Simulation of Ideally Switched Circuits in SystemC

Breytner Joseph Fernández-Mesa
Liliana Andrade
Frédéric Pétrot

UGA Université Grenoble Alpes
iima
CARS
Outline

1. Interest of Switched Circuit Simulation at the System Level
2. Requirements for the Simulation of Switched Circuits
3. Implementation in SystemC, our Contribution.
4. Case Study: Comparison with Industrial Tools
Outline

1. Interest of Switched Circuit Simulation at the System Level
2. Requirements for the Simulation of Switched Circuits
3. Implementation in SystemC, our Contribution.
4. Case Study: Comparison with Industrial Tools
Virtual Prototyping in SystemC

- **SystemC**: digital hardware.
- **Transaction Level Modeling (TLM)**: virtual prototypes.
- **Analog and Mixed Signals (AMS)**: continuous time.
- Example of multi-domain simulation: software + digital hardware + continuous time.
- One domain of recent interest: power.

**Figure** — Model of a Electronic Transmission Control Unit. (Pan et al. 2019)
Virtual Prototyping in SystemC

- SystemC: digital hardware.
- Transaction Level Modeling (TLM): virtual prototypes.
- Analog and Mixed Signals (AMS): continuous time.
- Example of multi-domain simulation: software + digital hardware + continuous time.
- One domain of recent interest: power.

**Figure** — Model of a Electronic Transmission Control Unit. (Pan et al. 2019)
Virtual Prototyping in SystemC

- **SystemC**: digital hardware.
- **Transaction Level Modeling (TLM)**: virtual prototypes.
- **Analog and Mixed Signals (AMS)**: continuous time.
- Example of multi-domain simulation: software + digital hardware + continuous time.
- One domain of recent interest: power.

**Figure** — Model of an Electronic Transmission Control Unit. (Pan et al. 2019)
Description:

- Powertrain virtual prototyping framework.
- Goal: increase energy lifespan to drive longer distances.
- Mean: simulation and architecture exploration.
Applications of Virtual Prototypes of Power System Models

Limitations:

- No diode primitives in SystemC AMS.
- Converter modeled as look-up tables.
- But limit architecture exploration.

**Figure** — Architecture of the powertrain of an electronic vehicle. (Chen et al, 2019)
Examples of Virtual Prototyping in SystemC

Remark
SystemC supports virtual prototyping but not power systems. Our goal: their accurate and efficient simulation.
1. Interest of Switched Circuit Simulation at the System Level

2. Requirements for the Simulation of Switched Circuits

3. Implementation in SystemC, our Contribution.

4. Case Study: Comparison with Industrial Tools
Power System Examples

- **Goal:** convert energy.
- **Example:** one voltage to a greater voltage.
- **Mean:** switched circuits.
- **Two kind of switches:** externally controlled and diodes.

**Figure – Examples of power converters**

1. **Buck**
2. **Buck-boost**
3. **Boost**
4. **Cuk**
Levels of Modeling Power Systems

MOSFET:
- Device geometry and physical phenomena.
- Very complex, 43 parameters in SPICE.
- Solved in CT.

Variable resistance:
- ON, very low resistance. OFF, very high resistance.
- Simpler but result in ill-conditioned equations.
- Solved in CT.

Ideal:
- Two discrete states ON and OFF.
- Piece-wise linear equations.
- CT/DE synchronization.
Levels of Modeling Power Systems

MOSFET:
- Device geometry and physical phenomena.
- Very complex, 43 parameters in SPICE.
- Solved in CT.

Variable resistance:
- ON, very low resistance. OFF, very high resistance.
- Simpler but result in ill-conditioned equations.
- Solved in CT.

Ideal:
- Two discrete states ON and OFF.
- Piece-wise linear equations.
- CT/DE synchronization.
Levels of Modeling Power Systems

MOSFET:
- Device geometry and physical phenomena.
- Very complex, 43 parameters in SPICE.
- Solved in CT.

Variable resistance:
- ON, very low resistance. OFF, very high resistance.
- Simpler but result in ill-conditioned equations.
- Solved in CT.

Ideal:
- Two discrete states ON and OFF.
- Piece-wise linear equations.
- CT/DE synchronization.
Simulation Requirements

$t < (t_1, 0)$

Figure – Buck converter simulation

- The switch has remained open for a long time.
- We close the switch.
- The inductor develops a current.
- Diode off because it is polarized negatively.

January 2021 Fernández-Mesa, Andrade, Pétrot/ TIMA Laboratory
Simulation Requirements

\[ t = (t_1, 0) \]

- The switch has remained open for a long time.
- We close the switch.
- The inductor develops a current.
- Diode off because it is polarized negatively.

**Figure** – Buck converter simulation
The switch has remained open for a long time.

We close the switch.

The inductor develops a current.

Diode off because it is polarized negatively.
An external event opens the switch: current should be zero.

Impulsive voltage to preserve current.

Impulsive voltage activates the diode, current passes.

The inductor preserves its current flow.

\[ t = (t_2, 1) \]
Simulation Requirements

\[ t = (t_2, 2) \]

- An external event opens the switch: current should be zero.
- Impulsive voltage to preserve current.
- Impulsive voltage activates the diode, current passes.
- The inductor preserves its current flow.

**Figure** – Buck converter simulation
Simulation Requirements

\[ t = (t_3, 0) \]

- Simulation continues, energy is dissipated by the resistance.
- The simulator calculates a negative current, by definition the diode should turn off.
- The simulator detects and locates this event.
- Simulation continues with the diode off.

**Figure** – Buck converter simulation
Simulation Requirements

\[ t = (t_4, 0) \]

Simulation continues, energy is dissipated by the resistance.

- The simulator calculates a negative current, by definition the diode should turn off.
- The simulator detects and locates this event.
- Simulation continues with the diode off.

**Figure** – Buck converter simulation
Simulation Requirements

\[ t = (t_5, 0) \]

- Simulation continues, energy is dissipated by the resistance.
- The simulator calculates a negative current, by definition the diode should turn off.
- The simulator detects and locates this event.
- Simulation continues with the diode off.

**Figure – Buck converter simulation**
Simulation Requirements

\[ t = (t_5, 0) \]

Simulation continues, energy is dissipated by the resistance.

- The simulator calculates a negative current, by definition the diode should turn off.
- The simulator detects and locates this event.
- Simulation continues with the diode off.

**Figure** – Buck converter simulation
Simulation Requirements

\[ t = (t_5, 0) \]

**Figure** – Buck converter simulation

- Simulation continues, energy is dissipated by the resistance.
- The simulator calculates a negative current, by definition the diode should turn off.
- The simulator detects and locates this event.
- Simulation continues with the diode off.
Simulation Requirements

\[ t = (t_6, 0) \]

- Simulation continues, energy is dissipated by the resistance.

- The simulator calculates a negative current, by definition the diode should turn off.

- The simulator detects and locates this event.

- Simulation continues with the diode off.

**Figure** – Buck converter simulation
Requirements for the Simulation of Ideally Switched Circuits

Requirements
Requirements for the Simulation of Ideally Switched Circuits

\[ t = (t_2, 1) \]

- CT modules that model the circuit must react timely to external events.
- The simulator must be able to advance time by zero steps.
- The simulator must monitor the circuit variables for internal events and locate the time of these events.

**Figure – Buck converter simulation**
Requirements for the Simulation of Ideally Switched Circuits

\[
t = (t_2, 2)
\]

- CT modules that model the circuit must react timely to external events.
- The simulator must be able to advance time by zero steps.
- The simulator must monitor the circuit variables for internal events and locate the time of these events.

**Figure – Buck converter simulation**
Requirements for the Simulation of Ideally Switched Circuits

\[ t = (t_5, 0) \]

CT modules that model the circuit must react timely to external events.

The simulator must be able to advance time by zero steps.

The simulator must monitor the circuit variables for internal events and locate the time of these events.

**Figure** – Buck converter simulation

Event detection: zero crossing
## Current Solutions

<table>
<thead>
<tr>
<th>Tool</th>
<th>Switch Model</th>
<th>Event Detection</th>
<th>Virtual Prototyping</th>
<th>Standard SystemC</th>
<th>Code Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPICE</td>
<td>MOSFET</td>
<td>--</td>
<td>No</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Ideal + Snubber circuits</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>SystemC AMS (ELN)</td>
<td>Variable resistance, no diode</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SystemC AMS (EPN)</td>
<td>Ideal</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SystemC A</td>
<td>--</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>VHDL-AMS</td>
<td>Ideal (equations)</td>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Verilog-AMS</td>
<td>Ideal (equations)</td>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>PLECS</td>
<td>Ideal</td>
<td>Yes</td>
<td>No, but Processor in the Loop</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Our solution</strong></td>
<td>Ideal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Figure** – Comparison of simulators
Outline

1 Interest of Switched Circuit Simulation at the System Level

2 Requirements for the Simulation of Switched Circuits

3 Implementation in SystemC, our Contribution.

4 Case Study: Comparison with Industrial Tools
1. Modeling: user connects primitives
2. Elaboration: simulator gets the equations
3. Simulation: simulator solves the equations
Our Solution

1. Modeling: user connects primitives
2. Elaboration: simulator gets the equations
3. Simulation: simulator solves the equations
• Primitive: basic modeling elements.

• Two types:
  - Computation
  - Communication

Figure – Simulation primitives
Modeling: the designer interconnects primitives

```cpp
class Buck: public ISC::cluster {
    // ...
    private:
    ISC::v_source src_1;
    ISC::switch_t switch_1;
    ISC::inductor ind_1;
    // etc.
};

Buck::Buck(sc_module_name name, /* other parameters */) :
    src_1("src_1", -V), // Give names,
    switch_1("switch_1"), // and values
    ind_1("ind_1", L1),
    // ... Initialize the rest of primitives
{
    src_1.terminal_a(ground_node);
    src_1.terminal_b(node_1);
    switch_1.terminal_a(node_2);
    switch_1.terminal_b(node_3);
    // ... connect the rest of primitives /
}
```

**Figure** – Example model of a converter
Our Solution

1. **Modeling**: user connects primitives
2. **Elaboration**: simulator gets the equations
3. **Simulation**: simulator solves the equations
Elaboration:

Goal: from the circuit, obtain a state space representation.

\[ x = \begin{bmatrix} i_L \\ v_C \end{bmatrix}, \quad y = \begin{bmatrix} v_D \\ i_D \end{bmatrix} \]

\[ \dot{x} = Ax + Bu \]

\[ y = Cx + Du \]
Elaboration:

Each topology has a state space representation.

\[
A = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{-1}{L} \\ 0 \end{bmatrix}
\]

\[
C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -1 \\ 0 \end{bmatrix}
\]
Each topology has a state space representation.

\[
A = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

\[
C = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
Use SystemC’s hierarchy traversal functions to get the list of nodes and elements.

Use the algorithm presented by Massarini, A. & Reggiani, U. (2001) to get a reduced tableau representation* based on a sequence of matrix operations.

* Refer to this article for the notation and procedure details.

In general:

\[
\begin{bmatrix}
S \\
0 & 1 & -C_1 & -C & \cdots & -D \\
0 & 0 & M^{(0)} & -A^{(0)} & \cdots & -B^{(0)}
\end{bmatrix} \begin{bmatrix} s \\ y \\ x \\ 0 \\ u \end{bmatrix} = 0
\]
Elaboration: Advantages of the Reduced Tableau Representation

Obtain the state space directly: \( A = M^{(0)-1}A^{(0)} \) and \( B = M^{(0)-1}B^{(0)} \).

\[ \dot{x} = Ax + Bu \]
\[ y = Cx + Du \]

Calculate the tableau for only one topology.

Derive for the rest of topologies by simple swap operations.
Elaboration: Advantages of the Reduced Tableau Representation

- Obtain the state space directly: $A = M^{(0)-1}A^{(0)}$ and $B = M^{(0)-1}B^{(0)}$.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- Calculate the tableau for only one topology.

- Derive for the rest of topologies by simple swap operations.

**Figure** – Switch toggling implies only swap operations on the reduced tableau.
Our Solution

1. Modeling: user connects primitives
2. Elaboration: simulator gets the equations
3. Simulation: simulator solves the equations
Simulation

Execution Phases

- **Suspension**
  - $t_{DG} = t_{CT} \land \neg \text{init}$
- **Output generation**
  - $t_{DG} < t_{CT}$
- **Back-tracking**
  - $t_{CT} \leq t_{DG}$
- **Checkpoint creation**
  - $t_{CT} < t_{DG}$
- **Catching Up**
  - $t_{CT} = t_{DG}$
- **Detection of events at $t_{DE}$**
  - $t_{CT} < t_{end}$
  - $t_{end} = t_{DE}$

CT MoC Interface

- **GET-DERIVATIVES**
  - $t_{DE}$: DE global sim. time
  - $t_{CT}$: CT sim. time
  - $\text{init}$: DE kernel initialization
  - $\text{ie}$: input event
- **IS-EVENT**
  - $|t_s - t_{CT}| > \varepsilon$
- **GENERATE-OUTPUTS**
  - Updated: CT model/state updated by input event
  - $t$: time of state event
  - $\varepsilon$: allowed numerical tolerance for finding time of state events
- **EXECUTE-UPDATES**
  - $t_{end}$: current CT execution end time
  - $\text{MoC interface function call}$

**Figure** – Direct CT/DE Synchronization presented at DATE’20 by us (Fernandez-Mesa et al, 2020).
It exposes an interface for integrating the CT modules.

- It consists of four functions.

**figure** — Fernandez-Mesa et al. (2020). Direct CT/DE Synchronization.
(1) **GET-DERIVATIVES**: return the derivative values.

In our case:

- return $\dot{x} = Ax + Bu$
(2) **Is-Event**: return true if there is an internal event and false otherwise.

In our case:

- Set diode currents and voltages as outputs ($y$)
- Calculate $y = Cx + Du$
- Return $y \leq 0$

**Figure** – Fernandez-Mesa et al. (2020). Direct CT/DE Synchronization.
(3) **GENERATE-OUTPUTS**: write the output ports.

- In our case:
  - Calculate $y = Cx + Du$
  - Write the ports.

---

**Figure** — Fernandez-Mesa et al. (2020). Direct CT/DE Synchronization.
(4) **EXECUTE-UPDATES**: update the CT model based on the external events.

In our case:

- **swap the columns** of the switches toggled by external events.
Our Solution

1. Modeling: user connects primitives
2. Elaboration: simulator gets the equations
3. Simulation: simulator solves the equations
Outline

1. Interest of Switched Circuit Simulation at the System Level
2. Requirements for the Simulation of Switched Circuits
3. Implementation in SystemC, our Contribution.
4. Case Study: Comparison with Industrial Tools
Case Study: SEPIC Converter

Model of a Single Ended Primary Inductor Converter (SEPIC).

External switch controlled by a discrete event pulse generator working at 100kHz with 50% duty cycle.

Figure – SEPIC model
Modeled and simulated in NGSPICE, MATLAB, PLECS, and in our simulator ISC: Ideally Switched Circuits in SystemC.

Similar accuracy as industry tools when studying system behavior: we omit the short-term switching phenomena.
## Case Study: SEPIC Converter

**Figure** – SEPIC simulation results: speed-up

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Statistics</th>
<th>Tool</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SEPI</td>
<td>Wall-Clock time (s)</td>
<td>NGSPICE</td>
<td>8.48</td>
<td>3.08</td>
</tr>
<tr>
<td>(0.04 s)</td>
<td>Speed-Up</td>
<td>MATLAB</td>
<td>0.0390</td>
<td>0.107</td>
</tr>
</tbody>
</table>

- Simulation speed of the same order of magnitude as the reference industry-level simulator.
- Speed-up gains from ideal switches, reduced tableau representation, and an appropriate CT/DE synchronization.

**Remark**

Speed and accuracy appropriate for system level simulation.
Conclusion

- Power system simulation is **implementable** in SystemC.
- Ideal switched circuits offer **sufficient accuracy and speed** for system level simulation.
- This can enable **architectural exploration**.
- Our solution enriches the SystemC **multi-domain simulation ecosystem**: Software + digital hardware + continuous time.
- **Available as open source at:**
  https://gricad-gitlab.univ-grenoble-alpes.fr/tima/sls/projects/systemc-ct
COSEDA Technologies GmbH — SystemC AMS Proof-of-Concept. 

Analogue and Mixed-Signal Extension to SystemC.

PLECS-Piece-Wise Linear Electrical Circuit Simulation for Simulink.
Advancing the SystemC Analog/Mixed-Signal (AMS) Extensions.  
Open SystemC Initiative.

SystemC AMS Extensions User’s Guide.  
Accellera systems initiative.

Open SystemC Initiative (OSCI).

Time-Domain Analysis of Networks with Internally Controlled Switches.  


IEEE Computer Society.

IEEE Computer Society.


IEEE.


Vhdl (vhsic (very high speed integrated circuits) hardware descriptive languages) prototype simulator. Technical report, Air Force Institute of Technology, Wright-Patterson Air Force Base School of Engineering.


Prom timed to hybrid systems. In Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems), pages 447–484. Springer.
A New Representation of Dirac Impulses in Time-Domain Computer Analysis of Networks with Ideal Switches.

Computer-Aided Time-Domain Large-Signal Analysis of Networks with Switches.

An Efficient Algorithm for the Formulation of State Equations and Output Equations for Networks with Ideal Switches.


Specialized Power Systems - MATLAB & Simulink.

SystemC-AMS Requirements, Design Objectives and Rationale.
In 2003 Design, Automation & Test in Europe (DATE), pages 388–393. IEEE.

Ngspice user’ manual version 32.
Simulation of Conventional vs. Ideal Switches: Diode Rectifier

Figure – Comparison of simulation when using conventional vs. ideal switches. Taken from "An introduction to solvers," 2011 IEEE Vehicle Power and Propulsion Conference, Chicago, IL, 2011, pp. 1-132

- Simulation steps: 1160 vs. 153.
- Wall-clock time: 0.6 s vs. 0.08 s.