SwitchSim: A SPICE-like Simulator for Switching Circuits Based on State-driven Unified Branch Stamps

Ze Hao[1], Jianwang Zhai[1], Ding Gong[3], Kang Zhao[1,†], Wenjian Yu[2,†]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing, China
[3]Peifeng Tunan Semiconductor Co. Ltd., Suzhou, China

***Abstract*—With the increasing scale and complexity of power electronic systems, circuit simulators capable of accurately capturing the switching behavior of devices such as IGBTs have become essential. However, existing dedicated simulators for switching circuits do not natively support the widely adopted SPICE netlist infrastructure, whereas general-purpose SPICE simulators often encounter numerical difficulties, such as singular matrices, when handling near-ideal switching models. To address this limitation, drawing on the electromagnetic transient simulation paradigm used in power system analysis, we propose *SwitchSim*, a SPICE-like simulator for switching circuits. The proposed simulator employs a state-parameterized unified branch stamp model to represent switching devices in an idealized form, and further introduces a time-stepping controller that jointly evaluates the step-to-step solution variation residual and the device-state consistency residual to enable valid and efficient transient advancement. Experimental results show that, using the commercial simulation software PLECS as the reference, SwitchSim achieves accuracy comparable to or better than the general-purpose circuit simulator LTspice and the dedicated ideal-switch simulation framework Siconos across the evaluated PLECS-solvable benchmarks. Moreover, SwitchSim successfully completes complex switching-circuit simulations for which PLECS fails to obtain a solution.**

***Index Terms*—Circuit Simulation, Switching Circuits, Modified Nodal Analysis**

I. Introduction

As integrated circuits continue to scale in size and complexity, circuit simulation has become an indispensable computer-aided tool, which formulates equations satisfying Kirchhoff's laws from circuit topology, device constitutive relations, and external excitations, and numerically solves circuit quantities such as nodal voltages and branch currents. Two major formulations commonly used in circuit simulation are Modified Nodal Analysis (MNA) and the State-Variable Method. MNA extends Nodal Analysis by taking nodal voltages as the primary unknowns and introducing the additional branch currents, whose resulting systems are typically differential-algebraic equations (DAEs) [1]. SPICE, the most widely used family of tools in circuit simulation, is built on the MNA framework [2], [3]. By contrast, the state-variable method typically selects independent state variables, such as capacitor voltages, to construct state-space equations, and the resulting systems are usually explicit ordinary differential equations (ODEs) [4]. Traditional state-variable modeling requires topology-specific state-variable selection and processing, which complicates its application to complex circuits [5].

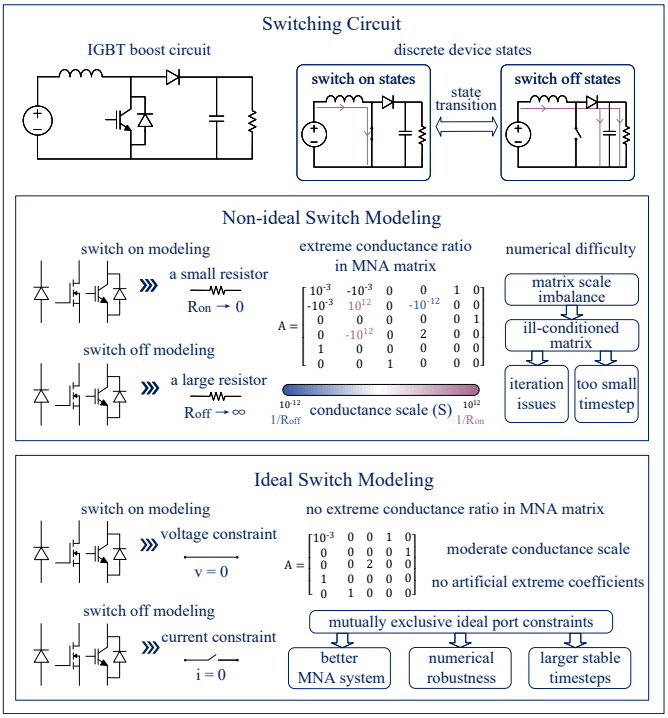

Fig. 1 Depending on the application context, devices such as IGBTs can be modeled either as continuous nonlinear devices or as switching devices; the latter can be further represented by either non-ideal or ideal switch models.

Although a mature methodology has been established for general-purpose circuit simulation, simulation of switching circuits remains challenging. Switching circuits are dynamic systems built around switching devices, such as diodes, IGBTs, and MOSFETs, whose circuit topology changes through discrete transitions between on and off states. They are widely used in power-electronic systems, including converters, rectifiers, and inverters. In such circuits, simulation generally focuses on the equivalent on/off characteristics of switching devices rather than their detailed internal physical processes [6]. As shown in Fig. 1, general-purpose SPICE

†Corresponding authors.

This work is supported by the National Natural Science Foundation of China (No. 62404021), the Beijing Natural Science Foundation (No. L2603021), the Fundamental Research Funds for the BUPT (No. 2025AI4S20), and the Research Initiation Project for Introduced Talents of BUPT (No. 510124058, 510224062).

simulators commonly approximate switches using a small on-state resistance and a large off-state resistance. However, the resulting extreme resistance ratios can lead to ill-conditioned matrices, convergence failures, or excessively small time steps [7]. In contrast, ideal switch modeling represents the on state by a zero-voltage constraint and the off state by a zero-current constraint, avoiding such artificial numerical stiffness. PLECS, a dedicated switching simulator, directly supports ideal switches by automatically generating piecewise-linear state-space models associated with different switching states [8], [9]. However, these models are topology dependent and fundamentally differ from the unified MNA formulation and netlist-based representation adopted by SPICE.

In addition to commercial tools such as PLECS, some studies and open-source tools have also focused on ideal switching circuit simulation. A representative approach is based on nonsmooth dynamical systems (NSDS), where the on/off behavior of switching devices is formulated as complementarity constraints, variational inequalities, or differential inclusions [10], [11], which can be implemented in open-source platform Siconos [12]. NSDS provides a mathematical framework for ideal switching circuits, but its primary abstractions are dynamical systems and nonsmooth laws, rather than circuit devices and branch variables. Users must explicitly define the dynamical systems in programmatic form, making it difficult to integrate simulation into the mature netlist infrastructure. Another line of work investigates how ideal switches can be handled within the MNA framework. Yildiz et al. proposed a unified ideal switch model that uses a variable to describe the states of switches [7], [13]. This method has been successfully applied in switched capacitor circuits [14]. However, that work primarily focuses on switching actions, does not provide an end-to-end simulator architecture that integrates switching-state representation with device models. It therefore does not fully address the challenge of simulating ideal switching circuits within the MNA framework.

This work aims to develop an end-to-end MNA framework for ideal switching circuits, bridging the advantages of SPICE for circuit simulation with ideal switch modeling. On the one hand, the proposed framework retains SPICE-like netlists as input, allowing mature circuit descriptions, topology templates, and parameter-sweep scripts to be used directly. On the other hand, it reformulates the traditional SPICE solution paradigm, which is dominated by continuous semiconductor models and Newton iterations, into a transient simulator tailored to strictly ideal switching devices. Specifically, drawing on event monitoring and state-driven techniques in electromagnetic transient (EMT) analysis for large-scale power grids and power systems [15], this paper develops ideal constraints for common switching devices, and proposes an end-to-end state-driven simulator as an alternative to the Newton-iteration-based solution flow of traditional SPICE. The main contributions are as follows:

- We propose **SwitchSim**, a SPICE-like simulator that integrates MNA ideal switch modeling with transient solving through device-state validation.
- Building on the unified ideal switch model, we develop fixed-sparsity unified branch stamps for diodes, MOSFETs, and IGBTs using ideal port constraints while preserving device-specific switching semantics.
- We develop a dual-residual adaptive time-stepping (DRATS) controller, which jointly evaluates the solution variation residual (SVR) and the device-state consistency residual (DCR), to regulate the next step size and integration method for robust transient advancement.

II. Preliminaries

A. Switching Circuit Simulation

The core objective of circuit simulation is to solve for the voltage and current states within a circuit system. For a given circuit in an MNA-based simulation, we choose a reference node whose voltage is defined as 0 V and uses the voltages of all non-reference nodes as the primary unknowns. For voltage sources, current-controlled devices, and other devices whose branch currents cannot be expressed solely as functions of nodal voltages, we introduce the corresponding branch currents as additional unknowns, together with the branch equations associated with these unknowns. The result to be determined can be written as a combination of N nodal voltages and M additional branch currents:

$$x(t)=\begin{bmatrix}v(t)\\ i(t)\end{bmatrix}=\begin{bmatrix}v_1(t) & \cdots & v_N(t) & i_1(t) & \cdots & i_M(t)\end{bmatrix}^T, \tag{1}$$

where $v(t) \in \mathbb{R}^N$ denotes the voltages of the non-reference nodes, and $i(t) \in \mathbb{R}^M$ denotes the introduced additional current variables. The MNA equations governing the circuit state can be written in the following DAE form

$$A(t)x(t) + C\dot{x}(t) = z(t), \tag{2}$$

where $A(t)$ denotes the algebraic relation matrix, which incorporates resistive relations, independent- and controlled-source constraints, and coupling constraints; C denotes the dynamic matrix associated with storage devices, containing the derivative terms introduced by capacitors and inductors; and $z(t)$ represents the independent sources and equivalent right-hand-side vector terms. The difference between switching circuit simulation and general-purpose circuit simulation lies in the way switching components are modeled. Consider an ideal switching device connected between ports p and n. Its voltage drop and branch current are defined as:

$$U_s = v_p - v_n, \qquad i_s : \ p \to n. \tag{3}$$

An ideal switch has two states. In the on state, the branch voltage is constrained to zero; in the off state, the branch current is constrained to zero:

$$\begin{cases} s=1 & : \quad U_s = 0 \\ s=0 & : \quad i_s = 0 \end{cases}. \tag{4}$$

B. Unified Ideal Switch Model

The unified ideal switch model is a switching device stamping approach. In a straightforward ideal switch model, the MNA matrix must be rebuilt whenever the switching state changes and the effective circuit topology is modified. In contrast, this approach uses a state parameter to encode the ideal switch state, allowing the switch to be stamped into fixed entries of the MNA matrix. When the switching state changes, the model only updates the numerical values at the corresponding matrix entries.

To represent both the on and off states with the same matrix structure, a state parameter $s \in \{0, 1\}$ and a dimension-balancing factor Z_0 are introduced, allowing the two constraints in Equation (4) to be written in a unified form as

$$s(v_p - v_n) + (1 - s)Z_0 i_s = 0, \tag{5}$$

where Z_0 is used only to make the voltage and current terms dimensionally consistent and does not represent the physical on-state resistance of the device. In the MNA formulation, i_s is introduced as an additional current unknown, and the corresponding local MNA stamp of the ideal switch can be expressed as

$$\begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ s & -s & \cdots & (1-s)Z_0 \end{bmatrix} \begin{bmatrix} v_p \\ v_n \\ \vdots \\ i_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{6}$$

where the first row represents the branch current i_s flows out of node p, the second row represents that i_s flowing into node n, and the last row is the unified state constraint in Equation (5). When the switching state changes, only the numerical coefficients in the last constraint row are updated, while the sparsity pattern of the matrix remains unchanged.

The state-parameterized unified branch stamp model proposed follows the core idea of the unified ideal switch model, but it does not simply regard devices as pure bidirectional switches controlled by current or voltage. We preserve the necessary conduction direction and reverse freewheeling path, then define the corresponding state-update rules to ensure complete circuit semantics and correct numerical solution.

C. Implicit Integration for Transient Simulations

In transient analysis, the continuous-time DAE arising from MNA must be discretized in time by numerical integration, yielding an algebraic system to be solved at each time point. Two commonly used implicit integration schemes are the Backward Euler (BE) and the trapezoidal integration. Let the k-th time step be defined as:

$$h_k = t_k - t_{k-1}. \tag{7}$$

Let $x^k = x(t_k)$. The BE approximates the derivative term at t_k as:

$$\dot{x}(t_k) \approx \frac{x^k - x^{k-1}}{h_k}. \tag{8}$$

Substituting this approximation into Equation (2) yields the effective algebraic equation at the current time step:

$$\left(A(t_k) + \frac{C}{h_k}\right) x^k = z^k + \frac{C}{h_k} x^{k-1}. \tag{9}$$

Backward Euler is a first-order method, whose global error scales as $O(h)$; therefore, its accuracy may be insufficient when higher temporal resolution is required. To achieve higher accuracy, second-order schemes, whose global error scales as $O(h^2)$, are widely used. The trapezoidal integration is a representative example. Its basic idea is to approximate the time integral over one step using trapezoidal quadrature, and the effective algebraic equation at t_k can be written as:

$$\left(A(t_k) + \frac{2C}{h_k}\right) x^k = z^k + z^{k-1} + \left(\frac{2C}{h_k} - A(t_{k-1})\right) x^{k-1}. \tag{10}$$

Trapezoidal integration provides higher accuracy, but it introduces little numerical damping. Numerical damping refers to the intrinsic attenuation imposed by the time-integration scheme on high-frequency numerical components. By contrast, BE is strongly numerically dissipative.

In switching-circuits simulation, switching events can introduce high-frequency and discontinuity-like numerical components at the discrete-time level. If the integration scheme provides insufficient damping, these components may manifest as nonphysical numerical oscillations [16]. To address this issue, we propose the DRATS controller, which dynamically selects the numerical integration method based on switching events and the circuit state. It introduces only the minimum number of BE steps required to provide sufficient numerical damping, thereby suppressing nonphysical oscillations while avoiding excessive damping in smooth regions, which improves the overall accuracy of the simulation.

III. Proposed SwitchSim

A. State-driven SPICE-like Simulation Framework

This subsection describes the framework-level design of SwitchSim. Building on the conventional SPICE flow and tailored to the requirements of ideal-switch simulation, SwitchSim uses device-state validation to determine whether a trial time step is accepted. After an accepted step, the DRATS controller evaluates solution variation, device-state proximity, and switching events to schedule the next step size and integration mode, as shown in Fig. 2. The state-driven design provides the basis for implementing the state-parameterized unified branch stamp model, while DRATS regulates transient advancement within this framework.

In SwitchSim, each switching device is represented as a combination of ideal port constraints and state-update rules, replacing the non-ideal switch representation based on extreme resistance approximations. At a trial time step, all devices are stamped into the global MNA system according to their current provisional states. After solving the MNA system, the device states are updated from the resulting nodal voltages and branch currents. If any state changes, the system is re-stamped and re-solved until the device states

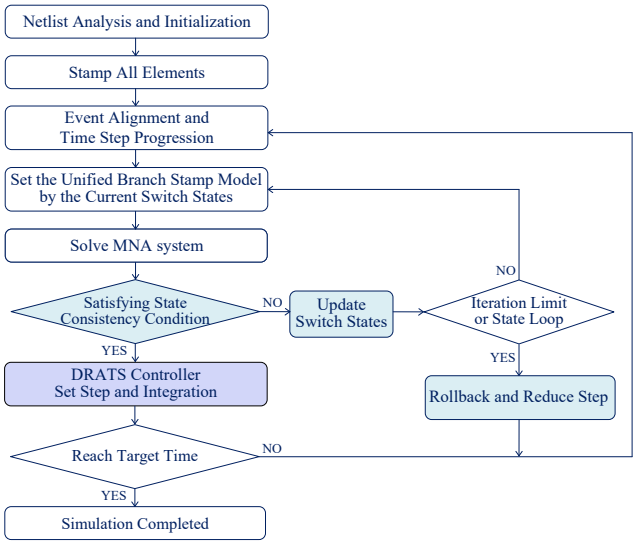

Fig. 2 Flowchart of switch circuit simulation. SwitchSim adopts a state-driven approach to advance time steps based on the traditional SPICE process and introduces the DRATS controller to ensure simulation progress.

become self-consistent. If the linear solve fails, a state cycle is detected, or the state-iteration limit is reached, the continuous and discrete states are rolled back and the time step is reduced for retry; otherwise, the current step is accepted.

The difficulty of simulating ideal switching circuits does not only lie in state determination, another important challenge is subsequent solution settings. SPICE simulators typically employ adaptive time-stepping mechanisms, allowing larger time steps in smooth regions and smaller time steps in regions with rapid dynamics. However, because ideal switching circuits impose device-state consistency requirements and involve explicit topology changes, traditional local truncation error alone is insufficient for making reliable decisions. Before each trial step, the time step is aligned with the nearest known switching event, and BE is used for event or recovery steps while trapezoidal integration is used in smooth regions. After the step is accepted, DRATS evaluates the SVR and DCR together with the observed state transition to determine the next time step and recovery mode, serving as the supervisory mechanism for the transient solving process.

B. State-parameterized Unified Branch Stamp Model

The state-parameterized unified branch stamp model is the core modeling method of SwitchSim, by which we model three common switching-circuit devices, namely diodes, MOSFETs, and IGBTs. In addition to realizing switching actions through basic port constraints and state-update rules, the proposed models preserve essential device semantics, such as conduction directionality and freewheeling paths, to ensure correct modeling of switching circuits.

For a diode, denote the voltage drop and branch current by U_D and i_D, respectively, where $U_\mathrm{D} = v_p - v_n$. In the on state, the diode is modeled as a short circuit, and the branch current is constrained to be nonnegative in the prescribed conduction direction. In the off state, the diode branch current is zero, and the diode must not be forward biased:

$$\begin{cases} s_D = 1: & U_D = 0, \quad i_D \geq 0 \\ s_D = 0: & i_D = 0, \quad U_D \leq 0 \end{cases}. \tag{11}$$

Its unified branch constraint can be written as:

$$s_D(v_p - v_n) + (1 - s_D) Z_0 i_D = 0. \tag{12}$$

In the MNA formulation, i_D is introduced as an additional current unknown, and the corresponding local stamp contribution of the diode can be expressed as:

$$\begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ s_D & -s_D & \cdots & (1-s_D)Z_0 \end{bmatrix} \begin{bmatrix} v_p \\ v_n \\ \vdots \\ i_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{13}$$

The diode state transition is determined by U_D and i_D in the current solution, with the state-update rule given by:

$$\begin{cases} s_D : 0 \to 1 & \text{if} \quad U_D > \varepsilon_v \\ s_D : 1 \to 0 & \text{if} \quad i_D < -\varepsilon_i \end{cases}, \tag{14}$$

where ε_v and ε_i denote the diode voltage and current tolerances, respectively. They are used to filter perturbations at the numerical-noise level, and both default to 10^{-9}.

For MOSFETs and IGBTs, in addition to the main conduction path, a reverse freewheeling path is also formed by the device structure or module-level packaging. If a MOSFET or an IGBT is abstracted only as a single ideal switch branch, then when the main switch is off and an inductive load requires a freewheeling path, the simulator cannot correctly provide a reverse current path. This may produce incorrect nodal voltages and switching states, or even lead to a topologically inconsistent or unsolvable circuit configuration. To address this issue, SwitchSim introduces an additional diode branch for both MOSFETs and IGBTs, and explicitly represents their reverse freewheeling behavior using the state-parameterized unified branch stamp model. For a MOSFET, let the drain, source and gate be denoted by D, S, and G, respectively. The voltage drop and branch current of the channel branch are defined as:

$$U_\mathrm{ch} = v_D - v_S, \qquad i_\mathrm{ch}: \ D \to S. \tag{15}$$

A body-diode branch is added to the NMOS, with its port voltage and branch current defined as:

$$U_\mathrm{bd} = v_S - v_D, \qquad i_\mathrm{bd}: \ S \to D. \tag{16}$$

In the MNA formulation, i_ch and i_bd are introduced as additional current unknowns, and the corresponding local stamp contribution of the NMOS can be expressed as:

$$\begin{bmatrix} 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & -1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_\mathrm{ch} & -s_\mathrm{ch} & \cdots & (1-s_\mathrm{ch})Z_0 & 0 \\ -s_\mathrm{bd} & s_\mathrm{bd} & \cdots & 0 & (1-s_\mathrm{bd})Z_0 \end{bmatrix} \begin{bmatrix} v_D \\ v_S \\ \vdots \\ i_\mathrm{ch} \\ i_\mathrm{bd} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

The state of the channel is determined directly by the gate-source voltage:

$$s_{\mathrm{ch}} = 1 \quad \text{if} \quad v_G - v_S \geq V_{\mathrm{th}}. \tag{18}$$

If the channel is on, the body diode is constrained to the off state. If the channel is off, the body diode is updated according to the natural commutation rule of an ideal diode:

$$\begin{cases} s_{\mathrm{bd}} = 0 & \text{if} \quad s_{\mathrm{ch}} = 1 \\ s_{\mathrm{bd}} : 0 \to 1 & \text{if} \quad (s_{\mathrm{ch}} = 0) \wedge (U_{\mathrm{bd}} > \varepsilon_v) \\ s_{\mathrm{bd}} : 1 \to 0 & \text{if} \quad (s_{\mathrm{ch}} = 0) \wedge (i_{\mathrm{bd}} < -\varepsilon_i) \end{cases}. \tag{19}$$

Similarly, the local stamp contribution of PMOS can be obtained as:

$$\begin{bmatrix} 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & -1 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{\mathrm{ch}} & -s_{\mathrm{ch}} & \cdots & (1-s_{\mathrm{ch}})Z_0 & 0 \\ s_{\mathrm{bd}} & -s_{\mathrm{bd}} & \cdots & 0 & (1-s_{\mathrm{bd}})Z_0 \end{bmatrix} \begin{bmatrix} v_D \\ v_S \\ \vdots \\ i_{\mathrm{ch}} \\ i_{\mathrm{bd}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{20}$$

Based on this, the state update rules for the channel and body diodes are the same as those for NMOS.

For an IGBT, let the collector, emitter, and gate be denoted by C, E, and G, respectively. The voltage drop and branch current of the main conduction branch are defined as:

$$U_{\mathrm{m}} = v_C - v_E, \qquad i_{\mathrm{m}} : \ C \to E. \tag{21}$$

An antiparallel-diode branch is added to the IGBT, with the following port voltage and branch current:

$$U_{\mathrm{ad}} = v_E - v_C, \qquad i_{\mathrm{ad}} : \ E \to C. \tag{22}$$

In the MNA formulation, i_{m} and i_{ad} are introduced as additional current unknowns, and the corresponding local MNA stamp contribution of the IGBT can be expressed as:

$$\begin{bmatrix} 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & -1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{\mathrm{m}} & -s_{\mathrm{m}} & \cdots & (1-s_{\mathrm{m}})Z_0 & 0 \\ -s_{\mathrm{ad}} & s_{\mathrm{ad}} & \cdots & 0 & (1-s_{\mathrm{ad}})Z_0 \end{bmatrix} \begin{bmatrix} v_C \\ v_E \\ \vdots \\ i_{\mathrm{m}} \\ i_{\mathrm{ad}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{23}$$

Unlike MOSFETs, the main conduction branch of an IGBT should not sustain a reverse current, and reverse freewheeling is instead provided by the antiparallel diode. Therefore, SwitchSim first determines the state of the antiparallel diode and then updates the state of the IGBT main conduction branch. This ordering ensures physical admissibility while avoiding the indeterminate branch-current allocation that would arise from two parallel ideal zero-voltage constraints. The state-update rule for the antiparallel diode is as follows:

$$\begin{cases} s_{\mathrm{ad}} : 0 \to 1 & \text{if} \quad [(v_G - v_E < V_{\mathrm{th}}) \wedge (U_{\mathrm{ad}} > \varepsilon_v)] \\ & \qquad \vee [(s_{\mathrm{m}} = 1) \wedge (i_{\mathrm{m}} < -\varepsilon_i)] \\ s_{\mathrm{ad}} : 1 \to 0 & \text{if} \quad i_{\mathrm{ad}} < -\varepsilon_i \end{cases}. \tag{24}$$

After updating the antiparallel diode, the state-update rule for the IGBT main conduction branch can be written as

$$\begin{cases} s_{\mathrm{m}} : 0 \to 1 & \text{if} \quad (v_G - v_E \geq V_{\mathrm{th}}) \wedge (s_{\mathrm{ad}}^{+} = 0) \\ s_{\mathrm{m}} : 1 \to 0 & \text{if} \quad (v_G - v_E < V_{\mathrm{th}}) \vee (s_{\mathrm{ad}}^{+} = 1) \end{cases}, \tag{25}$$

where s_{ad}^{+} denotes the antiparallel-diode state updated according to Equation (24). By defining the switching device models above, the state-parameterized unified branch stamp model achieves two-layer unification. First, at the physical level, the differences among switching devices are not reflected in the modeling form, but by their state-update logic, branch directionality, and the way additional freewheeling branches are incorporated. Second, at the numerical level, all ideal switching states are represented by state parameters, introducing no extreme numerical entries and not altering the effective sparsity pattern of the matrix. This design enables the simulator to preserve the topological semantics of practical power devices while avoiding the disadvantages in non-ideal switch models.

C. Dual-residual Adaptive Time-stepping Controller

During transient time advancement, ideal switching circuits introduce additional complexities beyond those encountered in general circuit simulation. BE provides numerical damping around switching events, whereas trapezoidal integration is preferred in smooth regions for higher accuracy. Meanwhile, the time grid should resolve both known switching events and state-dependent commutations. Fixed events, such as PULSE-source edges, can be directly aligned with the time grid, while state-dependent events are estimated from the evolution of device states. Accordingly, SwitchSim employs DRATS controller to coordinate continuous solution variation and switching-state evolution and to schedule the subsequent time step and integration mode.

The solution variation residual (SVR) measures the magnitude of nodal-voltage variation between two adjacent accepted time instants v^k and v^{k-1}, serving as an indicator of the smoothness of the continuous solution. It is defined as

$$R_{\mathrm{SVR}} = \frac{\max_{i \in \mathcal{N}} \left| v_i^k - v_i^{k-1} \right|}{\max\left(1, \max_{i \in \mathcal{N}} \left| v_i^k \right|\right)}, \tag{26}$$

where $\mathcal{N}$ denotes the node set. The numerator gives the maximum nodal-voltage change over the current step, while the denominator normalizes it by the current voltage scale. A small SVR indicates smooth continuous evolution, whereas a large SVR indicates pronounced dynamics and therefore limits the growth of the subsequent time step.

The device-state consistency residual (DCR) characterizes both the consistency of the current device state and its proximity to a switching boundary. For the j-th switching boundary at t_k, let $\delta_j^k \geq 0$ denote the normalized distance from the current state to that boundary. Its approach rate is defined as $a_j^k = \max\left(0, (\delta_j^{k-1} - \delta_j^k)/h_k\right)$, and the estimated boundary-approach time is $T_{\mathrm{bd},j}^k = \delta_j^k/(a_j^k + \epsilon)$, where ϵ is a small positive constant for numerical regularization. The corresponding predicted boundary residual is $\widehat{R}_{\mathrm{DCR},j}^k = h_k/(T_{\mathrm{bd},j}^k + h_k)$. In

addition, let ρ_j^k denote the normalized violation of the ideal voltage/current constraints associated with this boundary, as defined by the device-state conditions in Section III-B. The global DCR is defined as

$$R_{\mathrm{DCR}} = \max_{d\in\mathcal{S}} \max_{j\in\mathcal{B}_d} \left\{ \rho_j^k,\ \widehat{R}_{\mathrm{DCR},j}^k \right\}, \tag{27}$$

where $\mathcal{B}_d$ denotes the set of switching boundaries associated with device d, and $\mathcal{S}$ denotes the set of switching devices. The same prediction gives the boundary step limit $h_{\mathrm{bd},j}^k = \delta_j^k / a_j^k$, which constrains time advancement near a predicted natural commutation.

The two residuals are jointly used to determine the subsequent time-step scale. Let τ_{SVR} and τ_{DCR} denote the normalization thresholds of SVR and DCR, respectively, and let R_{cap} denote the upper contribution of DCR to the generic scaling rule. DRATS defines

$$\begin{aligned} \Gamma_k &= \max\left[\frac{R_{\mathrm{SVR}}^k}{\tau_{\mathrm{SVR}}}, \min\left(\frac{R_{\mathrm{DCR}}^k}{\tau_{\mathrm{DCR}}}, R_{\mathrm{cap}} \right) \right]; \\ h_{k+1} &= \mathrm{clip}\left[\bar{h}_k \max(\Gamma_k, \epsilon_\Gamma)^{-\alpha}, h_{\min}, h_{\mathrm{cap}}^k \right], \end{aligned} \tag{28}$$

where Γ_k is the unified scheduling indicator, α controls the sensitivity of step adaptation, $\bar{h}_k$ denotes the growth-reference step size, $h_{\min}$ is the minimum allowed time step, and h_{cap}^k is the current upper bound determined by the applicable maximum step, the next known event, and the nearest predicted switching boundary. The operator $\mathrm{clip}(x, l, u) = \min(\max(x, l), u)$ restricts x to $[l, u]$. For an event-aligned step, $\bar{h}_k$ retains the pre-event time scale to avoid unnecessary regrowth from the shortened event step.

DRATS further coordinates time-step adaptation with the integration scheme. In the normal smooth mode, trapezoidal integration is used in continuous regions. A step aligned with a known switching event uses BE, and an accepted step with an actual device-state transition activates a short recovery mode interval in which BE provides additional numerical damping. The solver subsequently returns to smooth as the transient settles. If a known event does not cause a device-state transition, no multi-step recovery is introduced. Thus, SVR primarily regulates the temporal resolution of continuous dynamics, while DCR and event alignment constrain time advancement around switching boundaries.

IV. Experimental Results

The proposed SwitchSim is implemented in C++, where the linear algebra backend was built on the open-source Eigen 3.4.0 library [17]. In particular, sparse linear systems are solved using Eigen's SparseLU solver, whose computation is decomposed into a symbolic analysis stage and a numerical factorization stage. The proposed Unified Branch Stamp Model preserves a fixed MNA sparsity pattern for each transient simulation, so that symbolic analysis can be reused and subsequent time steps only require performing numerical factorization. All experiments were conducted on a workstation running Windows 11, with an Intel Core Ultra 9 185H processor and 32 GB memory.

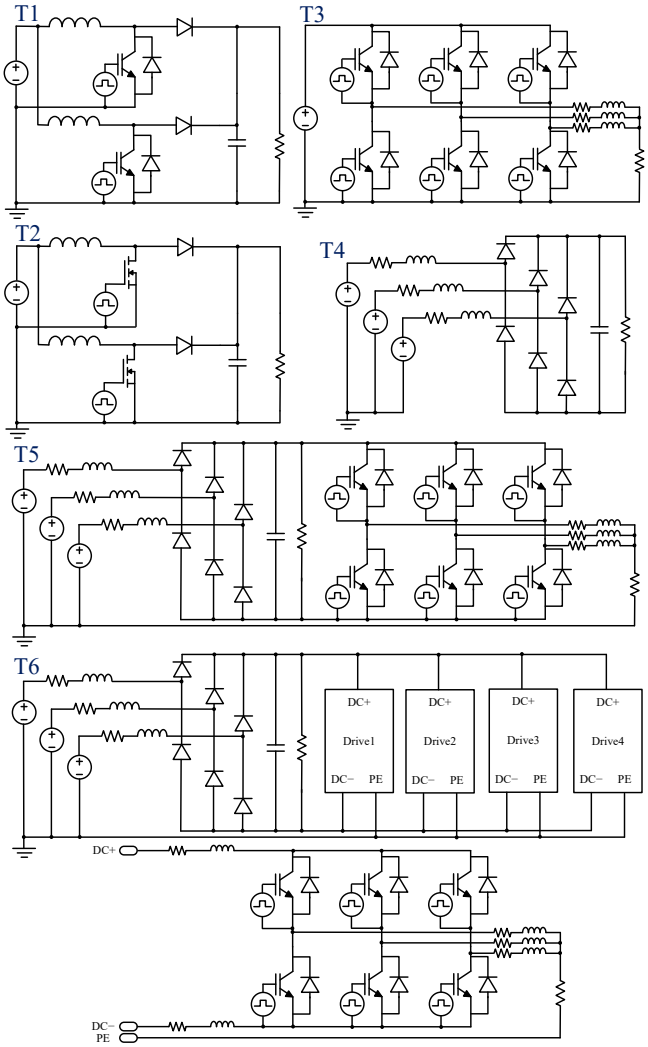

Fig. 3 The test circuits used in the experiment. For the three-phase inverter bridges in T3, T5, and T6, phases A, B, and C are phase-shifted by $120°$, with complementary upper/lower gate signals. In T6, using phase A of Drive1 as the reference, Drives1 to 4 are delayed by $0°$, $22.5°$, $45°$, and $67.5°$.

A. Test Case

Six representative switching circuits, denoted by T1-T6, are tested, as shown in Fig. 3. T1 to T4 are IGBT boost converter, MOSFET boost converter, three-phase two-level inverter, and three-phase six-pulse diode rectifier bridge, respectively. T5 adds an IGBT inverter and a three-phase RL load to the three-phase rectifier and DC-link of T4, forming a system-level power chain for evaluating simulation robustness in realistic complex scenarios. T6 further extends T5 to four three-phase IGBT inverter-drive units, and is used to evaluate state switching and sparse matrix-structure reuse in a larger-scale IGBT system. These circuit structures are based on a series of power-electronics and drive-related standard example models from the Modelica Standard Library [18], which enables the proposed method to be evaluated from multiple perspectives, including functional correctness, numerical accuracy, system-level robustness, and computational efficiency.

TABLE I Comparison of Simulation Results with PLECS as Reference

Method	T1			T2			T3			T4			T5			T6		
	MAE(V)	E_{rel}	RT(s)	MAE(V)	E_{rel}	RT(s)	MAE(V)	E_{rel}	RT(s)	MAE(V)	E_{rel}	RT(s)	MAE(V)	E_{rel}	RT(s)	MAE(V)	E_{rel}	RT(s)
PLECS [9]	–	–	**0.14**	–	–	**0.27**	–	–	**0.15**	–	–	**0.15**	–	–	**0.24**	NA	NA	NA
Siconos [19]	7.79	3.81%	1.67	7.79	3.81%	1.67	**4.30**	**1.61%**	0.94	4.46	0.71%	0.42	4.73	2.00%	6.35	–	–	121.2
LTspice [20]	0.52	0.25%	1.34	1.19	0.58%	13.12	6.10	2.28%	0.18	NA	NA	NA	NA	NA	NA	NA	NA	NA
ngspice [21]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
SwitchSim	**0.07**	**0.04%**	0.94	**0.07**	**0.04%**	0.92	4.33	1.62%	1.51	**<0.01**	**<0.01%**	0.93	**4.12**	**1.74%**	2.64	–	–	**118.9**

NA means that the simulation of the current circuit using this tool failed.

B. Comparison with Other Simulators

PLECS [9], the commercial software supporting dedicated switching circuit simulation, is used as the reference simulator for the accuracy comparison. SwitchSim is further compared with NSDS-based open-source simulation framework Siconos [19], free software LTspice 26.0.1 [20], and open-source circuit simulator ngspice Version 46 [21]. This comparison covers both dedicated switching-circuit simulation and general-purpose SPICE simulation.

For each of the six test circuits, a 0.2-s transient simulation is conducted, and a representative voltage waveform is selected for accuracy evaluation. For T1 and T2, the boost output voltage is used to assess the response of the IGBT- and MOSFET-based DC/DC converters. For T3, one phase output voltage of the three-phase two-level inverter is selected to characterize PWM switching and load response. For T4, the DC output voltage of the three-phase diode rectifier is used to evaluate natural commutation. For T5, one phase load voltage is selected to assess the inverter response in the system-level power chain. For T6, the line voltage of Drive 1 is used to evaluate the larger four-drive common-DC-bus system.

All waveforms are evaluated at 1000 uniformly distributed time instants using strict same-time comparison. When a simulator does not provide a value exactly at a sampled instant, linear interpolation between its neighboring simulation points is used. The reported accuracy metrics are mean absolute error (MAE) and relative error (E_{rel}), defined as

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{k=1}^{N}|y_k - r_k|; \\ E_{\text{rel}} &= \frac{\sum_{k=1}^{N}|y_k - r_k|}{\sum_{k=1}^{N}|r_k|} \times 100\%, \end{aligned} \tag{29}$$

where r_k and y_k denote the PLECS reference and the corresponding simulator output at the k-th sampled instant, respectively. In particular, for T6, PLECS terminates the simulation with the error message "State discontinuity after switching." We therefore compare the SwitchSim simulation results with Siconos. The resulting waveforms exhibit an MAE of 0.78 V and a relative error of 0.24%, indicating close agreement between the two simulators.

TABLE I shows that the results produced by SwitchSim closely match those of PLECS. For all test circuits, the average relative error remains within 2%. By contrast, for general-purpose SPICE simulators, substantial simulation difficulties still arise even when the switching devices are modeled as non-ideal switches using extreme resistance values. LTspice successfully simulates only three of the test circuits, whereas the open-source ngspice fails on all circuits, with the simulations stopping after advancing only to the microsecond level. These results underscore the need for a SPICE-like simulator dedicated to switching circuits. To provide a more intuitive comparison, we visualize the output voltage waveform of each circuit, as shown in Fig. 4. SwitchSim and PLECS waveforms closely agree over most of the simulation interval, and we can gauge the simulation accuracy of SwitchSim.

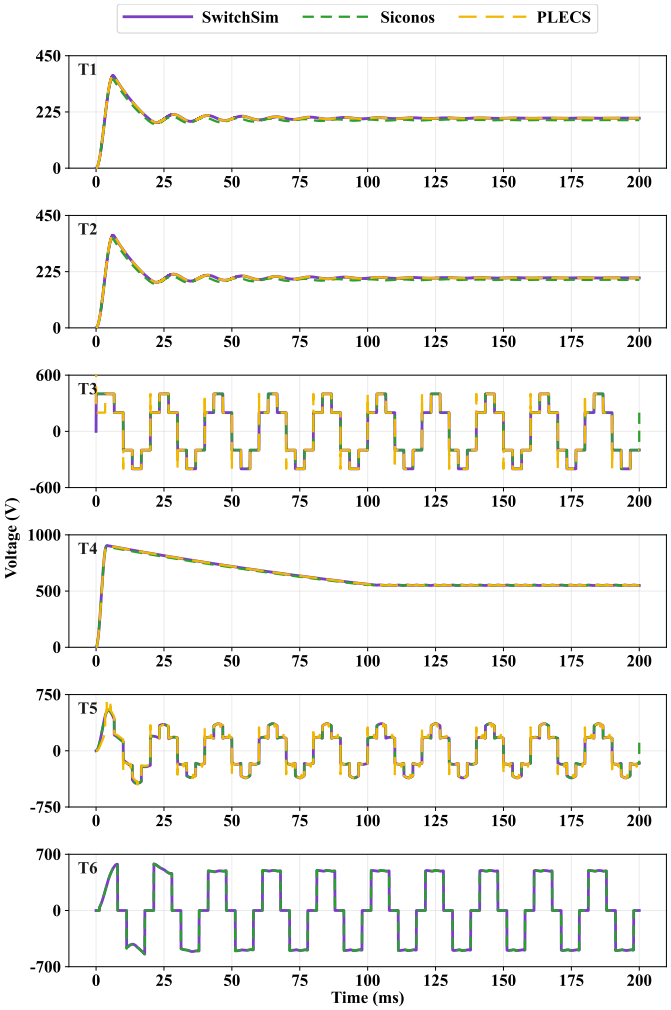

Fig. 4 Comparison of representative voltage waveforms for the six test circuits across different simulators.

TABLE II Ablation Results of Unified Branch Stamp Model

Circuit	Method	#Symb-Fact	Symb-Reuse	RT	Rel. Error
T1	w/o UBSM	24949	70.53%	0.85s	0.01%
	SwitchSim	2	100.00%	0.94s	0.04%
T2	w/o UBSM	24949	70.53%	0.88s	0.01%
	SwitchSim	2	100.00%	0.92s	0.04%
T3	w/o UBSM	360	98.35%	1.30s	1.62%
	SwitchSim	2	99.99%	1.51s	1.62%
T4	w/o UBSM	22159	55.22%	1.24s	<0.01%
	SwitchSim	2	99.99%	0.93s	<0.01%
T5	w/o UBSM	NA	NA	NA	NA
	SwitchSim	2	99.99%	2.64s	1.74%
T6	w/o UBSM	NA	NA	NA	NA
	SwitchSim	2	99.99%	118.85s	–

TABLE III Ablation Results of DRATS Controller

Circuit	Method	Rst-Dens.	#Solve/Step	RT	Rel. Error
T1	w/o DRATS	152.28%	2.91	0.49s	1.25%
	SwitchSim	52.64%	1.57	0.94s	0.04%
T2	w/o DRATS	152.28%	2.91	0.44s	1.25%
	SwitchSim	52.64%	1.57	0.92s	0.04%
T3	w/o DRATS	1.68%	1.02	1.40s	1.66%
	SwitchSim	1.32%	1.02	1.51s	1.62%
T4	w/o DRATS	766.75%	8.67	21.64s	<0.01%
	SwitchSim	42.33%	1.43	0.93s	<0.01%
T5	w/o DRATS	4.05%	1.05	2.78s	1.70%
	SwitchSim	3.79%	1.05	2.64s	1.74%
T6	w/o DRATS	9.87%	1.13	123.20s	–
	SwitchSim	8.83%	1.11	118.85s	–

C. Ablation Study

To evaluate the roles of two designs in SwitchSim, this subsection presents ablation studies on the state-parameterized unified branch stamp model and the DRATS controller. Since these designs address different aspects of the simulation, they are evaluated using different sets of comparison metrics.

The ablation results for the state-parameterized unified branch stamp model are reported in TABLE II. The ablated method retains ideal-switch semantics but uses state-dependent branch stamps, causing the MNA sparsity pattern to change with switching states. We report the number of symbolic-analysis calls (#Symb-Fact), the fraction of numerical factorizations that reuse an existing symbolic pattern (Symb-Reuse), runtime and relative error. With the proposed model, only two symbolic analyses are required in all six cases and the reuse rate remains nearly 100%. Without it, repeated symbolic analyses are required, with the reuse rate dropping to 70.53% for T1/T2 and 55.22% for T4, while T5 and T6 fail to complete. The runtime difference is small or mixed for several smaller cases, indicating that the unified branch stamp enables fixed-pattern reuse and improves robustness for complex switching circuits.

The ablation results for DRATS are shown in TABLE III. The ablated configuration disables step regulation and event alignment and directly permits the relaxed maximum step size. Rst-Dens. measures state-triggered restamps normalized by accepted time steps, while #Solve/Step measures the average number of linear-system solves per accepted step. These metrics characterize state-transition and iterative-solution overhead. DRATS reduces Rst-Dens. in all six cases and reduces or maintains #Solve/Step. In T1 and T2, DRATS incurs additional runtime but reduces the relative error, indicating that DRATS adaptively balances temporal resolution and state-transition overhead.

V. Conclusion

In this paper, we propose SwitchSim, a SPICE-like simulator for switching circuits based on a state-driven framework and ideal switch modeling, avoiding the numerical issues of extreme-resistance models in general-purpose SPICE simulators. Building on the prior unified ideal switch model, SwitchSim extends it to state-parameterized branch stamps for practical switching devices and employs a dual-residual adaptive time-stepping controller to coordinate and optimize the transient time-advancement process. Evaluations of six representative switching circuits demonstrate the effectiveness of the proposed framework.

References

[1] C.-W. Ho *et al.*, "The modified nodal approach to network analysis," *IEEE TCAS*, 1975.

[2] K. S. Kundert, *The Designer's Guide to SPICE and SPECTRE*. Springer, 1995.

[3] A. Vladimirescu *et al.*, "The simulation of mos integrated circuits using SPICE2," Tech. Rep., 1980.

[4] A. Massarini *et al.*, "Analysis of networks with ideal switches by state equations," *IEEE TCAS-I*, 1997.

[5] R. Frasca *et al.*, "Linear passive networks with ideal switches: Consistent initial conditions and state discontinuities," *IEEE TCAS-I*, 2010.

[6] V. Acary *et al.*, *Nonsmooth Modeling and Simulation for Switched Circuits*. Springer, 2010.

[7] A. B. Yildiz, "A MNA-based unified ideal switch model for analysis of switching circuits," *J. Circuits Syst. Comput.*, 2013.

[8] J. H. Allmeling and W. P. Hammer, "PLECS: Piece-wise linear electrical circuit simulation for Simulink," in *Proc. IEEE PEDS*, 1999.

[9] Plexim, "PLECS Standalone, Version 5.0.6," 2026. [Online]. Available: https://www.plexim.com/download/standalone

[10] M. K. Camlibel *et al.*, "Switched networks and complementarity," *IEEE TCAS-I*, 2003.

[11] V. Acary *et al.*, "Time-stepping numerical simulation of switched circuits within the nonsmooth dynamical systems approach," *IEEE TCAD*, 2010.

[12] V. Acary and F. Pérignon, "Siconos: A software platform for modeling, simulation, analysis and control of nonsmooth dynamical systems," *SNE*, 2007.

[13] A. B. Yildiz *et al.*, "Time-domain analysis of circuits with ideal switches by nodal equations," in *Proc. PEDS*, 2005.

[14] G. Birtek *et al.*, "Analysis of switched capacitor circuits based on unified ideal switch model," in *Proc. GPECOM*, 2020.

[15] H. W. Dommel, "Digital computer solution of electromagnetic transients in single- and multiphase networks," *IEEE TPAS*, 1969.

[16] W. Nzale *et al.*, "Accurate time-domain simulation of power electronic circuits," *Electr. Power Syst. Res.*, 2021.

[17] G. Guennebaud *et al.*, "Eigen," 2010. [Online]. Available: https://libeigen.gitlab.io

[18] Modelica Association, "Modelica Standard Library, Version 4.1.0," 2025. [Online]. Available: https://doc.modelica.org/Modelica%204.1.0/Resources/helpDymola/Modelica.html

[19] Siconos Team, "Siconos: A software package for modeling and simulation of nonsmooth dynamical systems," 2024. [Online]. Available: http://siconos.org

[20] Analog Devices, "LTspice," 2026. [Online]. Available: https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html

[21] H. Vogt *et al.*, *Ngspice User's Manual, Version 46*, 2026. [Online]. Available: https://ngspice.sourceforge.io/docs/ngspice-manual.pdf