An Event-Driven Parallel Acceleration Real-Time Simulation for Power Electronic Systems Without Simulation Distortion in Circuit Partitioning

Jialin Zheng, *Student Member, IEEE*, Zhengming Zhao, *Fellow, IEEE*, Yangbin Zeng, *Member, IEEE*, Shiqi Ji, *Senior Member, IEEE*, and Liqiang Yuan, *Member, IEEE*

***Abstract*—The trend toward higher switching frequency and larger scale of power electronic systems poses challenges for real-time simulation. This article presents an event-driven hardware-in-the-loop (ED-HIL) simulation framework that does not rely on the small fixed simulation step-size related to the switching frequency. ED-HIL framework can improve the calculation efficiency by positioning switch events and using variable simulation step-size. Further, a parallel acceleration and circuit partitioning (PACP) solver is proposed based on this framework. The PACP solver partitions the circuit with energy storage elements and ensures that the partitioning does not introduce simulation distortion by using the higher order derivatives of the energy storage elements. Meanwhile, it achieves a process-level paralleling through a shared memory architecture in order to accelerate the simulation. As a result, the proposed PACP solver based on the ED-HIL framework can achieve about three times the simulation scale of a commercial FPGA-based real-time simulator at 1/16 the hardware cost under the same conditions. An accurate real-time HIL example of a power electronic transformer with 32 switches at a switching frequency of 20 kHz has been implemented on a personal computer. The simulation results are analyzed by comparing with experiment, offline software, and commercial real-time simulator.**

***Index Terms*—Parallel simulation, power electronic systems, real time simulation.**

I. Introduction

REAL-TIME simulation and hardware-in-the-loop (HIL) have been widely used in power electronic systems (PES) testing [1]–[4]. The rapid development of PES requires real-time HIL simulators to be more economical, high-precision and real-time at the same time [5]–[7]. However, high-precision simulations need large computational resources [8]–[10]. Conventional real-time simulators can hardly achieve the real-time goal of high-precision simulation of complex PES with limited computational resources [11], [12]. To achieve the real-time goal, there are three approaches to accelerate complex PES simulations.

1) The most straightforward approach is using specific computing hardware. Using specific computing hardware is generally the most straightforward approach. Due to the different processor architectures, GPUs are typically used for acceleration in larger step-size simulation for large-scale PESs [13], [14]. FPGAs have reconfigurable parallel processing capabilities and on-chip memory enabling them to achieve nanosecond step-size simulation [15]. However, both of them are of high hardware cost and programming difficulty [16], [17].
2) Simplifying the switch model is an effective way to accelerate complex PES simulations. Real-time simulations often use the associated discrete circuit Adc switching model to reduce computations as the Adc switching model has a constant system conductance matrix [18], [19]. However, the virtual switching losses and parameter sensitivity limit its application to complex PES [20]–[22]. Thus, several enhanced Adc models have been proposed, including a generalized small-step model incorporating parametric historical current source information [20], a constant-conductance model for converters based on response matching [21], and a rational setting of parameters to reduce losses [22].
3) Simulation algorithm improvements, including improving algorithm computational efficiency [8] and using parallel computing techniques [12], are also very worthwhile research methods. Real-time simulation requires accurate capture of control signals by using high-speed sampling thereby limiting the simulation algorithm to small fixed step-size [23], which is less computationally efficient [24]. In conventional real-time simulation based on high-frequency sampling, it can hardly meet expectations to rely solely on simulation algorithm improvements. Therefore, real-time simulation is usually accelerated by system partitioning and the use of parallel computing techniques [25]–[35].

The system partitioning and parallel computing techniques for PES simulation are mainly divided into two types.

Manuscript received 5 April 2022; revised 10 June 2022; accepted 13 July 2022. Date of publication 18 July 2022; date of current version 6 September 2022. This work was supported in part by the State Key Program of National Natural Science Foundation of China under Grant U2034201 and in part by the China Postdoctoral Science Foundation under Grant 2021M701844. Recommended for publication by Associate Editor S. Golestan. *(Corresponding author: Yangbin Zeng.)*

The authors are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: zhengjl19@mails.tsinghua.edu.cn; zhaozm@tsinghua.edu.cn; ybzeng@tsinghua.edu.cn; sxjisq@gmail.com; ylq@tsinghua.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2022.3191877.

Digital Object Identifier 10.1109/TPEL.2022.3191877

One is the latency insertion method proposed by Schutt-Aine [25], which makes all branches of the whole system become inductive branches and all nodes are grounded through parallel capacitors by forcing the insertion of inductors and capacitors [26]. Then, the forward Euler method and the central integration method are applied to the state and input variables in the branch difference equations, respectively. This will differ the half-step time delay between each branch current and node voltage, thus decoupling them from each other and allowing parallel calculations [27]. However, the insertion of additional inductors and capacitors changes the dynamic characteristics of the system [28], making the simulation step length only take small values (nanosecond or sub-microsecond level), which greatly affects the simulation efficiency [29].

The other way to implement parallel computing is to decouple by energy storage elements, i.e., coupling variables. A combination of slowly varying coupling variables is used to decouple the system and subdivide it into subsystems [30]. The energy storage elements are computed using the forward Euler method, which takes advantage of the parallelizability of its explicit integration but introduces single-step latency [12], [33]. However, the single-step latency may lead to inaccurate simulation results or even non-numerically stable results, which is called simulation distortion [32]. Further, the frequent movement of the switch can exacerbate the problem [34]. Thus, a submicrosecond fixed simulation step is generally required to ensure computational accuracy and numerical stability while using these methods in the real-time simulations. This kind of simulation methods significantly reduces the simulation efficiency and requires specific computing hardware, such as field programmable gate array (FPGAs) to deal with large amount of calculations [35].

In contrast, with the goal of reducing the amount of computation, this article presents a CPU-based event-driven HIL (ED-HIL) simulation framework, which uses the simulated modulation process to obtain time-stamped switching events. The proposed framework uses a variable-step solver driven by switching events information to improve simulation efficiency. Conventional parallel simulation methods are designed for fixed-step solvers so that they do not work well in this variable-step solver framework. Therefore, a parallel accelerated and circuit partitioning (PACP) solver is proposed for process-level parallelism. This parallel acceleration method uses a multi-instruction, multidata shared memory structure that allows further scaling of the simulation without increasing the hardware cost. The proposed method has the following benefits.

1) The ED-HIL framework supports the use of variable step-size to improve efficiency, so that real-time simulation of tens of kHz switching frequency can be realized on the CPU.
2) The PACP solver does not force the insertion of a single-step delay without generating the corresponding simulation distortion. This solver can increase the simulation scale using the CPU's multi-core.
3) The hardware is accessible and economical for the simulation. Therefore, this simulation method is easy to use and can be promoted.

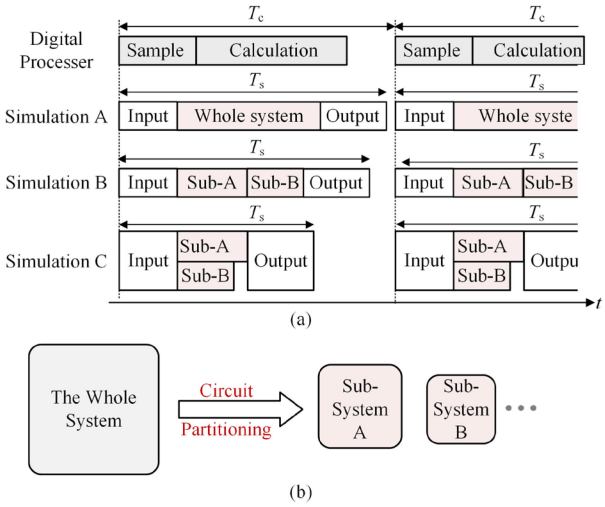


Fig. 1. Time constraints for real-time simulation. (a) Execution timeline of several real-time simulation. (b) Circuit partitioning for a PES.

The rest of this article is organized as follows. Section II introduces the ED-HIL framework. Section III introduces the modeling of PES with circuit partitioning in the PACP solver. Section IV introduces parallel acceleration numerical integration approach of the PACP solver. In Section V, a case study of multiactive bridge converter is presented to verify the performance of the proposed simulator. Finally, Section VI concludes this article.

II. Proposed ED-HIL Framework

Real-time HIL simulations have strict time constraints due to the data interaction with external hardware. In this section, the time constraint for RT simulation is discussed first, then the ED-HIL simulation framework is proposed according to the time constraint.

A. Time Constraints for RT Simulation

Real-time simulation can be divided into three parts in each simulation step, which are the input process, the calculation process and the output process in Fig. 1(a). To work with the real digital processer, the simulator first samples the analog and digital input signals from the digital processer. Then, the simulator calculates new values based on the sampled input signal. The newly calculated values are sent to the real digital processer as output signals finally. The total time of the above three parts is defined as the calculation time T_s and the control period of the digital controller can be defined as T_c. It is necessary to satisfy the following time constraint to ensure that the external controller believes it is controlling the real PES

$$T_s \leq T_c. \tag{1}$$

If (1) cannot be satisfied, it is necessary to reduce T_s. The time of the input process and the output process is limited by the hardware and remain basically constant. The simulation time

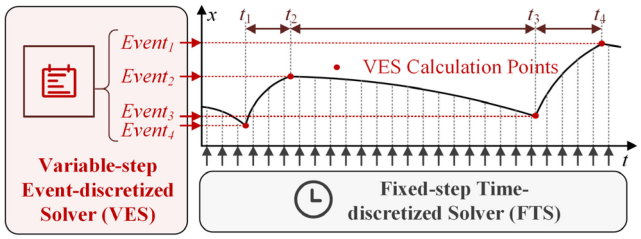


Fig. 2. ED-HIL framework and comparison between VES and FTS.

can be changed and derived from the following equation:

$$T_s = M/v \tag{2}$$

where v is the computational speed, which is mainly influenced by the computing hardware [36], and M denotes the total computational load of a control cycle T_c, and M can be expressed by the following equation:

$$M = \sum_{i=1}^{n} m_i \tag{3}$$

where m_i denotes the one-step computational load, which is proportional to the time complexity of the computation, and n denotes the number of simulation steps in a control period T_c. In general, the one-step computational load increases sharply with the system size. Specifically, the time complexity of one-step computation usually increases quadratically with n to solve an n-dimensional system [37].

Increasing v is the most direct way to reduce T_s, but it brings additional high hardware cost. To avoid the cost, T_s can be decreased by reducing the one-step computation load and reducing the number of computation steps.

It can reduce one-step computation load m_i by using circuit partitioning and parallel acceleration, as shown in Fig. 1(a). Simulation *A* processes the whole system sequentially. Simulation *B* reduces the computation load by dividing the whole system into two small subsystems. the effect of circuit partitioning is shown in Fig. 1(b). Simulation *C* parallelizes the subsystems based on the circuit partitioning, which can further reduce the computation load of each core and the computation time depends on the matrix size of the largest subsystem.

B. ED-HIL Framework

Another computation load reduction method is reducing the number of simulation steps, which is precisely the goal of ED-HIL framework. As shown in Fig. 2, the digital controller requires only one interaction of data in a control cycle T_c, which is also referred to as a sync-event. It is obvious that there is no restriction on the selection of simulation step-size between two sync-events.

The selection of simulation step-size is generally limited to small fixed step-size in conventional RT simulation. The reason is that the switching pulse signal from the digital controller is rarely synchronized with the simulation step selected by the simulator. The simulator has to select a small fixed-step time-discretized solver (FTS) to precisely sample the switching events, as shown in Fig. 2. The switching delay caused by

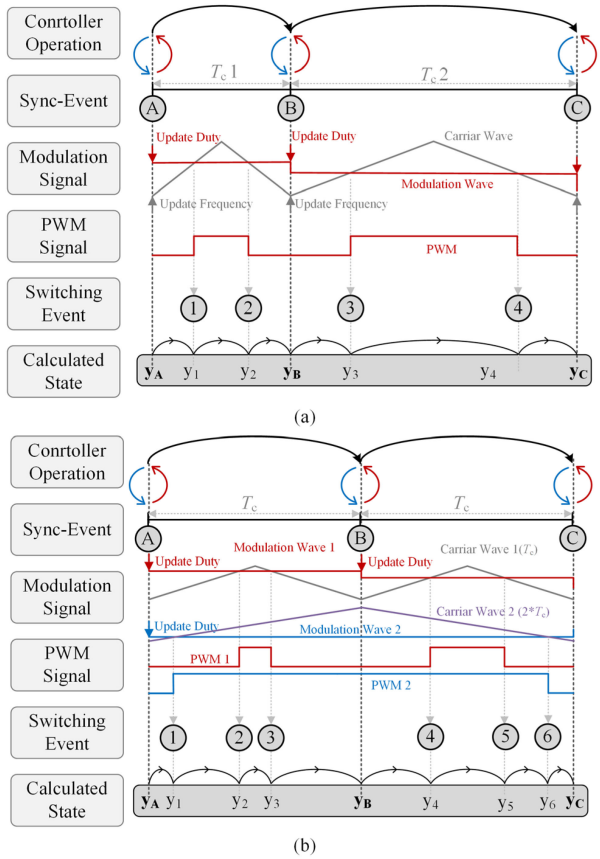


Fig. 3. ED-HIL framework. (a) Variable frequency condition. (b) Multifrequency condition.

sampling switching events can be reduced by decreasing the step size, thus ensuring the accuracy and numerical stability of the simulation [21]. However, FTS introduces a large number of unnecessary calculations, resulting in some problems, such as stronger time constraints on the simulation and higher hardware requirements.

The ED-HIL framework is proposed to solve the problems posed by the FTS solver. Specifically, the controller generates multiple modulating signals for converters, carrier signals and modulation signals are synchronized at the beginning of each control cycle T_c, as is shown in Fig. 3. The modulation signal generated by the controller remains constant between the two sync-events. The ED-HIL framework calculates the PWM signals by comparing the sampled modulating signals with the simulated carrier wave. The ED-HIL framework can obtain the moment and state of all switching events at the beginning of each control cycle. Therefore, the limitation of small fixed simulation steps is broken, and a variable-step event-discretized solver (VES) can be employed on the ED-HIL framework.

In addition, the ED-HIL framework can also deal with variable switching frequency condition [see Fig. 3(a)] and multiple switching frequency condition [see Fig. 3(b)]. In these special conditions, ED-HIL can still simulate the modulation process of the controller to find all switching events.

As shown in Fig. 2, the number of VES computation steps (four points) is much less than the number of FTS computation steps (23 points) in time t_1-t_4. Even if no switching event occurs,

the step size can be flexibly adjusted according to the sync-events and the state variables can be updated to avoid simulation errors. The simulation can be solved more efficiently by using a variable step size based on switching events and sync-events. As a result, the number of computation steps and computation burden can be reduced significantly.

Reducing the number of computation points does not neglect the transition process of state variables between switching events. The ED-HIL framework uses a numerical integration algorithm with variable-step and variable-order based on switching events and sync-events. The algorithm can select the optimal number of steps and orders for numerical integration based on specified absolute error tolerances and information about switching and synchronization events. The maximum order of the algorithm is limited to prevent the order of the algorithm from becoming unlimited.

In addition, ED-HIL requires the information about the controller reference waveform, which may cause a little extra modification to the microcontroller. ED-HIL anticipates all switching events in a switching cycle in advance by simulating the modulation process of the microcontroller. This avoids the problems associated with traditional high frequency sampling methods.

To further improve the efficiency of the simulation, the PACP solver is proposed based on the ED-HIL framework. The details of the PACP solver will be explained in the Section III.

III. Modeling With Circuit Partitioning of the PACP Solver

The circuit partitioning of the PACP solver is proposed to establish the prerequisite for parallel accelerating the simulation in this section. Based on energy storage elements, the circuit partitioning method divides the entire PES into the subsystems with coupling variables. Then, the modeling of subsystems with the coupling variables is given based on the discretization of the state space equations.

A. Circuit Partitioning Based on Energy Storage Elements

The circuit partitioning method based on parallel capacitors is illustrated by Fig. 4(a). The whole PES consists of two subsystem and a capacitor connected in parallel with them. The PACP solver partitions the whole system into two separated subsystems and a coupling system representing the capacitor.

To avoid the distortion caused by the single-step latency, the proposed PACP solver introduces the coupling variables in the circuit partitioning process. The coupling variables are considered as the time-varying sources in both the subsystem and a coupling system at each simulation step. The algebraic relations of the coupling variables between different systems are first described according to the mechanism of the energy storage elements. In the coupling system k, the relationship of voltage and current of a capacitor C is related as follows:

$$i_A(t) - i_B(t) = C\frac{dx_C(t)}{dt} \tag{4}$$

where $i_A(t)$, and $i_B(t)$ are the input variables of the coupling system k. The voltage $x_C(t)$ is the state variable of the coupling system k.

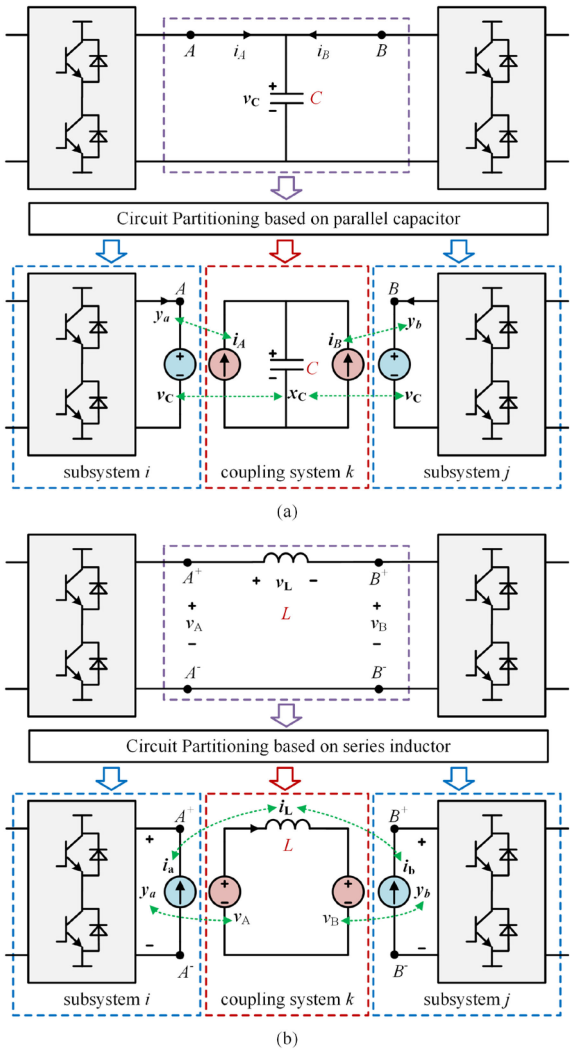


Fig. 4. Circuit partitioning based on (a) parallel capacitor. (b) Series inductor.

The coupling relationship between subsystem i, j and coupling system k can be expressed as

$$w_k = \begin{bmatrix} i_A \\ i_B \end{bmatrix} = \begin{bmatrix} y_a \\ y_b \end{bmatrix} = y_{i,j}(\alpha)$$
$$w_{i,j} = \begin{bmatrix} v_a \\ v_b \end{bmatrix} = \begin{bmatrix} x_C \\ x_C \end{bmatrix} = x_k(\alpha) \tag{5}$$

where $y_i(\alpha)$ and $y_j(\alpha)$ are referred to as the output variables in subsystems i and j associated with the coupling system k. w_k is the input variables in the coupling system k. v_a and v_b are referred to as the coupling variables in subsystems i and j, and x_k is the state variable in coupling system k.

The schematic diagram of the circuit partitioning with a series inductor as the coupling element is shown in Fig. 4(b), and the modeling equations and coupling relationships are similar to those of the capacitor as the coupling element.

$$v_A(t) - v_B(t) = L\frac{dx_L(t)}{dt} \tag{6}$$
$$w_k = \begin{bmatrix} v_a \\ v_b \end{bmatrix} = \begin{bmatrix} y_a \\ y_b \end{bmatrix} = y_{i,j}(\alpha)$$

$$w_{i,j} = \begin{bmatrix} i_a \\ i_b \end{bmatrix} = \begin{bmatrix} i_{\mathrm{L}} \\ i_{\mathrm{L}} \end{bmatrix} = x_k(\alpha). \tag{7}$$

B. Subsystems Modeling With Coupling Variables

A general PES can be modeled as a set of state space equations, (8) is referred to as the state equation and (9) is referred to as the output equation

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{8}$$

$$y\,(t) = Cx\,(t) + Du\,(t) \tag{9}$$

where $\boldsymbol{x}(t)$ is an $n \times 1$ vector, called the state variable, used to express the independent state variables, such as capacitor voltage and inductor current in the system. $\boldsymbol{u}(t)$ is an $m \times 1$ vector, called the input variable, which is used to represent the external input to the system, such as the grid voltage. $\boldsymbol{A}$ is an $n \times n$ matrix and $\boldsymbol{B}$ is an $n \times m$ matrix, which represent the connection between the state variables $\boldsymbol{x}(t)$ and the input variables $\boldsymbol{u}(t)$ in the whole PES, respectively.

The state equation of coupling system k can be expressed inductively as follows:

$$\dot{x}_k = G_k w_k \tag{10}$$

where x_k denotes the state variable of coupled system k, u_k denotes the coupling variable, and G_k is the associated matric.

The subsystems have an additional coupling variable $\boldsymbol{w_i}(t)$ compared to the general system. The state space equation of the subsystem i containing the coupling variable is shown as

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + E_i w_i(t) \tag{11}$$

where i denotes the subsystem number, $\boldsymbol{w}(t)$ is an $s \times 1$ vector, and $\boldsymbol{E}$ is an $n \times s$ matrix.

The output equation of subsystem i can be expressed as

$$y_i(t) = C_i x_i(t) + D_i u_i(t) + F_i w_i(t) \tag{12}$$

where C_i, D_i, and F_i are the associated matrices.

The relationship between the subsystem i, j and the coupling system k [(5), (7)] can be expressed inductively as

$$\begin{cases} w_i = w_j = x_k \\ w_k = y_{i,j}(\alpha) \end{cases} \tag{13}$$

where $\boldsymbol{x_k}$ denotes the independent state variables in the coupling system k, and $\boldsymbol{w_k}$ is the input variables associated with the subsystems.

The CPU core resources of the real-time simulator are limited by the hardware conditions. Therefore, the basic principles of PES circuit model partitioning are important. The basic principles of the proposed partitioning method are as follows. On the one hand, an automated circuit partitioning method is used to reduce the computational load by partitioning the circuit into subsystems as much as possible [32]. On the other hand, the partitioned subsystems are assigned to CPU multicore resources according to the principle of balanced computational load.

However, the limitation of the circuit partitioning method is that the circuit can be divided into subsystems by any energy storage element by the above partitioning method, which is not necessary. For example, it results in a tiny subsystem (only one capacitor or one inductor) in converters containing *LC* filters. Therefore, the circuit partitioning needs to add the restriction that only the energy storage element connected to the converter is used as a coupling element.

IV. Numerical Integration Approach of the PACP Solver

This section presents a numerical integration approach based on Taylor series expansion and implemented on a multiple instruction multiple data (MIMD) shared memory architecture. The coupling variables are considered the variables with high-order derivatives rather than the variables with zero-order derivative (constant value) in each simulation step, avoiding simulation distortion in circuit partitioning.

A. Numerical Integration Approach Based on Taylor Series Expansion

Based on the discretization and Taylor expansion of the continuous state equation, the numerical solution of the equation represented by (8) at time point $t = t_{n+1}$ can be expressed as

$$x(t_{n+1}) = x(t_n) + \sum_{r=0}^{p} \frac{x_n^{(r)}}{r!} \Delta t_n^r \tag{14}$$

where $\Delta t_n = t_{n+1}$-t_n denotes the n th computational step, $x_n{}^{(\mathrm{r})}$ denotes the r th order derivative of $\boldsymbol{x}(t)$ at $t = t_n$, and p denotes the order of the Taylor series.

It is worth mentioning that the proposed PACP solver discretizes the state space (8) as being a linear time-invariant (LTI) system in order to easily obtain the higher order derivatives $x_n{}^{(\mathrm{r})}$. The LTI system has the following recursive property:

$$\begin{cases} x^{(r+1)}(t) = Ax^{(r)}(t) + Bu^{(r)}(t) \\ y^{(r)}(t) = Cx^{(r)}(t) + Du^{(r)}(t), \end{cases} \tag{15}$$

where $x^{(\mathrm{r})}(t)$ and $u^{(\mathrm{r})}(t)$ denote the r th order derivatives of the state variable vector and the input variable vector, respectively.

The coupling system (10) and subsystem (11), (12) containing coupling variables have the same recursive properties as the general PES can be determined by

$$x_i^{(r+1)}(t) = A_i x_i^{(r)}(t) + B_i u_i^{(r)}(t) + E_i w_i^{(r)}(t) \tag{16}$$

$$y_i^{(r)}(t) = C_i x_i^{(r)}(t) + D_i u_i^{(r)}(t) + F_i w_i^{(r)}(t) \tag{17}$$

$$x_k^{(r+1)} = G_k w_k^{(r)}. \tag{18}$$

The simulation step-size can be adjusted according to the time interval of the switching events obtained from the proposed sampling method in Section II. The order and step-size of the Taylor polynomial (14) are adjusted according to the LTE formula (19) to meet the error requirements of the simulation

$$\varepsilon_k = \frac{x_n^{(p+1)}}{(p+1)!} \Delta t_n^{p+1} + O\left(\Delta t_n^{p+2}\right) \approx \frac{x_n^{(p+1)}}{(p+1)!} \Delta t_n^{p+1}. \tag{19}$$

The proper update of the coupling variables becomes a priority to ensure the accuracy of parallel simulation based on circuit partitioning. The proposed PACP solver uses the coupling variables with high-order derivatives and updates the derivatives of the coupling variables at each simulation step.

Specifically, the first task is to calculate the derivatives of the state variables of each system. The derivatives of the state variables $\boldsymbol{x}_i^{(r)}(t)$ can be easily obtained in each step of the simulation according to (16), (17) in the subsystem i. The derivatives of the state variables $\boldsymbol{x}_k^{(r)}(t)$ can be easily obtained by (18) in the coupling systems k. The next task is to calculate the derivatives of the coupling variables. The equation for the derivative of the coupling variable ($\boldsymbol{w}_i^{(r)}(t)$, $\boldsymbol{u}_k^{(r)}(t)$) can be derived from (13), as follows:

$$w_i^{(r)}(t_n) = x_k^{(r)}(t_n) \tag{20}$$

$$w_k^{(r)}(t_n) = y_i^{(r)}(t_n). \tag{21}$$

It can be found that the derivatives of the coupling variables are the derivatives of their corresponding state variables and output variables from (20), (21). The second task is to update the derivatives of the coupling variables of the subsystems without additional calculations, since the derivatives of the state variables corresponding to the coupling system have already been calculated at this point. In detail, the $r+1$ th order derivative of the state variable $\boldsymbol{x}_i^{(r+1)}(t_n)$ of the subsystem i is first calculated according to (16) (at this time $\boldsymbol{w}_i^{(r)}(t_n)$ is readily available). And then the output variable $\boldsymbol{y}_i^{(r)}(t_n)$ of the subsystem i is calculated according to (17), and the coupling variable $\omega_k^{(r)}(t_n)$ of the coupling system k is updated according to (21), thus solving the state variable $\boldsymbol{x}_k^{(r)}(t_n)$ of the coupling system k according to (18).

The third task is to perform the numerical integration when the derivative order satisfies the error requirement. The state variables can be calculated according to (14), and the coupling variable i can also be expressed by Taylor expansions as follows:

$$w_i(t_{n+1}) = w_i(t_n) + \sum_{r=0}^{p} \frac{w_n^{(r)}(t_n)}{r!} \Delta t_n^r. \tag{22}$$

The correct update of the derivatives of each variable is the most important in the whole process. The derivatives are updated within each simulation step in the PACP solver. Thus, the interference of simulation distortion can be avoided from the circuit partitioning. The calculations within the subsystem can be performed in parallel, since the derivative calculation and numerical integration of each subsystem do not affect each other.

B. Parallel Simulation Implementation Based on MIMD Shared Memory Architecture

The implementation of the proposed simulation algorithm uses a multicore CPU system based on MIMD shared memory architecture. In this article, the discussion is limited to single CPU systems with multiple cores, which have small interprocessor communication latency between cores. Data exchange between cores is guaranteed using multiple atomic clocks, which are important for efficient process synchronization.

The subsystems and coupling systems obtained by circuit partitioning are first assigned to different cores for computation. The coupling system contains only one state variable, which has low computational load. Even if it contains several coupling circuits, the computational load is smaller than that of the subsystems, so all coupling circuits can be assigned to one CPU core. Subsystems with a large computational load can be processed faster using parallel techniques.

The simulation tasks of the parallel simulation are assigned in Fig. 5. Process 0 is used to handle all the calculations of the coupling system (small amount of computation), controls the synchronization of the remaining processes and is responsible for updating the coupling variable. The other processes perform the subsystem simulation calculations and cooperate with the scheduling of process 0.

Process 0 controls the synchronization of the remaining processes based on the information of all the switching events of the system. The proposed method sets the step size based on the time of all switching events of the whole system, due to the fact that each switching action has an effect on the whole system. In each step of the simulation, the information about the derivatives of all state quantities is calculated. The numerical integration is performed only at the moment of the switching event acting on this subsystem to reduce unnecessary calculations. The derivative calculation flow of the proposed method is as follows, and the algorithm flow is shown in Fig. 5.

1) Using (20), processes i and j update the coupling variables $\boldsymbol{w}_i^{(r)}(t)$, $\boldsymbol{w}_j^{(r)}(t)$ required by the subsystem processes i, j of the r th order derivative.
2) Using (16), (17), processes i and j calculate the $r+1$ th derivative of the state variables $\boldsymbol{x}_i^{(r+1)}(t)$of subsystem i,j and the r th order derivative of the output variable $\boldsymbol{y}_i^{(r)}(t)$ respectively.
3) Using (21), process *0* updates the r th order derivatives of the coupling variables $\boldsymbol{u}_k^{(r)}(t)$ required by the coupling system k.
4) Using (18), process *0* calculates the $r+1$ th order derivative of the state variables $\boldsymbol{x}_k^{(r)}(t)$ of the coupling system k.

The advantage of the proposed method is that the derivatives of all the coupling variables are prepared in advance in each step of the calculation in the different processes. The highest order p of the Taylor expansion is confirmed according to the error requirement needed for the simulation (19). The above steps are computed sequentially from $r=0$ to $r=p$ to obtain the derivatives of each order of all variables, as shown in Fig. 6(a). By using the Taylor expansion and the sharing of coupling variables, all coupled variables are regarded as time-varying variables rather than constant variables in each step. Therefore, there is no simulation distortion in the circuit partitioning and numerical integration.

The derivatives of coupling variables between different processes are communicated through the shared memory architecture. To ensure consistent data and synchronization of the simulation, synchronization measures had to be taken for the data in the shared memory. The proposed approach relies on two layers of data synchronization, as shown in Fig. 6. The purpose of the first layer of data synchronization is to ensure that the derivatives of each order are updated correctly. The second layer of data synchronization is to ensure the correct order of the integration steps.

The proposed approach implements a synchronization mechanism based on atomic locks on shared memory and specifies a two-level interaction protocol, as shown in Fig. 6. Fig. 6(a) represents the different processes and operation time diagrams,

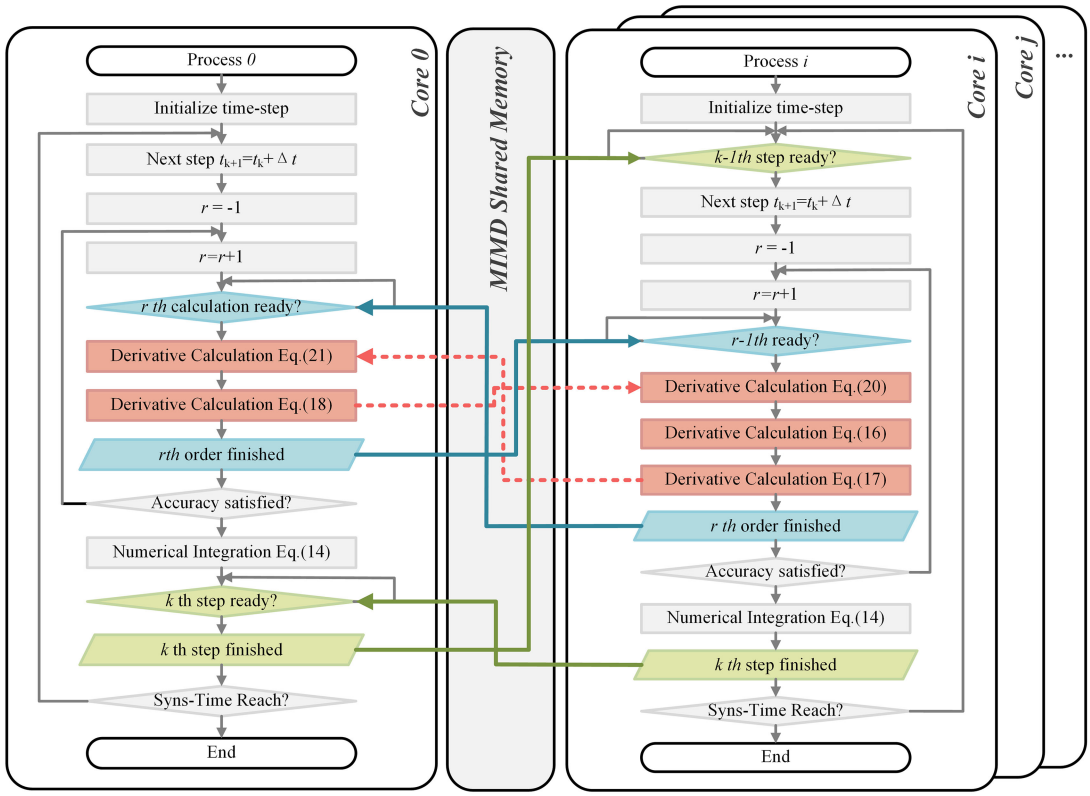


Fig. 5. Flowchart of parallel acceleration algorithm with two layers of data synchronization.

and Fig. 6(b) represents the direction of data and semaphore transfers between different CPU cores and their timing.

The corresponding execution process is shown in Fig. 6(a), and the communication of data in shared memory is shown in Fig. 6(b). The blue content in Fig. 5 is the first level of atomic locking. It is shown in Fig. 6(a) that the coupling system calculates the r th-order derivative only after all subsystems have completed the calculation of the r th-order derivative. Subsequently, the coupling system finishes calculating the r th-order derivative before the subsystem's derivative calculation proceeds to the $r + 1$ th-order calculation. The green content of Fig. 5 represents the second level of atomic locking. The different sizes of the subsystems lead to different computation times for each core. As shown in Fig. 6(a), to ensure the correct sequence of steps in the simulation, the next calculation step is allowed to start only after the process with the longest calculation time in the previous step has finished.

C. Discussions

The parallel numerical integration method proposed in this section is implemented in the framework of a shared memory-based multicore CPU system. For a more balanced parallel computation, the PACP solver has a method of allocating multicore parallel computation load. This computation load allocation method is mainly for the case where the number of the subsystems is larger than the number of the accessible CPU cores. Specifically, the PACP solver counts the number of state variables of each subsystem to estimate the computation load of the subsystem. The small subsystems are combined and allocated to one CPU core and the large subsystems is allocated separately to other CPU cores. Therefore, it is ensured that the computational load of all cores is close by evaluating the number of state variables for each subsystem.

Although energy-storage-element-based circuit partitioning are commonly used in real-time simulations [30], [31], the new contribution of the proposed PACP solver is to reuse the derivative information obtained by the subsystem integration to increase the order of the Taylor expansions of the coupling variables. This derivative reuse does not require additional computation due to the precisely designed of the integration process. The PACP can eliminate the single-step latency caused by the low order Taylor expansion of the coupling variables in the energy-storage-elements-based circuit partitioning in conventional real-time simulations. Hence, the circuit partitioning approach in PACP is more suitable for the proposed variable-step integration solver and does not produce simulation distortion due to circuit partitioning.

In addition, the model of the power converter is not simplified to ensure the high fidelity of the simulation. Therefore, the high-order matrix brought by large-scale power switching may affect the key factor of the simulation calculation performance. In this article, an automated semisymbolic state equation generation method is used to reduce the number of elements at each matrix update [38]. Meanwhile, the PACP solver proposed in this article reduces the dimensionality of the system matrix by partitioning

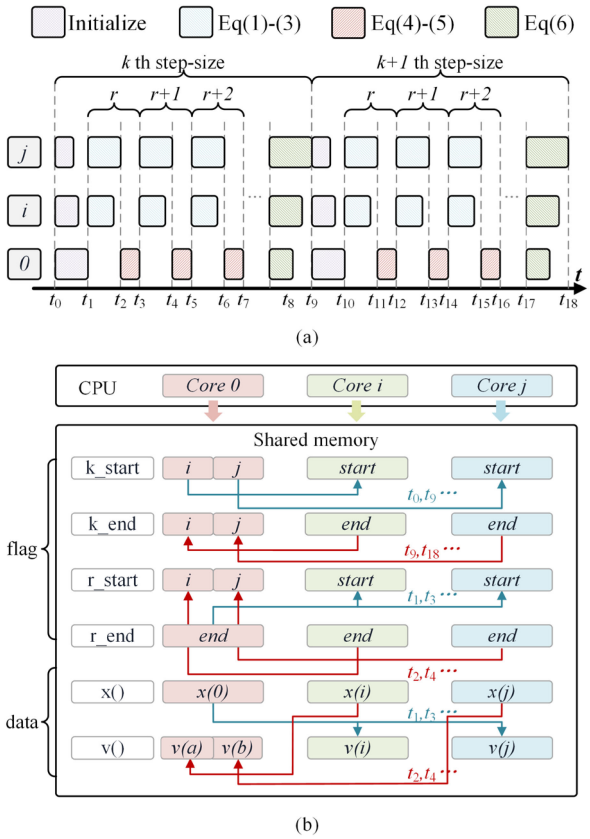


Fig. 6. MIMD shared memory architecture. (a) Execution timeline of the PACP solver. (b) Data flow in shared memory.

the higher-order matrix and increases the computational speed by computing the matrix of each partition in parallel.

V. Experimental Verification and Simulation Results

In this section, a complex power electronics system, a 100 kW four-port multimodule multi active bridge (MMAB), is simulated as the studied case to verify the effectiveness of the ED-HIL framework and the PACP solver.

A. Hardware Platform

In this article, a 100 kW MMAB is selected as a case study. The MMAB can flexibly change the power flow in the distribution network and plays an important role in the smart grid. However, the MMAB design needs to consider various operating conditions, including extreme conditions that are difficult to achieve in the real world, such as testing the response of the MMAB when the grid voltage suddenly drops. HIL simulation is a good solution for MMAB design due to its ability to simulate various extreme operating conditions. The topology of the studied MMAB is shown in Fig. 7(a) and the photography is shown in Fig. 7(b). The topology is modular in design and contains two H-bridges and one high frequency transformer (HFT) in each port. The HFT sides of these ports are connected together in parallel through a common low-voltage high frequency bus, and the other side can change voltage levels. All HFTs in this topology have a voltage ratio of 1:1, allowing the ports to be isolated from each other. Each port can output either dc directly at the capacitor or ac through the inverter, allowing energy routing to the grid, renewable energy sources and various loads. The power rating of each port is 50 kW, and the power rating of the MMAB is 100 kW. The device under study contains 32 switching devices, and the frequency of the high frequency bus is 20 kHz in order to increase the power density. It is a challenge to implement real-time computing hardware simulation for a system of this scale and frequency.

The PACP solver was developed on a personal computer (PC) as the simulation hardware platform considering the cost and openness. The PC is generally available and inexpensive, which facilitates the diffusion of this HIL simulation method. The ED-HIL simulation framework and the PTAC simulation approach proposed in this article are applied to the PC, and the real-time HIL simulation of the above MMAB is implemented on this platform. The computing unit of this simulation platform is a CPU (Intel i7-10700) and contains 16 Gb of computing memory, as shown in Fig. 7(c). The proposed simulator is deployed on multiple cores of Intel CPU, as is shown in Fig. 8. The simulation program needs to be deployed on a real-time Linux system to make it more suitable for real-time tasks. The proposed simulator consists of two main parts, the ED-HIL simulation framework that is mainly used to acquire and sequence switching events for flexible adjustment of step size, and the PACP solver that is based on a shared memory communication mechanism to perform simulation computations in parallel on multiple cores.

The control system of the MMAB is deployed in the real controller Zynq AC7Z035, which communicates with the proposed real-time simulator through the PCIe bus with a communication bandwidth of about 5 Gb/s. A commercial real-time simulator, Op5700 of Opal-RT, is also used for real-time HIL simulation of the MMAB, as shown in Fig. 7(d), to compare the performance with the proposed real-time simulator.

B. Accuracy Validation of the Simulation Results

The MMAB is simulated by the proposed ED-HIL framework with PACP solver. The offline simulation software DSIM is used as a benchmark to verify the accuracy of the proposed the PACP solver.

DSIM is based on Taylor expansions for integration calculations, and the accuracy of the simulation is verified by comparison with the other commercial simulation software [8]. It is proved that the simulation speed of DSED algorithm used in DSIM is 1000 times faster than the other commercial simulation software [9]. Therefore, the comparison with the offline simulation DSIM can better illustrate the high simulation efficiency and accuracy of the PACP solver.

The ports 1 and 4 of the studied MMAB are connected to a single-phase ac power supply of 220 V per phase. Ports 2 and 3 are connected to different sizes of loads, and the other circuit parameters are shown in Appendix. In this section, the control system parts are simulated by the simulator so as to ensure that the control system parts of the real-time simulator and the offline simulation are consistent. The switching frequency

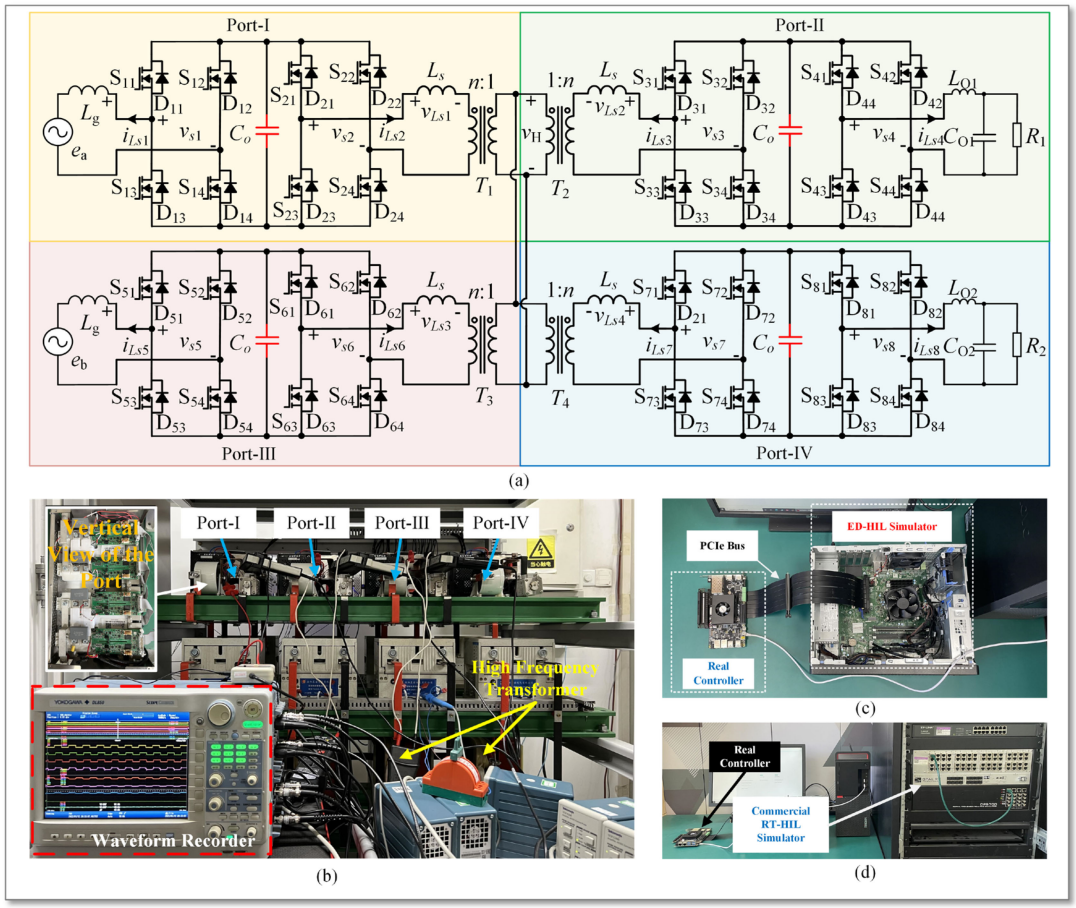


Fig. 7. Hardware platform. (a) Topology of the studied system. (b) Photography of the studied system. (c) Hardware of the proposed ED-HIL framework with the PACP solver. (d) Commercial RT-HIL simulation hardware.

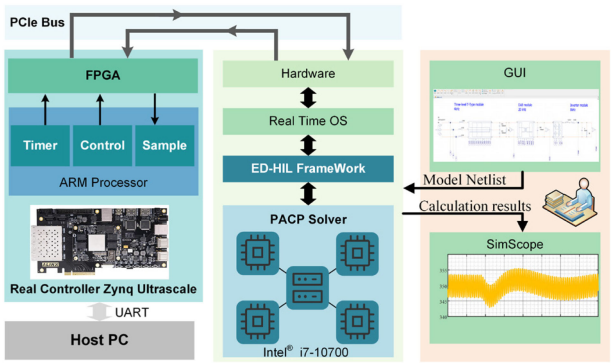


Fig. 8. Hardware Architecture of the proposed Simulator.

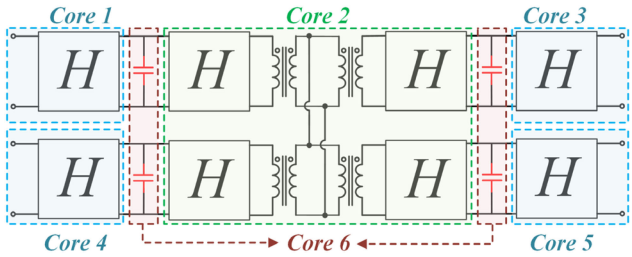


Fig. 9. Circuit Partitioning of the studied case.

of the H-bridge on the HFT side is 20 kHz, which is the highest switching frequency of the whole system.

The proposed solver uses the four capacitors of the MMAB [drawn in red in Fig. 7(a)] as energy storage elements. The whole system is partitioned into five parts by these energy storage elements and deployed on five CPU cores for parallel simulation, as is shown in Fig. 9.

The voltage and current waveforms of the MMAB with a sudden load change at port 2 are shown in Fig. 10. The black line is the DSIM waveform, and the red line is the simulation waveform of PACP solver. It can be seen from these plots that the real-time simulation results are very close to the offline simulation results in both steady-state and switching transients.

The simulation waveform of PACP solver and the simulation waveform of offline simulation software match well as shown in Fig. 10. It is worth mentioning that the offline software DSIM does not have simulation distortion caused by circuit partitioning [9]. Using the waveform of the offline software as a benchmark, it can be found that the waveform and calculation points of PACP solver are highly matched with those of the offline software in the detailed waveforms (cv. 1-8). Therefore, it can be illustrated that

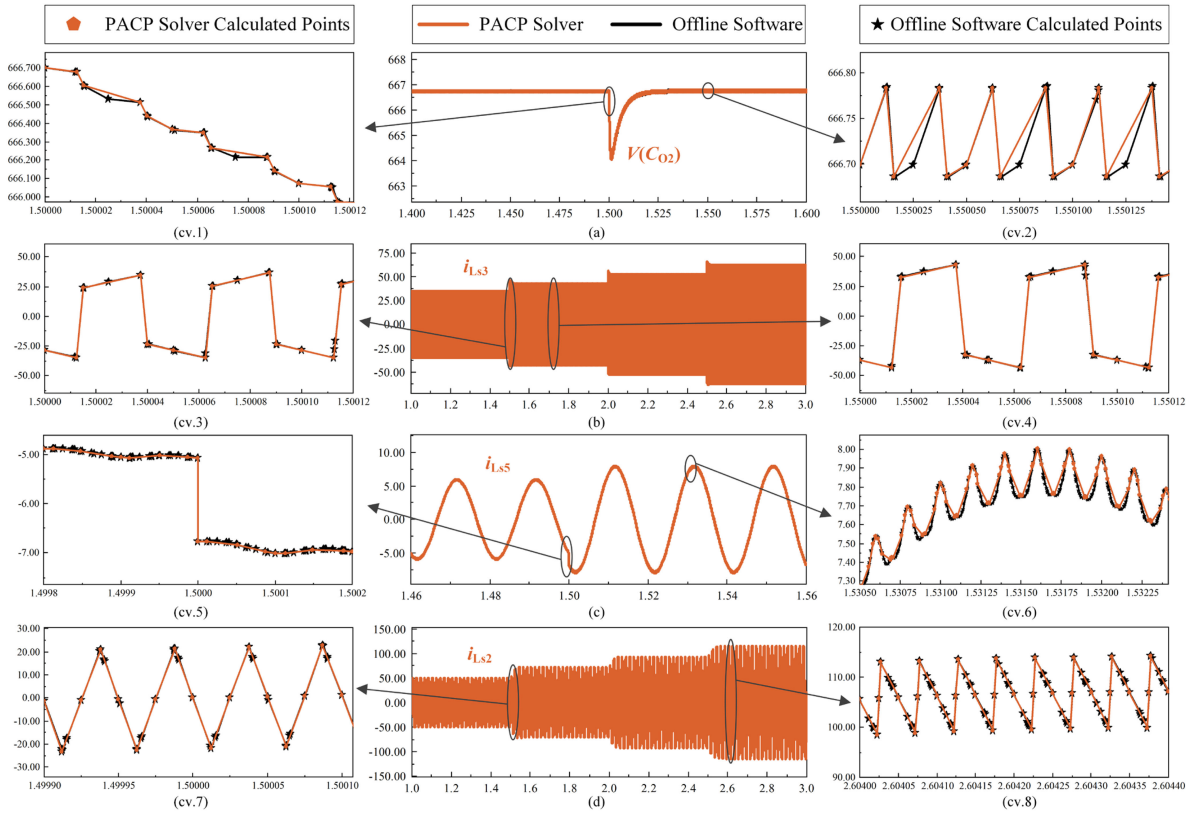


Fig. 10. Simulation results comparisons between commercial offline software (DSED solver) and proposed PACP solver. (a) Capacitor voltage port II. (b) MMAB current in port II. (c) Inductor current in port III. (d) MMAB current in port I.

no simulation distortion is generated in the circuit partitioning and parallel computation of the PACP solver.

The subtle differences between PACP solver and DSIM simulation waveforms are due to the different calculation steps. To maintain high accuracy, the proposed solver updates the derivative information at the moment of all switching events in the whole system, which is similar to the derivative update method used in DSIM. However, the proposed method only computes the integration at the moment of the switching event in each subsystem, compared to DSIM, which still needs to compute the integration at the moment of each switching event in the whole system. It can be seen from cv. 1-8 in Fig. 10 that the PACP solver can have fewer calculation points and same accuracy compared to commercial offline software DSIM. The details of the proposed method and the method used in DSIM in terms of relative error, calculation points, and calculation time are compared in Table I, and is shown in Fig. 11(a) and (b).

In summary, the PACP solver proposed in this article achieves real-time simulation while ensuring high computational accuracy and without causing additional simulation distortion.

TABLE I
SIMULATION EFFICIENCY COMPARISON

	Proposed PACP Solver			Offline software
Test Case	3s dynamic process with load change			
CPU Core	6			1
Calculation Points	Core 1 400974 Core 4 400 974	Core 2 542216 Core 5 449 997	Core 3 449997 Core 6 542 216	1056340
CPU Time Per Control Cycle(us)	Core 1 10.5 Core 4 10.3	Core 2 40.0 Core 5 9.6	Core 3 10.8 Core 6 5.5	139.5
Relative Error of i_{Ls5}	5.48e-6			1.30e-7
Relative Error of C_{o1} voltage	4.32e-5			7.85e-5

C. Comparison of HIL Simulation With Experiments

In this section, a three-port high-frequency soft-switching scenario is performed in proposed HIL simulator and experiments. Real controllers are used for both simulation and experiments. The power circuit part of the MMAB is simulated on the developed real-time simulator. The results are as follows. It is given in Fig. 12 that the comparison for the simulation and experimental results of the voltage and current in MMAB.

The MMAB waveform is the most challenging part of the real-time HIL simulation. Since the voltage of the MMAB changes in steps, this places a higher demand on the computational accuracy of the simulation and the sampling of the control signal. The shift phase ratios for ports 1, 2, and 3 are 0.1, 0.3, and 0.2, respectively.

It should be further noted that the spurious parameters on the HFT are not considered in the real-time simulation. These

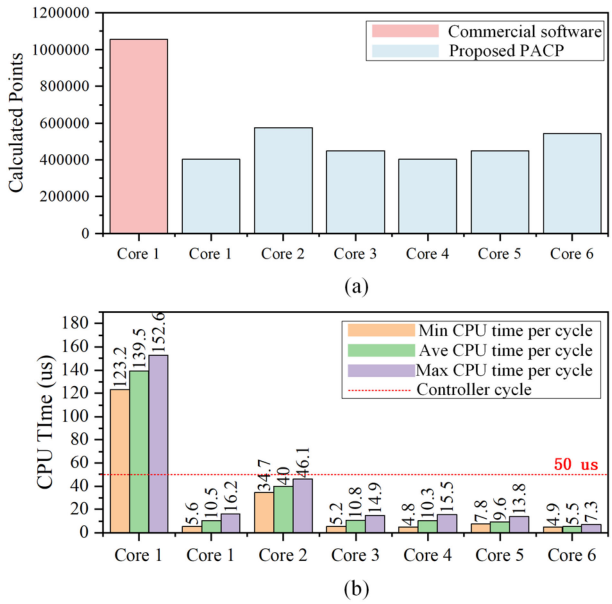


Fig. 11. Comparison of the PACP solver and offline software. (a) Comparison of calculated points. (b) Comparison of calculation time.

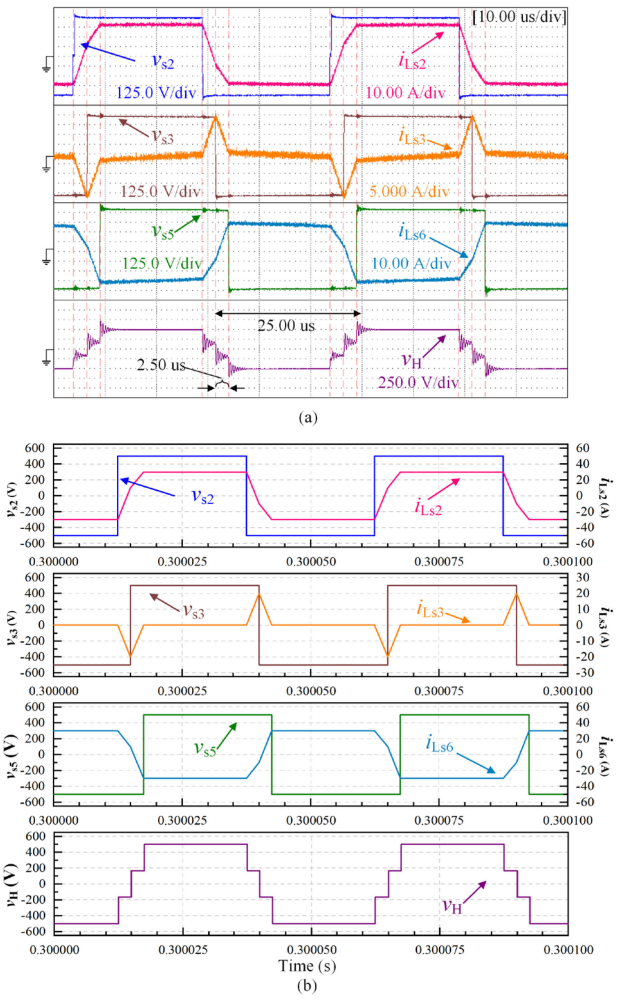


Fig. 12. MMAB current and voltage from experiment and the ED-HIL framework. (a) Experiment result. (b) ED-HIL simulation results.

spurious parameters cause oscillations in the high frequency bus, but have little effect on the port voltages and currents which can be proved in Fig. 12. Generally, the spurious parameters are not considered in the real-time simulation because the stiffness problems caused by spurious parameters can slow down the simulation. Even without spurious parameters, the simulation results of the high frequency switching process in MMAB are still very close to the experimental results.

In this section, the soft-switching function of the three ports of the MMAB is implemented in the experiment and HIL, respectively. It is demonstrated that the performance of the real controller in the virtual simulation of HIL can be consistent with that in the real experiment under the proposed ED-HIL simulation framework. That is, the proposed ED-HIL framework can work properly and have a satisfactory performance compared with the experiments.

D. Comparison of HIL Simulation Performance With Commercial Simulator

To further illustrate the advantages of the proposed simulator, a comparison of the simulation performance of the proposed simulator and the commercial RT simulator is performed.

In this section, the MMAB power distribution experiments are simulated on two *RT* simulators. As mentioned earlier, the isolation of the ports from each other is an advantage of this topology of MMAB. The MMAB power distribution experiments performed take advantage of it. Specifically, when the power of one port changes, the voltage and power of the other two ports remain unchanged except for the power of the port connected to the grid. Fig. 13 shows the current waveform of the H-bridge on the ac side of each port and the output dc of the H-bridge on the other side. Fig. 13(a) shows the results of the commercial real-time simulator and Fig. 13(b) shows the results of the proposed real-time simulator. It can be seen that the simulation results are very close to each other. The commercial simulator outputs the waveform to the oscilloscope via the digital to analog converter, resulting in a slight difference in the waveform. The details of the current variation show that the proposed simulator can achieve the same simulation accuracy as the commercial simulator.

It should be noted that the computing hardware used in this article is a multicore CPU (Intel i7-10700) based on the x86 architecture. The computing hardware used in the commercial real-time simulator is FPGA (Xilinx Virtex 485t). The price of Intel i7-10700 is $300, while the price of Virtex 485t is $6000, the latter is 20 times more than the former. This does not take into account the price of the software on which the commercial real-time simulator depends. It fully illustrates the cost advantage of this real-time simulator, which will be beneficial to the promotion of HIL.

More comparisons are made in this article to compare the simulation scale of the two simulators. The test system is shown in Fig. 14(b). A three-phase two-level circuit is used as a sub-module, as is shown in Fig. 14(a). The sub-module is combined by connecting it in parallel to represent different simulation scales. The scale of the test is varied by changing the number

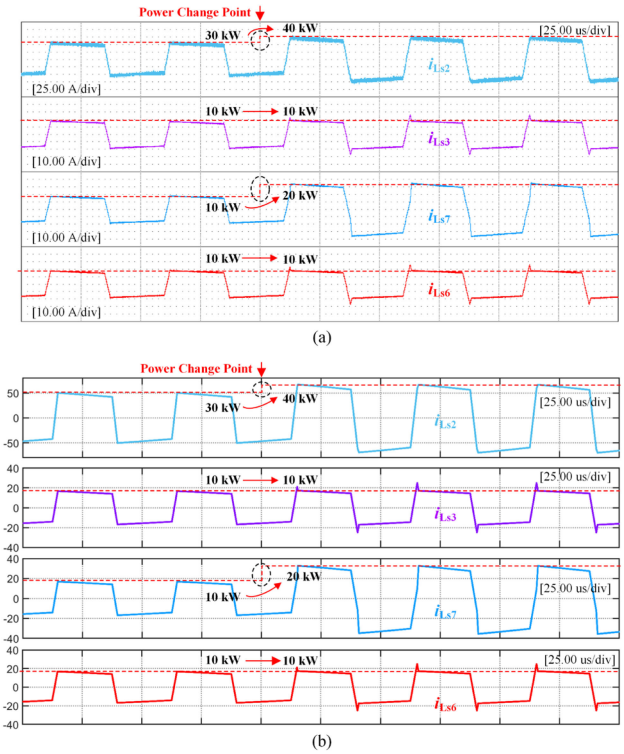


Fig. 13. MMAB current from the ED-HIL framework with commercial RT-HIL simulator and the PACP solver. (a) Commercial RT-HIL results. (b) Proposed RT-HIL results.

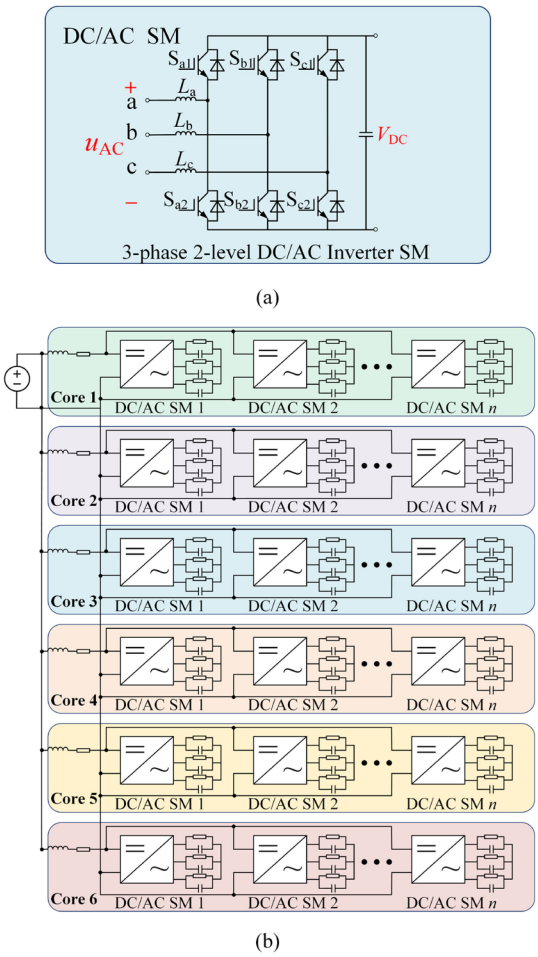


Fig. 14. Test system for simulation scale comparison. (a) Topology of the dc/ac SM. (b) Topology of the whole system.

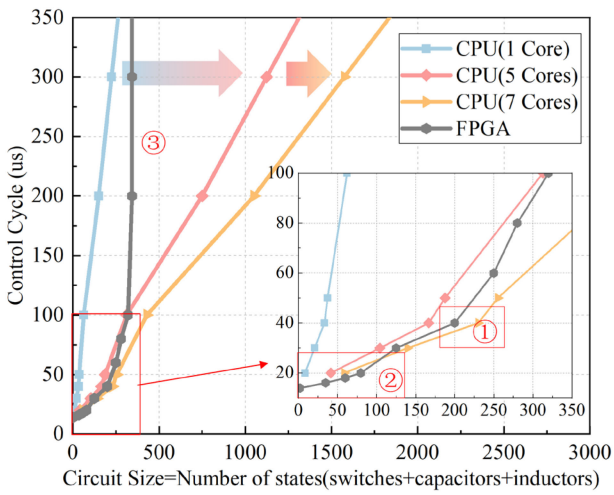


Fig. 15. Comparison on simulated circuit size and control frequency.

of cores performing the simulation and the number of dc/ac simulated on each core. The control system uses PI control to avoid the influence of other factors on the results. The minimum control period (highest control frequency) that can be achieved at each simulation scale is found while satisfying the real-time requirements (the desired waveform can be achieved). The test results are shown in Fig. 15.

It can be seen that the simulation scale of the proposed method is smaller than that of the commercial simulator at any control frequency without circuit partitioning and parallel computing. The simulation can be significantly scaled up by using circuit partitioning and multicore parallel acceleration method. Starting from the control cycle of 40 us, the simulation scale of the proposed method is larger than that of the commercial simulator. In addition, the maximum simulation scale of the commercial simulator is 344 dynamic components, which includes up to 144 switches, due to the hardware computing resource limitation of the FPGA. In comparison, the maximum size of the proposed method in this article is larger, due to the flexibility of CPU computation. Moreover, parallel acceleration does not require additional hardware cost due to the fact that a CPU naturally contains multiple cores.

One drawback of this method is that the proposed method cannot be simulated at higher switching frequencies (switching cycles less than 20 us). There is a delay of 12–15 us in the transfer of the control signal from the real controller to the CPU because of the complex computer architecture on the CPU. This delay essentially prevents the simulator from working at this control frequency. A detailed performance comparison about the two real-time simulators is given in Table II

Overall, the proposed PACP solver is valid for both simple systems and complex systems. For simple circuits, partitioning can reduce the computational burden of each subsystem and

TABLE II
HARDWARE RESOURCE COMPARISONS

	Proposed PACP Solver	Commercial RT Simulator
Hardware Platform	CPU (Intel i7-10700)	FPGA (Xilinx Virtex7 485t)
Hardware Cost	$323.00	$5,244.00
Largest Circuit Scale at f = 5 kHz	1050	344
Largest Circuit Scale at f = 20 kHz	255	225
Highest Switching Frequency (kHz)	50	100
Advantage and Disadvantage	Larger Scale, Lower Cost	Higher Switching Frequency

therefore increase the maximum switching frequency for normal simulations. For more complex systems, the simulation can be scaled up at different switching frequencies.

E. Comprehensive Evaluation

The proposed simulator consists of the ED-HIL framework with PACP solver. It avoids the requirement of expensive computing hardware by obtaining more information from controllers. The above experiments demonstrate that this framework can be effective for HIL simulation. The niche of the proposed simulator is as below. The simulator achieves the same accuracy as system-level offline software through a system-level model, since simplified modeling approaches are not used. By improving the algorithm and using parallel computing techniques, the simulator is able to use a general-purpose CPU. It makes the hardware architecture simple and economical. Therefore, the simulator is easy to use and can be promoted. Moreover, it can realize the simulation scale and switching frequency which could be achieved in FPGA-based HIL simulators.

Specifically, the proposed simulator provides a significant improvement in simulation scale and control frequency compared with commercial CPU-based real-time simulators. Compared with the FPGA-based real-time simulators, the proposed simulator performs better at switching frequencies below 25 kHz. The control frequency of proposed real-time simulator is limited to 50 kHz for high accuracy. For the applications with the higher control frequency (>50 kHz), the FPGA-based commercial simulator is a better choice. The simulation frequency and simulation scale allowed by proposed simulator are sufficient for HIL simulation of most applications.

The configuration of the power electronics circuit affects the selection of energy storage elements and the acceleration ratio of parallel computing simulations. It is important to note that different circuit configurations lead to different parallel acceleration ratios. Therefore, the normal operation of the proposed simulator needs guarantee that the simulation of all subsystems after partitioning can meet the corresponding real-time requirements. For cases that cannot be satisfied, the model of the simulation needs to be simplified.

VI. CONCLUSION

An event-driven HIL simulation framework is proposed in this article to improve the solution speed by predicting switching events and scheduling simulation steps. Further, a PACP solver is proposed to scale up the real-time simulation without simulation distortion. The proposed simulator is deployed to a PC to reduce cost and increase convenience. This article implements real-time HIL simulation for a MMAB containing 32 switches with a switching frequency of 20 kHz. The simulation results are compared with offline simulation, experimental and commercial real-time simulation results. According to the comparison results, it can be found that the proposed simulator can achieve the same accuracy and three times circuit scale of real-time HIL simulation with only 1/16 hardware cost of the commercial real-time simulator. The proposed simulator is based on the general form of the power electronic state equation and therefore can apply to other power electronic circuits. The simulator fills the gap of CPU-based real-time HIL simulation of high-frequency large scale power electronic circuits.

APPENDIX

TABLE III
CONTROL PARAMETERS OF THE FOUR PORTS MMAB

Control Parameter	Symbol	Value
Scenario I		
Port I DC-Capacitor Rated Voltage	V_1	700
Port II DC-Capacitor Rated Voltage	V_2	666.7
Port III DC-Capacitor Rated Voltage	V_3	700
Port III DC-Capacitor Rated Voltage	V_4	700
Scenario II		
Port I DC-Capacitor Rated Voltage	V_1	500
Port II DC-Capacitor Rated Voltage	V_2	500
Port III DC-Capacitor Rated Voltage	V_3	500
MMAB Phase Shift Radio of Port I	d_1	0.0
MMAB Phase Shift Radio of Port II	d_2	0.2
MMAB Phase Shift Radio of Port III	d_3	0.1
Scenario III		
Port I DC-Capacitor Rated Voltage	V_1	700
Port II DC-Capacitor Rated Voltage	V_2	700
Port III DC-Capacitor Rated Voltage	V_3	700
Port III DC-Capacitor Rated Voltage	V_4	700

TABLE IV
SYSTEM PARAMETERS OF THE FOUR PORTS MMAB

Circuit Parameter	Symbol	Value
Port Line Inductance	L_g	2 μH
DC Bus Capacitor	C_{oi}	1 mF
HFT Leakage Inductance	Ls	42 μH
HFT Turns Radio	n_i:1	1:1
Switching Frequency	f_s	20 kHz
SiC MOSFET	S_{11}-S_{84}	CAS120M12BM2

REFERENCES

[1] C. Dufour, V. Jalili-Marandi, J. Bélanger, and L. Snider, "Power system simulation algorithms for parallel computer architectures," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2012, pp. 1–6.

[2] F. Li et al., "Review of Real-time simulation of power electronics," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 796–808, 2020.

[3] L. Herrera, C. Li, X. Yao, and J. Wang, "FPGA-Based detailed real-time simulation of power converters and electric machines for EV HIL applications," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1702–1712, Mar. 2015.

[4] K. Sidwall and P. Forsyth, "Advancements in real-time simulation for the validation of grid modernization technologies," *Energies*, vol. 13, no. 16, Jan. 2020, Art. no. 16.

[5] A. Benigni, T. Strasser, G. De Carne, M. Liserre, M. Cupelli, and A. Monti, "Real-Time simulation-based testing of modern energy systems: A review and discussion," *EEE Ind. Electron. Mag.*, vol. 14, no. 2, pp. 28–39, Jun. 2020.

[6] S. K. Mazumder et al., "A review of current research trends in power-electronic innovations in cyber–physical systems," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 5, pp. 5146–5163, Oct. 2021.

[7] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, "Review of hardware platforms for real-time simulation of electric machines," *IEEE Trans. Transp. Electrific.*, vol. 3, no. 1, pp. 130–146, Mar. 2017.

[8] Y. Zhu, Z. Zhao, B. Shi, and Z. Yu, "Discrete state event-driven framework with a flexible adaptive algorithm for simulation of power electronic systems," *IEEE Trans. Power Electron.*, vol. 34, no. 12, pp. 11692–11705, Dec. 2019.

[9] B. Shi, Z. M. Zhao, Y. Zhu, Z. Yu, and J. Ju, "Discrete state event-driven simulation approach with a state-variable-interfaced decoupling strategy for large-scale power electronics systems," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 11673–11683, Dec. 2021.

[10] B. Shi, Z. M. Zhao, J. Ju, Z. Yu, and Y. Zhu, "Switching transient simulation and system efficiency evaluation of megawatt power electronics converter with discrete state event-driven approach," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2180–2190, Mar. 2022.

[11] A. Benigni and A. Monti, "A parallel approach to real-time simulation of power electronics systems," *IEEE Trans. Power Electron.*, vol. 30, no. 9, pp. 5192–5206, Sep. 2015.

[12] S. Debnath, "Parallel-in-Time simulation algorithm for power electronics: MMC-HVdc system," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 8, no. 4, pp. 4100–4108, Dec. 2020.

[13] D. Shu, Y. Wei, V. Dinavahi, K. Wang, Z. Yan, and X. Li, "Cosimulation of shifted-frequency/dynamic phasor and electromagnetic transient models of hybrid LCC-MMC DC grids on integrated CPU–GPUs," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6517–6530, Aug. 2020.

[14] J. Sun, S. Debnath, M. Saeedifard, and P. R. V. Marthi, "Real-Time electromagnetic transient simulation of multi-terminal HVDC–AC grids based on GPU," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7002–7011, Aug. 2021.

[15] H. Bai, H. Luo, C. Liu, D. Paire, and F. Gao, "A device-level transient modeling approach for the FPGA-Based real-time simulation of power converters," *IEEE Trans. Power Electron.*, vol. 35, no. 2, pp. 1282–1292, Feb. 2020.

[16] A. Rothstein, L. Siekmann, and V. Staudt, "Combined-Hardware-in-the-Loop system and its test-bench verification," in *Proc. Int. Conf. Optim. Elect. Electron. Equip. Int. Aegean Conf. Elect. Mach. Power Electron.*, May 2017, pp. 581–586.

[17] A. Rothstein, L. Siekmann, and V. Staudt, "Realization of AHIL concept on a SoC based FPGA-ARM9 platform for power electronic applications," in *Proc. 11th IEEE Int. Conf. Compat., Power Electron. Power Eng.*, 2017, pp. 689–694.

[18] S. Y. R. Hui and C. Christopoulos, "A discrete approach to the modeling of power electronic switching networks," *IEEE Trans. Power Electron.*, vol. 5, no. 4, pp. 398–403, Oct. 1990.

[19] S. Y. R. Hui and S. Morrall, "Generalised associated discrete circuit model for switching devices," *IEE Proc. Sci., Meas. Technol.*, vol. 141, no. 1, pp. 57–64, Jan. 1994.

[20] K. Wang, J. Xu, G. Li, N. Tai, A. Tong, and J. Hou, "A generalized associated discrete circuit model of power converters in real-time simulation," *IEEE Trans. Power Electron.*, vol. 34, no. 3, pp. 2220–2233, Mar. 2019.

[21] J. Xu et al., "FPGA-Based submicrosecond-level real-time simulation of solid-state transformer with a switching frequency of 50 kHz," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 4, pp. 4212–4224, Aug. 2021.

[22] M. R. Larijani and M. R. Zolghadri, "Design and implementation of an ADC-based real-time simulator along with an optimal selection of the switch model parameters," *Electr. Eng.*, vol. 103, pp. 2315–2325, 2021, doi: 10.1007/s00202-021-01237-1.

[23] M. Matar and R. Iravani, "FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 852–860, Apr. 2010.

[24] S. Cao, N. Lin, and V. Dinavahi, "Flexible time-stepping dynamic emulation of AC/DC grid for Faster-Than-SCADA applications," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2674–2683, May 2021.

[25] J. E. Schutt-Aine, "Latency insertion method (LIM) for the fast transient simulation of large networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 1, pp. 81–89, Jan. 2001.

[26] T. Sekine and H. Asai, "Block-Latency insertion method (Block-LIM) for fast transient simulation of tightly coupled transmission lines," *IEEE Trans. Electromagn. Compat.*, vol. 53, no. 1, pp. 193–201, Feb. 2011.

[27] A. Benigni, A. Monti, and R. A. Dougal, "Latency-Based approach to the simulation of large power electronics systems," *IEEE Trans. Power Electron.*, vol. 29, no. 6, pp. 3201–3213, Jun. 2014.

[28] S. Fan, H. Ding, A. Kariyawasam, and A. M. Gole, "Parallel electromagnetic transients simulation with shared memory architecture computers," *IEEE Trans. Power Del.*, vol. 33, no. 1, pp. 239–247, Feb. 2018.

[29] L.-A. Grégoire, H. F. Blanchette, J. Bélanger, and K. Al-Haddad, "Real-Time simulation-based multisolver decoupling technique for complex power-electronics circuits," *IEEE Trans. Power Del.*, vol. 31, no. 5, pp. 2313–2321, Oct. 2016.

[30] T. Kato, K. Inoue, T. Fukutani, and Y. Kanda, "Multirate analysis method for a power electronic system by circuit partitioning," *IEEE Trans. Power Electron.*, vol. 24, no. 12, pp. 2791–2802, Dec. 2009.

[31] C. Liu, R. Ma, H. Bai, F. Gechter, and F. Gao, "A new approach for FPGA-based real-time simulation of power electronic system with no simulation latency in subsystem partitioning," *Int. J. Elect. Power Energy Syst.*, vol. 99, pp. 650–658, Jul. 2018,.

[32] C. Liu, H. Bai, S. Zhuo, X. Zhang, R. Ma, and F. Gao, "Real-Time simulation of power electronic systems based on predictive behavior," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8044–8053, Sep. 2020.

[33] T. Kato, K. Inoue, and T. Ogawa, "Automatic circuit partitioning for parallel simulation of a power electronic system," in *Proc. IEEE 15th Workshop Control Model. for Power Electron.*, 2014, pp. 1–7.

[34] M. Yushkova, A. Sanchez, and A. de Castro, "Strategies for choosing an appropriate numerical method for FPGA-based HIL," *Int. J. Elect. Power Energy Syst.*, vol. 132, Nov. 2021, Art. no. 107186.

[35] Z. Li, J. Xu, K. Wang, G. Li, and P. Wu, "A discrete small-step synthesis real-time simulation method for power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3667–3676, Apr. 2022.

[36] Y. Li, R. Chen, B. Sensale-Rodriguez, W. Gao, and C. Yu, "Real-time multi-task diffractive deep neural networks via hardware-software co-design," *Sci. Rep.*, vol. 11, no. 1, May 2021, Art. no. 1.

[37] Z.-H. Zhan et al., "Matrix-Based evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 2, pp. 315–328, Apr. 2022.

[38] Z. Yu, Z. Zhao, B. Shi, Y. Zhu, and J. Ju, "An automated Semi–symbolic state equation generation method for simulation of power electronic systems," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 3946–3956, Apr. 2021.

Jialin Zheng (Student Member, IEEE) received the B.S. degree in electrical engineering in 2019 from Beijing Jiaotong University, Beijing, China. Since 2019, he has been working toward the Ph.D. degree in electrical engineering at the Department of Electrical Engineering, Tsinghua University, Beijing, China.

His research interests include simulation of power electronic systems, modeling of power semiconductor devices, and modeling for high-capacity power electronics devices.

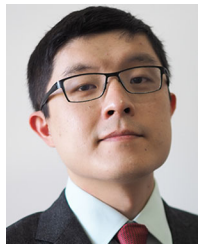
Zhengming Zhao (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from Hunan University, Changsha, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 1991.

He is currently a Professor with the Department of Electrical Engineering, Tsinghua University. His research interests include high-power conversion, power electronics and motor control, and solar energy applications.

Yangbin Zeng (Member, IEEE) received the B.Sc. degree in building electrical and intelligent from Xiangtan University, Xiangtan, China, in 2015, the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2021.

He is currently a Postdoctoral Researcher with the Department of Electrical Engineering, Tsinghua University, Beijing, China. His current research interests include real-time simulation techniques, power router, and soft-switching techniques.

Dr. Zeng was the recipient of the IEEE International Future Energy Challenge "Innovation Award" Prize as the student leader (2018).



Shiqi Ji (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2010 and 2015, respectively.

Since 2015, he has been joined the Center for Ultra-Wide Area Resilient Electric Energy Transmission Networks, The University of Tennessee, Knoxville, TN, USA, and became a Research Assistant Professor, in 2019. Since 2020, he has been with Tsinghua University, as an Assistant Professor, and then became an Associate Professor in 2021.

He has authored or coauthored more than 50 technical articles. His research interests include semiconductor device modeling, medium-voltage and high-power converter design, high-voltage SiC device characterization and application techniques, and grid-connected converter design.

Liqiang Yuan (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1999 and 2004, respectively.

In 2008, he was an Associate Professor with the Department of Electrical Engineering, Tsinghua University, where he is currently a Full Professor. His current research interests include the application techniques of semiconductor devices, solid-state transformer, and high-power converters.