tackled. An extensive digital verification activity in multiple scenarios is enabled by a completely automatic solution for model export and integration within SystemVerilog testbenches. The case of a filter for audio applications is used as an example for compensator design, and a methodology is proposed to generate Verilog-A code for DT STSP models, allowing the simulation of LTI digital portions of mixed-signal systems in analog environments that support Verilog-A.

References

[1] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 5th edition, 2010.

[2] Tomas Martinez-Marin. State-space formulation for circuit analysis. *IEEE Transactions on Education*, 53(3):497–503, 2010.

[3] Chung-Cheng Chen, Jian Ke, and Yen-Ting Chen. New method of state-space formulation for degenerate circuit and coupling circuit. *Mathematical Problems in Engineering*, 2019:1–13, 06 2019.

[4] Shera Elvis. State-space "switching" model of dc-dc converters in systemverilog. In *DVCon Europe 2017; Design and Verification Conference and Exhibition Europe*, 2017.

[5] G.G.E. Gielen, H.C.C. Walscharts, and W.M.C. Sansen. Isaac: a symbolic simulator for analog integrated circuits. *IEEE Journal of Solid-State Circuits*, 24(6):1587–1597, 1989.

[6] K. Singhal and J. Vlach. Symbolic analysis of analog and digital circuits. *IEEE Transactions on Circuits and Systems*, 24(11):598–609, 1977.

[7] G. Alderson and P. Lin. Computer generation of symbolic network functions-a new theory and implementation. *IEEE Transactions on Circuit Theory*, 20(1):48–56, 1973.

[8] James W. Nilsson and Susan A. Riedel. *Electric circuits*. Prentice Hall, 9th edition, 2011.

[9] A. Brambilla, A. Premoli, and G. Storti-Gajani. Recasting modified nodal analysis to improve reliability in numerical circuit simulation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(3):522–534, 2005.

[10] R.K. Brayton, G.D. Hachtel, and A.L. Sangiovanni-Vincentelli. A survey of optimization techniques for integrated-circuit design. *Proceedings of the IEEE*, 69(10):1334–1362, 1981.

[11] J. Bandler, P. Liu, and H. Tromp. A nonlinear programming approach to optimal design centering, tolerancing, and tuning. *IEEE Transactions on Circuits and Systems*, 23(3):155–165, 1976.

[12] Seobin Jung, Yunju Choi, and Jaeha Kim. Variability-aware, discrete optimization for analog circuits. In *DAC Design Automation Conference 2012*, pages 536–541, 2012.

[13] G. Hachtel, T. Scott, and R. Zug. An interactive linear programming approach to model parameter fitting and worst case circuit design. *IEEE Transactions on Circuits and Systems*, 27(10):871–881, 1980.

[14] robstab: Robust stability of uncertain system. https://it.mathworks.com/help/robust/ref/dynamicsystem.robstab.html. Accessed: 2024-04-16.

[15] Scam: Symbolic circuit analysis in matlab. https://lpsa.swarthmore.edu/Systems/Electrical/mna/MNA6.html. Accessed: 2024-04-16.

[16] netlist2linss. https://github.com/bencholmes/netlist2linss. Accessed: 2024-04-16.

[17] L. Odess and H. Ur. Nullor equivalent networks of nonideal operational amplifiers and voltage-controlled sources. *IEEE Transactions on Circuits and Systems*, 27(3):231–235, 1980.

[18] Boguslaw Szlachetko and Zbigniew Swietach. Analysis of continuous-to-discrete transformation effect on a synthesis filter bank project. In *2018 22nd International Microwave and Radar Conference (MIKON)*, pages 276–279, 2018.

[19] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 1998.

[20] Matteo De Ferrari, Francesco Stilgenbauer, Edoardo Botti, Cristiano Meroni, Edoardo Bonizzoni, and Piero Malcovati. A mixed analog-digital class-d amplifier with third-order loop filter for audio applications. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2023.