

MPSoC-Based Dynamic Adjustable Time-Stepping Scheme with Switch Event Oversampling Technique for Real-time HIL Simulation of Power Converters

Jialin Zheng, *Student Member, IEEE*, Yangbin Zeng, *Member, IEEE*, Zhengming Zhao, *Fellow, IEEE*, Weicheng Liu, *Student Member, IEEE*, Han Xu, *Student Member, IEEE*, Haoyu Wang, and Di Mou, *Member, IEEE*

Abstract—Hardware-in-the-loop (HIL) simulation is widely used in the design and test process of power converters. However, the conventional fixed-step HIL simulations are accompanied by significant computational costs and struggle to accurately locate and calculate switching events in high-frequency applications. Therefore, this paper proposes a dynamic adjustable time-stepping (DAT) simulation scheme with a switch event oversampling (SEO) technique. In details, the SEO technique uses oversampled control signals and simulated modulation processes to predict all switching events information accurately throughout a control cycle. Meanwhile, with the switching events information, the DAT scheme can adjust the simulation time-stepping in advance, ensuring simulation accuracy by adjusting the order of numerical algorithm. Furthermore, the iterative calculation can be used to locate the natural commutation of diodes accurately due to the flexible time-stepping and order. The global performance and the hardware utilization of the proposed scheme are evaluated by numerical experiments and HIL experiments in a 32-switches modular multi active bridge converter with a switching frequency of 400 kHz. The results demonstrate that the proposed scheme can improve the simulation accuracy by 42 times while reducing 90% of the computation memory compared with commercial HIL simulator.

Index Terms—Hardware-in-the-loop (HIL), real-time systems, power converters, multi-processor system-on-chip (MPSoC).

I. INTRODUCTION

HARDWARE-IN-THE-LOOP (HIL) simulation has become a popular technology for developing control systems for power converters [1]–[3]. HIL simulation provides a fully digital test bench by connecting the real-time simulator to the real controller under test, instead of relying on the power-level environment in traditional development [4]. Therefore, HIL simulation offers advantages in terms of hardware cost, development speed and test reliability [5].

The primary goal of HIL simulators is to achieve a digital replica that closely resembles the real system, which requires the HIL simulator input/output to be as close as possible to the

real system [6]. However, due to the unsynchronized clocks of the simulator and the controller [7], the HIL simulator can only obtain input signals and calculate output results at simulation step intervals. It results in inaccurate detection of switching events and negative effects on the output accuracy (such as non-system oscillations, etc.) [8]. Thus, a small step size is crucial for HIL simulation of high-switching-frequency (HSF) applications like LLC and DAB [9], [10]. The typical switching frequency range of these converters is from 20kHz to 400kHz [11], requiring a nanosecond-level simulation step-size to provide high-accuracy input/output [12].

Currently, mainstream HIL simulators use embedded hardware such as field-programmable gate arrays (FPGAs) and system-on-chips (SoCs) to provide ultra-fast computation capability for the HSF applications [13], [14]. However, the computational resources of the embedded hardware are still limited by chip space and manufacturing processes [15]. Meanwhile, the demand of nanosecond-level step-size response brings about significant computational costs [16]. Therefore, the main challenge faced by HIL simulators for HSF applications is the conflict between the need to improve input/output precision and the heavy computation costs with the limited hardware resources. To tackle this challenge, three key parts of HIL simulation, namely the modeling, algorithm, and sampling, can be targeted for improvement.

1. Model simplification: The switch model can accurately characterize system-level behavior using large and small resistances [17]. However, the changing resistances lead to time-varying system matrices, requiring significant extra computational costs [18]. To simplify the model, methods such as pre-stored matrices and optimized matrix inversion are proposed [19], [20]. However, these methods heavily rely on hardware resources and use serial matrix operations, making it hard to meet real-time constraints in HSF applications. In addition, although the associate discrete circuit model can reduce simulation step-size by keeping the system matrix constant, it may introduce virtual errors [21]. To reduce errors, related optimized models are proposed through parameter optimization and discrete small steps [22], [23]. However, these models are still limited in the accuracy of HSF applications, as errors increase with higher switching frequencies. [24]

2. Algorithm acceleration: The computational cost of HIL solvers generally increases as the scale of the power

This work was supported by the National Natural Science Foundation of China under Grant 52207209, and funded by China Postdoctoral Science Foundation under Grant 2021M701844. (Corresponding author: Yangbin Zeng.)

J. Zheng, Y. Zeng, Z. Zhao, W. Liu, H. Xu, H. Wang, and D. Mou, are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: zhengjl19@mails.tsinghua.edu.cn, ybzeng@tsinghua.edu.cn, zhaozm@tsinghua.edu.cn; liuwc21@mails.tsinghua.edu.cn, xuhan21@mails.tsinghua.edu.cn, hw2972@columbia.edu, dimou428@126.com.)

converters increases [25]. Therefore, parallel techniques are often employed to reduce simulation step sizes in situations with high computational costs [25]–[28]. One type of parallel technique is to decouple the system using specific components like switches and storage elements, and further compute each subsystem in parallel [26], [27]. Another type of parallel technique is to implement parallelization in the solving algorithm itself, using methods such as inserting delays and utilizing predictive information to solve state variables and all components in parallel [25], [28]. Overall, both parallel techniques can reduce the simulation step size, but this comes with the cost of sacrificing numerical stability or requiring more hardware resources [29], [30].

3. Oversampling technique: The minimum step sizes for state-of-the-art commercial and academic HIL simulators are in the range from hundreds of nanoseconds to tens of nanoseconds [12], [16], [31], which are sufficient for maintaining output accuracy. However, smaller sampling step-size is required to improve the accuracy of input signal. Thus, the oversampling technique samples multiple times using the smallest resolution that can be achieved by the hardware [8], [32]. The key issue with oversampling is how to use the additional information for further calculations. Based on oversampled information, one approach uses interpolation/extrapolation or even recomputation of state variables to reduce negative effects caused by input errors [6], [33]. However, the computation costs and response time increase with the number of switching events, making it difficult for HSF applications. Another method that HIL simulator can use is time-averaging method [10], [34]. It reads all switching information for one cycle and generates an averaged switching information for the next step input. This method can handle multiple switching events within one step, but high-frequency switching harmonics may be lost [34].

Overall, these methods can reduce the sampling and simulation step sizes of HIL simulators but require the model precision sacrifice, additional hardware resources, etc. Therefore, these methods cannot completely solve the aforementioned challenge. The common point of these HIL simulators is that they use fixed sampling and simulation step, making it hard to simultaneously improve input/output accuracy and reduce computation costs. Even with oversampling techniques, fixed-step HIL simulators is hard to fully utilize the oversampling information. Additionally, fixed-step HIL simulators have difficulties in handling the natural commutation processes directly, as determining natural commutation state accurately requires iterative calculations [12]. Therefore, the main idea of this paper is to develop a scheme that fully utilizes oversampling information and can adaptively adjust simulation step sizes for better performance.

In this paper, a switch event oversampling (SEO) technique is proposed to enhance sampling performance. By oversampling control signals and simulating modulation processes of the controller, it can predict all the switching events within an entire cycle. It can also use the known information to achieve cycle synchronization between the

simulator and the controller, reducing the limitation on simulation step sizes. Moreover, a dynamic adjustable time-stepping (DAT) scheme is proposed, which adjusts the step sizes according to the switching event obtained from the SEO technique. As the step-size and integration order are adjustable, the DAT scheme can accurately locate and calculate the natural commutation process. The DAT scheme only needs to compute at the switching event moment, significantly reducing the requirement of hardware resources and computational costs compared with small fixed-step HIL simulators. Therefore, the proposed DAT scheme can simultaneously improve the accuracy of HIL simulator input/output and reduce computational costs with limited hardware resources.

The main contributions of this paper are as follows:

1. An SEO sampling technique which can capture switching signal actions accurately in HSF applications and reduce limitations on simulation step size.
2. A variable-step DAT simulation scheme, which improves the output accuracy while reducing computation costs and supports simulation of natural commutation processes.
3. A multi-processor system on chip (MPSoC) implementation, including computing tasks allocation in the heterogeneous architecture, flexible step scheduling and parallel acceleration of matrix calculation.

The rest of the paper is organized as follows. Section II introduces the SEO sampling technique. Section III explains how the DAT scheme works. Section IV details the implementation of the above scheme on the MPSoC hardware. Section V provides the HIL experiment on a multi-active-bridge case and compares it with other methods to evaluate the global performance and the hardware utilization of the proposed scheme. Finally, Section VI presents the conclusion.

II. SWITCH EVENT OVERSAMPLING TECHNIQUE

A. Sampling Error of HIL Simulators

HIL simulators typically perform sampling at the beginning of each step, as shown in Fig. 1(a). In this one-step sampling situation, the sampling step-size equals the simulation step-size, and is limited by the simulation computation time. Besides, due to the unsynchronized clocks of the simulator and the controller, the sampling time of the simulator rarely synchronizes with the time when the controller gate signal changes (known as a switching event). Therefore, the simulation can only detect switching events after they have occurred, resulting in the one-step delay issue.

Specifically, the sampled gate signal (PWM_{HIL}) is obtained by regular interval sampling of the controller's gate signal PWM with a sampling step of T_{disc} , which typically ranges from tens to hundreds of ns. The sampling delay between PWM_{HIL} and PWM includes the turn-on delay D_{on} and the turn-off delay D_{off} . The absolute error range of these delays caused by sampling ranges from 0 to T_{disc} , indicating that the sampling step size is a crucial factor in determining the step minimum possible error of PWM_{HIL} . Furthermore, mutative sampling delays lead to inaccurate duty cycles. T_{sw} represents

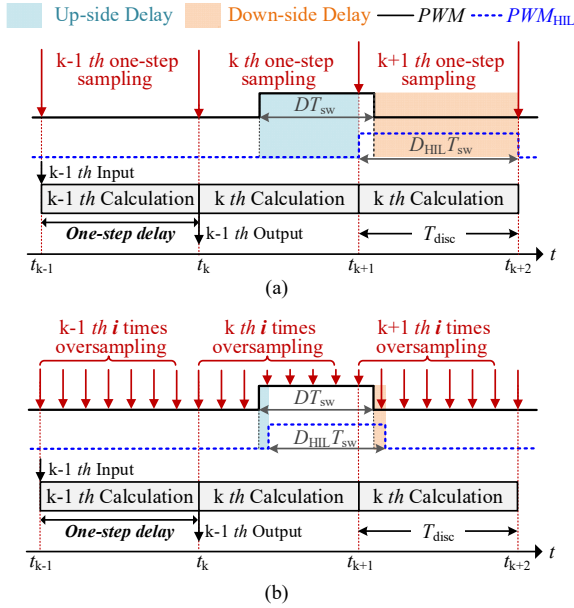


Fig. 1. Sampling error of HIL simulators. (a) one-step sampling situation. (b) oversampling situation. (The duty cycle of the PWM is for illustrative purposes only.)

the control cycle, and T_{on} represents the turn-on time of PWM , then the original duty cycle D and sampled duty cycle D_{HIL} of the gate signal can be expressed as

$$\begin{cases} D = T_{on} / T_{sw} \\ D_{HIL} = (T_{on} + D_{off} - D_{on}) / T_{sw}. \end{cases} \quad (1)$$

Compared to the original duty cycle, the absolute error upper bound of the sampled duty cycle is given by,

$$|D_{HIL} - D| < T_{disc} / T_{sw}. \quad (2)$$

According to Eq. (2), the sampling process inevitably confuses the original behavior of gate signals due to the finite discrete step size. The simulator uses the sampled gate signal for calculations, resulting in sampling errors transferring to the output errors. To decrease sampling errors within the finite simulation step size, oversampling techniques are commonly employed. As shown in Fig. 1(b), this technique involves sampling the gate signals i times within each simulation step. By doing so, the maximum absolute error range of sampling delay is reduced to T_{disc}/i , while the absolute error of duty cycle is reduced to:

$$|D_{HIL} - D| < T_{disc} / i \times T_{sw}. \quad (3)$$

However, the oversampling technique faces difficulties in utilizing the oversampled information. As shown in Fig. 1.b, the extra $i-1$ times of sampling information is difficult to be used in the simulation process. In addition, traditional methods, whether one-step sampling or oversampling, suffer from one-step delays due to the time required to complete the simulation computation. Smaller discretization step-size is needed to reduce the impact of one-step delays, but result in more computation points and less computation time per step, which increases the amount of meaningless computation and limits the complexity of the simulation.

B. Switch event oversampling Technique

To reduce the limitation of the simulation step size by one-

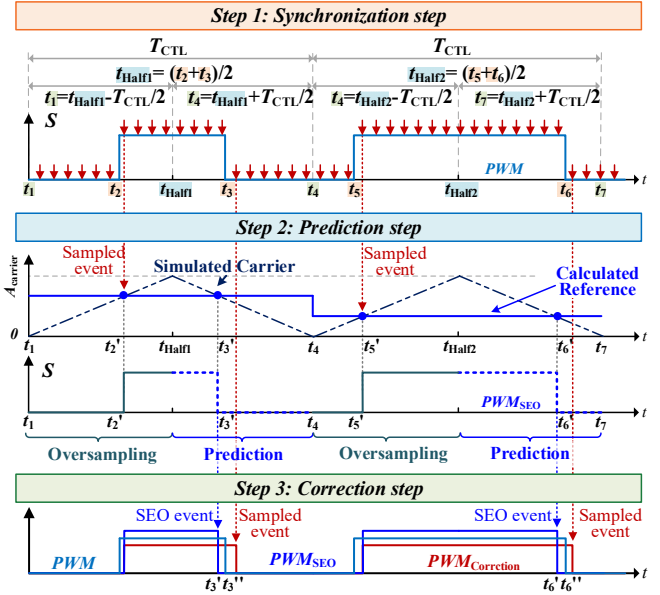


Fig. 2. SEO technique for the symmetric modulation case.

step delay, the paper proposes a switching event oversampling (SEO) technique. This SEO technique can be used for various modulation cases and is specifically optimized for the widely used symmetric modulation case. In the symmetric modulation case, the SEO sampling technique utilizes the symmetry of the modulation to predict the switching events and achieve cycle synchronization between the simulator and controller, resulting in calculation results without one-step delay. Specifically, the working principle of the SEO technique is shown in Fig. 2, which mainly includes three steps.

1. Synchronization step: the SEO technique uses the sampled gate signals to obtain the timestamp of the controller control cycle. For common SPWM modulation, the symmetry of the triangle wave modulation enables the simulator to find the midpoint of the control cycle by detecting the rising and falling edges of the gate signal. The SEO technique can then achieve cycle synchronization with the controller based on the known switch frequency. For phase-shift control, a reference driving signal can be chosen for synchronization. In addition, better synchronization performance can be achieved by directly providing a clock signal from the controller, if conditions allow.

2. Prediction step: the simulator's discrete step size can be chosen as the control cycle due to the synchronization step. The first half of the step size is referred to as the prediction step, where oversampling is utilized to obtain all switching events of the previous half control cycle, as shown in Fig. 2. The modulation process can be simulated using the previous half of the switching events and modulation information (e.g., carrier frequency and peak-to-peak value) to obtain the reference wave, which is constant within each control cycle. Then, the switching events of the latter half of the cycle can be predicted using the simulated reference wave. Therefore, the complete switching event sequence, including the occurrence time and state, can be obtained in half of a discrete cycle.

3. Correction step: the SEO technique utilizes oversampling

in the latter half of the cycle. The sampled switch events are then compared with the predicted switching events from step 2 to verify the accuracy of the prediction.

In conclusion, the SEO technique has the same sampling accuracy as general oversampling techniques for the first half of the cycle, as is shown in Fig. 2. As for the latter half of the cycle, the error of the two techniques is within one sampling step, which means that there is little change in the sampling accuracy. It can be found that the SEO technique avoids the one-step delay those conventional oversampling techniques would face in the symmetric modulation case, and ensures that the controller can obtain the proper and correct simulator results. Thus, the discretization step size can be extended from 1/20-1/100 of the control cycle to the size of one control cycle [35]. Such a large discretization step can help reduce the number of computation points and facilitate the use of computational resources for the necessary computation points.

C. Switching Event Sequence

For the asymmetric modulation case, the SEO technique can still be applied. Since this modulation case has no symmetry to be exploited, the SEO technique uses the basic oversampling scheme, as shown in Fig. 3. In this case, the key is to efficiently utilize the additional sampling signals. If all the gate signals are utilized as in the one-step sampling way, the large number of sampled points would lead to heavy computational costs. Meanwhile, averaging the oversampled signals would result in the loss of high-frequency information.

To balance computation cost and simulation accuracy, a switching event sequence is proposed in this paper to transmit information between the sampling part and the calculation part. It selects the timing and states of all switching events from all sampling signals of a discrete step.

$$s = \{e_1(t_1, state_1), e_2(t_2, state_2), \dots, e_k(t_k, state_k)\} \quad (4)$$

The oversampling method using a switching event sequence can be summarized in three steps, as is shown in Fig. 3,

1. During the computation step from t_{k-1} to t_k , perform i times samplings, and record the occurrence time gate signal's changing moment, which is the switching event e_1 and e_2 .
2. At the beginning of the step from t_k to t_{k+1} , arrange all the switching events (e_1 and e_2) obtained from Step 1 to form a switching event sequence.
3. During the computation steps from t_k to t_{k+1} , the simulation calculation can flexibly arrange the calculation sub-steps based on these switching events (e_1 and e_2).

Nevertheless, for the case of asymmetric modulation, the SEO technique has a one-step delay between the simulated output and the sampled input. To ensure that the simulator output is acceptable, the discrete step size and one-step delay of the simulation need to be limited. Since oversampling provides a sufficiently accurate input signal, its discrete step size can be larger than that of the one-step sampling method, with a typical value of $1\mu s$ [10]. In addition, the switching event sequence is compatible with all the modulation case mentioned above, and the following simulation scheme is designed based on this sequence.

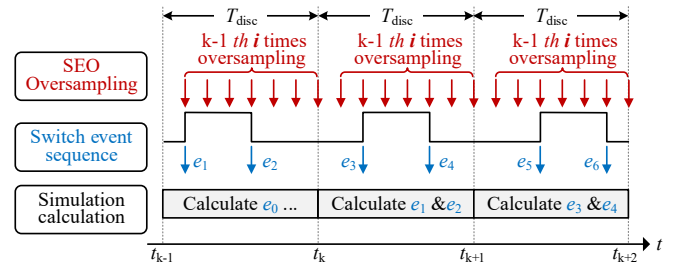


Fig. 3. SEO technique for the asymmetric modulation case. (The duty cycle of the PWM is for illustrative purposes only.)

III. DYNAMIC ADJUSTABLE TIME-STEPPING SCHEME

This section describes the dynamic adjustable time-stepping (DAT) scheme based on switching event sequences, including the model generation, variable-step numerical algorithms and natural commutation processing.

A. Model Generation with Switch Separation

In the k th simulation interval, the model of a power electronic system (PES) can be represented by a set of state equations and output equations,

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (5)$$

where $x(t)$ represents the state variables, $u(t)$ represents the input variables, $y(t)$ represents the output variables, and the system matrices A , B , C , and D represent the corresponding topology connections. These system matrices are updated based on switching actions. However, online generation of these matrices is challenging in terms of real-time constraints, especially for large-scale PES.

To solve this, the switch modules of a PES can be replaced with the following equivalent models. Specifically, the equivalent model of switch devices, a half-bridge module shown in Fig. 4(a) is taken as an example. The equivalent model in Fig. 4(b) includes two diodes, two voltage sources, one current source, and one on-resistance. The controlled sources are represented by switch functions as follows:

$$\begin{cases} v_{AC+} = s_1 \cdot v_{DC}, v_{AC-} = s_2 \cdot v_{DC} \\ i_{DC} = s_1 \cdot i_{DC+} + s_2 \cdot i_{DC-} \end{cases} \quad (6)$$

where v_{dc} , i_{dc+} and i_{dc-} can be obtained from the linear system, and the gate signals s_1 and s_2 can be obtained through the SEO technique mentioned earlier. v_{ac+} and v_{ac-} represent the AC-side voltage, and i_{dc} represents the DC-side current.

The matrix form of the switch function can be written as:

$$u_s = \begin{bmatrix} v_{AC} \\ i_{DC} \end{bmatrix} = \begin{bmatrix} K_{V,k} & \\ & K_{I,k} \end{bmatrix} \begin{bmatrix} v_{DC} \\ i_{AC} \end{bmatrix} = K_k \cdot y_s \quad (7)$$

where the vector u_s represents the equivalent nonlinear voltage and current sources, K_k denotes the switch matrix of the k th simulation step, and y_s represents the measurement vector of the equivalent source values.

The model introduces the equivalent input u_s into the state space, causing the original system matrix of the PES to become a combination of the switch matrix K_k and the linear system. As a result, only the K_k needs to be generated based

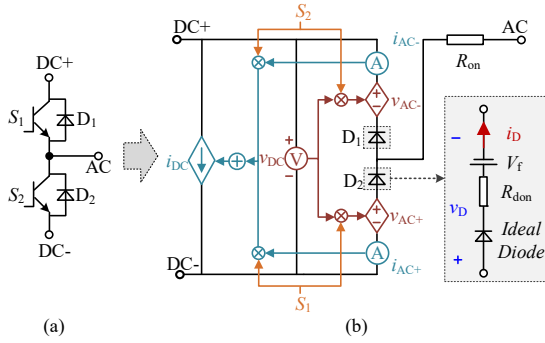


Fig. 4. Switch Equivalent Model with diode.

on switch events, while other system matrices no longer require generation.

The state space equation is separated to avoid unnecessary multiplication of zero elements in the switch matrix K_k . After separation, the state space equation of the entire circuit model can be expressed as follows:

$$\begin{cases} \dot{x} = A_0 x + [B_1 & B_2] \left(\begin{bmatrix} 0 \\ u \end{bmatrix} + \begin{bmatrix} K_k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_s \\ 0 \end{bmatrix} \right) \\ \begin{bmatrix} y \\ y_s \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \left(\begin{bmatrix} 0 \\ u \end{bmatrix} + \begin{bmatrix} K_k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_s \\ 0 \end{bmatrix} \right) \end{cases} \quad (8)$$

where u is an $m \times 1$ vector, u_s is an $n \times 1$ vector, and y_s is an $n \times 1$ vector. Simplification leads to the following state equation, output equation, and measurement equation:

$$\dot{x} = A_0 x + B_2 u + B_1 K_k y_s \quad (9)$$

$$y = C_1 x + D_{12} u + D_{11} K_k y_s \quad (10)$$

$$y_s = C_2 x + D_{22} u + D_{21} K_k y_s \quad (11)$$

B. Numerical Algorithm of the DAT Scheme

Step-size estimate: To efficiently utilize the switching event sequence, a numerical algorithm for the proposed DAT scheme is presented in Fig. 5. Specifically, a switching events interval Δt_s can be expressed:

$$\Delta t_s = t(e_k) - t(e_{k-1}) \quad (12)$$

In the case of symmetric modulation, considering the situation that no switching events occur within a switching period, the maximum simulation time-step Δt_{\max} is set to the switching period, which ensures the correct data interaction with the controller. For asymmetric modulation, the Δt_{\max} is typically chosen as small as possible, as single-step delays must be considered. The simulation step size can be arranged by finding the most urgent time intervals of maximum simulation time-step Δt_{\max} and all the switching events intervals.

Matrix inversion: In each simulation step, the state space equations (Eqs. 9-11) needs to be solved. Notably, the measurement equation (Eq. 11) is implicit and is not suitable using standard methods like Newton's iteration, as it results in a substantial increase in computational cost and uncertain iteration times. To address this, this paper reorganizes the Eqs. (9-11) as follows:

$$y_s = \underbrace{(I - D_{21} K_k)^{-1}}_{Q_k} (C_2 x + D_{22} u). \quad (13)$$

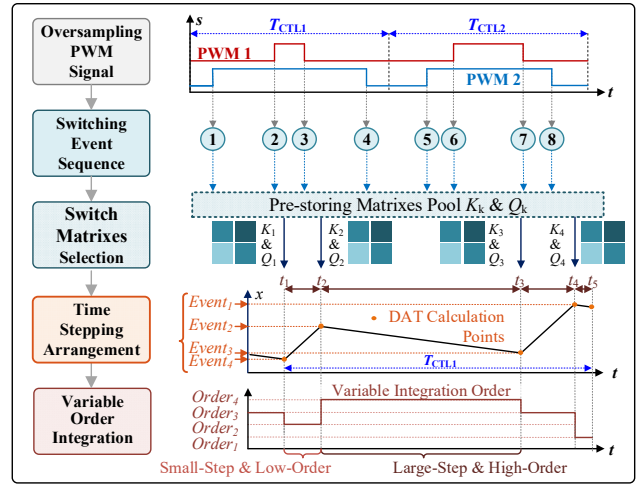


Fig. 5. The numerical algorithm for the DAT scheme.

$$\dot{x} = \underbrace{(A_0 + B_1 K_k Q_k C_2)}_{A_k} x + \underbrace{(B_2 + B_1 K_k Q_k D_{22})}_{B_k} u \quad (14)$$

$$y = \underbrace{(C_1 + D_{11} K_k Q_k C_2)}_{C_k} x + \underbrace{(D_{12} + D_{11} K_k Q_k D_{22})}_{D_k} u. \quad (15)$$

To simplify the notation, the matrices in Eqs. (14-15) are represented by A_k , B_k , C_k , and D_k in the following analysis. To better understand the meaning of these matrices, an example of a full-bridge circuit is given in Appendix.

To avoid the computation costs of matrix inversion, this paper pre-stores the switch matrix K_k and its inverse matrix Q_k . Compared to storing all system matrixes, this approach significantly reduces memory pressure but slightly increases online computation costs. These calculations can be performed during the prediction step of SEO technique and can be used for numerical integration during the correction step.

Numerical integration: The numerical integration can be performed by discretizing the state equation through a Taylor approximation, including most well-known discretization methods such as Euler, Trapezoidal, and Runge-Kutta methods. The p th order Taylor approximation of the state variables at time $t=k$ can be expressed as:

$$x_{k+1} = x_k + \sum_{i=1}^p \frac{x_k^{(i)}}{i!} (\Delta t_k)^i + \underbrace{\frac{x_k^{(p+1)}}{(p+1)!} (\Delta t_k)^{p+1}}_{R_n} \quad (16)$$

where Δt_k represents the simulation step size at time $t=t_k$, $x_k^{(i)}$ represents the i th derivative of state variable x at time $t=t_k$, and R_n denotes the truncation error.

In each time step, the PES system can be regarded as a Linear Time-Invariant (LTI) system, with more details found in [36]. Therefore, the superposition theorem and time-invariance properties of LTI systems can be applied to derive the various order derivatives of the state variable from Eq. 14, and can be expressed as,

$$x_k^{(i)} = A_k x_k^{(i-1)} + B_k u_k^{(i-1)}, i \geq 1 \quad (17)$$

where $u_k^{(i-1)}$ represents the $(i-1)$ th derivative of the system input variable, which can be easily obtained from the explicit expression of the input source.

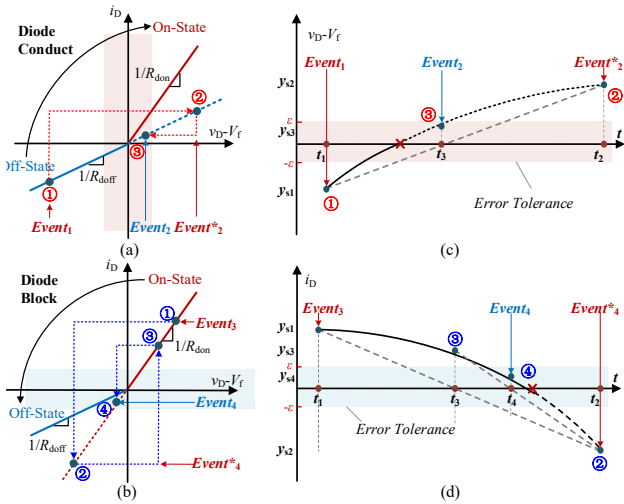


Fig. 6. Natural commutation processing of the DAT scheme. (a) the turn-on process of diodes. (b) the turn-off process of diodes. (c) the location of the turn-on diode event. (d) the location of the turn-off diode event.

Based on the truncation error shown in Eq. (16), a dynamic adjustment method of simulation step size and the integration order can be added in the DAT scheme. First, the simulation time step is estimated based on switching information, and the corresponding system matrix is loaded from storage. Second, the time derivative of the state variables $x(t)$ is calculated using Eq. (17), based on the time step and matrix information. Third, using the i -th order derivative of the state variables, the i -th term in the Taylor series of the state variable $x(t)$ is calculated, referred to as the i -th order increment $\Delta x_{k,i}$.

$$\Delta x_{k,i} = \frac{x_k^{(i)}}{i!} (\Delta t_k)^i \quad (18)$$

The increment can also be used to select the integration order that satisfies the accuracy requirements. Finally, after evaluating all state variables, the optimal order is determined and numerical integration is performed according to Eq. (16).

Natural Commutation Processing: The turn-off process and turn-on process of the diode are shown in Fig. 6(a) and Fig. 6(b), respectively. Unlike the externally controlled switching events, diode events (conduction or turn-off) passively change with the system state and are difficult to locate directly. Therefore, the monitoring equation should be set up to monitor the diode state changes, and can be expressed as follow,

$$|y_d(t) - \delta_{\text{bias}}| \leq \varepsilon \quad (19)$$

where $y_d(t)$ represents the diode current in the turn-off process, and the diode voltage in the turn-on process, δ_{bias} equals zero in the turn-off process, and equals V_f in the turn-on process, and ε represents the error tolerance.

As the derivatives of the state and input variables have been obtained in the numerical integration part, the Taylor approximation of the diode variable y_d can be easily derived. The q th Taylor approximation of y_d at time t_k is expressed as:

$$y_{d,k} = y_{d,k-1} + \sum_{i=1}^q \frac{y_{d,k-1}^{(i)}}{i!} (\Delta t_k)^i + \underbrace{\frac{y_{d,k-1}^{(q+1)}}{(q+1)!} (\Delta t_k)^{q+1}}_{R_{d,n}} \quad (20)$$

where $R_{d,n}$ denotes the truncation error of $y_{d,k}$.

Therefore, the diode event can be found by solving the root of the monitoring equation. The selection of tolerance directly affects the number of iterations, and it is recommended to be as small as possible to improve accuracy while meeting the real-time constraints in the DAT scheme. To meet the real-time constraints of simulating large-scale systems, linear interpolation can be used to locate diode events between switch events 1 and 2, as shown in Fig. 6(c). With only one operation, the diode conduction event can be located, and the diode switch state changes from (1)-(2)-(3). For small-scale PES with higher accuracy requirements, the secant method [36] can be applied between switch events 3 and 4, as illustrated in Fig. 6(d). It can be observed that locating the diode event requires two iterations, with the diode switch state changing from (1)-(2)-(3)-(4). The new diode events are then inserted into the switch event sequence, and the time step needs to be adjusted accordingly.

C. Discussions

The primary focus of this paper is designing disparate time-stepping ways for the different requirements of sampling and simulation. This allows for the efficient reduction of the sampling step to the hardware's limit, enhancing the accuracy of sampled switching events with minimal computational cost. Simultaneously, the simulation step can be flexibly adjusted based on the switching events, significantly reducing the number of computation points while utilizing high accuracy integration algorithms. Key contributions include a switching event prediction technique and a dynamic step-size adjustment integration method. The relaxed time constraint enables the simulation of natural commutation processes, facilitating HIL simulation for dead time and blocking scenarios.

In terms of sampling, the SEO technique uses the PWM signal of the first half switch cycle to predict the PWM signal of the latter half cycle, thereby overcoming the one-step delay between input and output in traditional HIL simulation. While the SEO technique samples the first half-cycle, the DAT can generate the corresponding matrix based on the switching information and perform the integration calculation in the latter half-cycle. Meanwhile, in terms of computation, the switch model of DAT considers the diode behavior, which incurs higher computational costs than typical $R_{\text{on}}/R_{\text{off}}$ models.

Despite this, the proposed scheme requires far fewer computational points than traditional fixed-step simulation. Taking Fig. 5 as an example, the typical step size of a traditional HIL simulation for an application with the switching frequency of 20kHz is 100ns, requiring 500 calculation points within a switching period (T_{ctl1}). In contrast, the DAT solution only calculates 5 points in T_{ctl1} , including the controller (t_5). Due to the substantial reduction in the number of calculation points, it can be inferred that, compared to traditional approaches, the proposed method significantly reduces additional hardware resource requirements and improves computational efficiency.

In addition, the DAT scheme adopts the Taylor series integration method, which possesses stability similar to that of

the explicit Runge-Kutta method. Although stability can be enhanced by increasing the order of integration, its numerical stability is still inferior to that of implicit methods, such as backward differentiation formulas (BDFs), and Adams-Moulton methods [36]. Therefore, when dealing with stiff systems containing numerous stray parameters, one approach is to first reduce the system's stiffness, for instance, by ignoring less significant stray parameters and applying the proposed DAT scheme, while another alternative is to utilize other implicit integration methods.

IV. MPSoC HARDWARE IMPLEMENTATION

A. MPSoC Hardware Platform

The Zynq[®] UltraScale+[®] XCZU5EV MPSoC is selected as the computational hardware for the proposed DAT scheme. This platform integrates the processing system (PS) of ARM's central processing unit (CPU) and the programmable logic (PL) of Xilinx's FPGA on a single chip, as shown in Fig. 7(a). The PS side includes four 1.5 GHz Cortex[®]-A53 processor cores, two 600 MHz Cortex[®]-R5 processor cores, and on-chip memory and peripheral devices. The PL side contains a large number of configurable logic blocks (CLBs). The chip also contains 12 128-bit data-width advanced extensible interface (AXI[®]) buses for low-latency and high-bandwidth communication between the PS and PL sides. This MPSoC architecture can provide highly flexible sequential computing capabilities as well as significant parallel acceleration capabilities.

The simulation platform for the DAT scheme is presented in Fig. 7(b). The MPSoC board is connected to the TI[®] 34H84 DAC board to convert the simulation results into analog signals, which are then obtained through Yokogawa[®]'s waveform recorder. A Zynq[®]-7000 board is chosen as the physical controller, and data communication between the controller and the MPSoC simulator is achieved through the small form-factor pluggable (SFP/SFP+) high-speed fiber optic interface. The simulation scheme is compiled and programmed onto the MPSoC board using Xilinx[®]'s Vitis[®] and Vivado[®] software on the host PC. The PS side is implemented through C++ code, while the PL side is implemented through VHDL code after being processed by the High-Level Synthesis (HLS) tool.

B. Simulation Tasks Allocation

The main calculation process of the DAT scheme consists of four simulation tasks, as shown in Fig. 8(a):

1. Event scheduling: responsible for converting the gate signals into a switching event sequence through the proposed SEO technique, and selecting the step size based on the switching event sequence.

2. Derivative calculation: the switching matrix K_k and Q_k are selected according to the switching events. Then, the derivatives of the state variables and output variables are calculated. Incremental calculation is then performed for order selection and accuracy evaluation.

3. Diode zero-crossing detection: check whether the diode

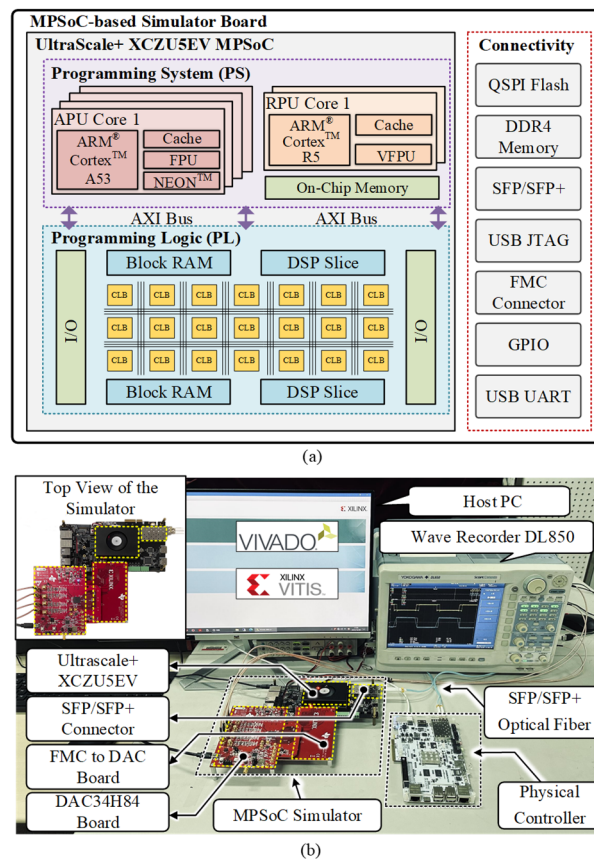


Fig. 7. MPSoC Hardware Platform. (a). Block diagram of Zynq UltraScale+ MPSoC device. (b). The hardware platform of the DAT simulator.

variable crosses zero. If a diode event occurs, a new event is added to the event scheduler and the step size is rearranged.

4. Numerical integration: The numerical solution of the state variables and output variables is calculated based on the incremental calculation, and then the simulation proceeds to the next step according to the event scheduler.

It can be observed that Task 1 and Task 3 mainly involve flexible algebraic computations, while Task 2 and Task 4 primarily focus on data-intensive matrix computations. In a MPSoC, the PS side contains one or more general-purpose processors that excel in executing logic-intensive tasks and complex algorithms. In contrast, the PL side consists of a large number of configurable logic blocks that can implement highly parallel and low-latency data-intensive tasks. Therefore, Task 1 and Task 3 are allocated to the PS side, while Task 2 and Task 4 are allocated to the PL side. This allocation allows the PS and PL to execute tasks that are more suitable for their respective architectures, thereby improving the overall execution efficiency and resource utilization of the simulation scheme.

C. Hardware Implementation

In the PL-side implementation, HLS optimization directives are utilized to control the trade-offs between resource utilization and performance acceleration. Specifically, the UNROLL directive is used to increase loop parallelism and throughput, but requires more resources. The PIPELINE

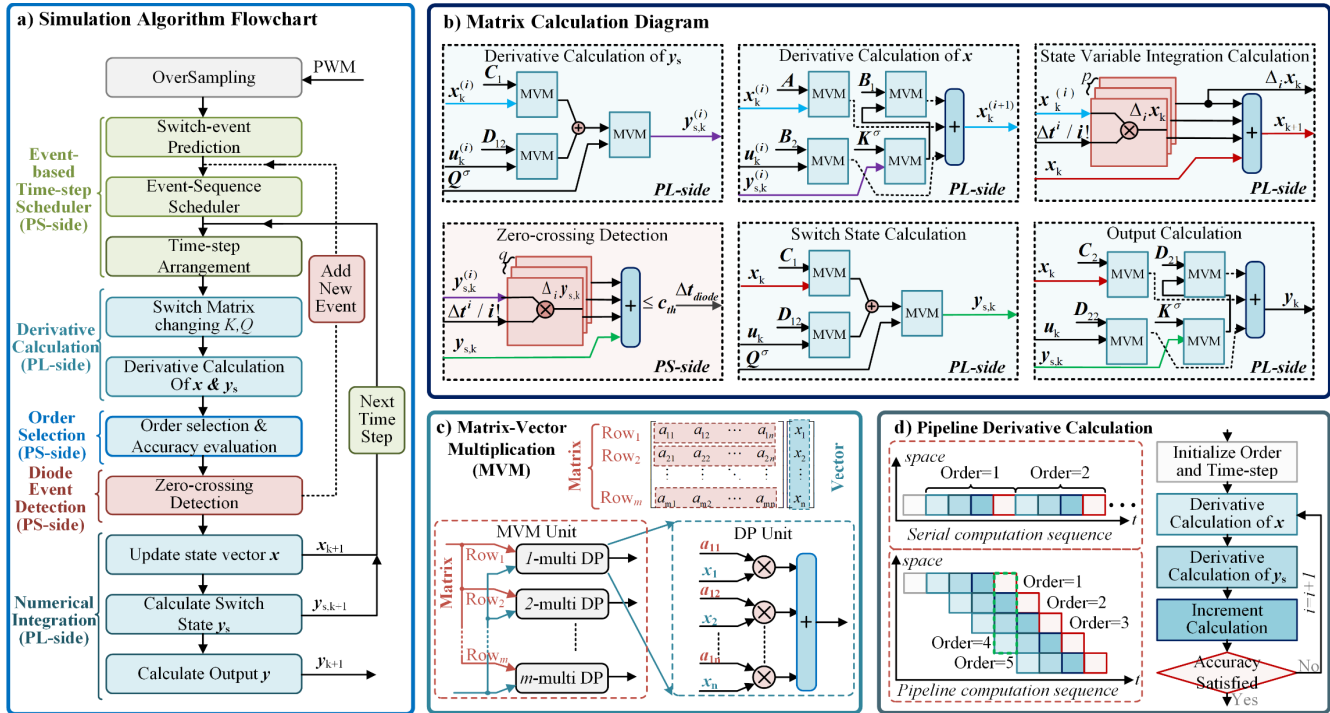


Fig. 8. Assignment and implementation of simulation tasks. (a). The flowchart of the DAT simulation scheme. (b) The matrix calculation diagrams. (c). The matrix product on FPGA. (d). The pipeline computation for the derivative calculation.

directive allows overlapping loop operations, allowing concurrent iterations and improving throughput without significantly increasing resource usage. ARRAY_PARTITION optimizes memory access by partitioning arrays, improving efficiency through concurrent access. If applied appropriately, these directives can assist in optimizing both performance and resource utilization in the implementation. Moreover, UltraRAM (URAM) can be used to achieve a higher memory density than Block RAM (BRAM), enabling larger data storage in the design. However, some specific design constraints, such as access mode, interface width, and depth, need to be considered. After the design is completed, the Timing Slack can be examined to determine if the design can operate normally. If not, optimizations may be required, such as changing the logic design, increasing pin delay, or reducing the clock frequency. Additionally, the AXI4 protocol is chosen as the communication protocol between the PS and PL due to its high data transfer rates and low latency. By minimizing the communication delay between the PS and PL, the impact of hardware latency on the overall performance of the DAT scheme is mitigated. Furthermore, the burst transmission capability of the AXI4 protocol effectively supports the coordination of PS and PL operations within the DAT scheme, particularly when considering diode events. For the physical communication between the proposed simulator and the external controller, the Aurora protocol is primary recommended because it can effectively handle the rapid transmission of large amounts of data while maintaining stable performance. Moreover, Aurora provides flexible configurations and low latency, allowing our design to strike a good balance between performance and complexity.

The simulation tasks are encapsulated into different

functional units in the hardware implementation. These units are shown in Fig. 8(b), along with the data flow direction and dependencies between data. The matrix-vector multiplication (MVM), frequently used in computationally intensive tasks executed in the PL, can be accelerated using the FPGA hardware, as shown in Fig. 8(c). Specifically, the MVM of a matrix ($M \times n$) with a vector ($n \times 1$) can be decomposed into the simultaneous dot products of m rows of the matrix with the vector. The dot product of the $n \times 1$ vector can be implemented in the PL using n multipliers and $n-1$ adders. Therefore, the computation time of an MVM can be optimized to the time required for one multiplication and $n-1$ additions operations.

Besides, the derivative calculation, increment calculation, and accuracy evaluation need to be executed sequentially in the simulation task. Such a serial structure requires a significant amount of computation time. Therefore, this paper adopts multiplexing and pipelining technique to accelerate computation and optimize resources, as shown in Fig. 8(d). Each order of calculation is sequentially calculated, and different orders are computed in a pipelined manner, reducing the time of multi-order derivative calculation significantly. It is worth noting that the proposed scheme prioritizes the Cortex-R5 cores of the MPSoC hardware for the processor core selection on the PS side. The Cortex-R5 cores, with features such as the Triple Timer Counter (TTC), provide superior real-time processing, deterministic performance, and real-time interrupt handling, all of which are critical to the employed simulation method. The Cortex-A53 cores, despite their superior computational power, are deemed less suitable due to their limitations in real-time and deterministic performance. Moreover, offloading most heavy tasks to the FPGA effectively diminishes reliance on the computational

power of the A53 cores.

Furthermore, when using the digital/analog (D/A) converter to send feedback signals to the controller, the variable step-size DAT scheme may face challenges in terms of time resolution. Specifically, the D/A converter needs to output analog signals at a fixed rate, implying that the DAT scheme must adjust the step size of the output results. For controllers with symmetric modulation, the DAT scheme can synchronize with controller timing using the SEO technique. Consequently, the output step size of the D/A converter can be set to the switching period, ensuring proper coordination between the DAT and the controller. To achieve a higher time resolution, interpolation or resampling of the simulation results can be considered before passing them through the D/A converter. Additionally, it is crucial to ensure that the simulation can be completed within the simulation step size.

V. CASE STUDY AND RESULTS ANALYSES

In this section, a four-port modular multi active bridge (MMAB) converter with 32 switches is chosen as the case study to validate the proposed scheme through numerical and HIL experiments.

A. Studied Case: Four-ports MMAB converter

The studied MMAB converter can be used for actively changing power flow to achieve smart grid functionality. However, its coordination control over the four ports poses significant design challenges. Besides, ensuring safety in extreme situations, such as low voltage ride-through and port blocking, further increases the design complexity. Therefore, using an HIL simulator to comprehensively evaluate the MMAB converter controller is a viable solution. However, the operating characteristics of the MMAB converter present significant challenges for HIL simulation. Firstly, the typical switching frequency for an n -port MMAB converter ranges from 20 kHz to 200 kHz, with an increasing trend. Secondly, each switching cycle contains $2 \times n$ switching events, which are hard to locate. Due to these challenges, the MMAB converter was selected as the case study for the proposed DAT simulation scheme in this paper. The converter's topology is shown in Fig. 9, and the relevant parameters are listed in Appendix. The studied converter consists of four modules, each of which contains a A-type H-bridge, a B-type H-bridge and a high-frequency transformer (HFT). The modules are connected through a 50kHz bus.

B. Accuracy Validation with Commercial Simulation

The studied case is applied to both the DAT scheme and a commercial software specialized in power electronics simulation for accuracy validation. The commercial offline simulation software is executed on a personal computer equipped with an intel i7-10700 processor and its results are taken as the reference. The DAT scheme is executed on the MPSoC hardware and the simulation results are stored on the solid-state disk. The control parts of both schemes are kept the same, with the control part for the DAT scheme obtained by converting the control part of the commercial scheme through

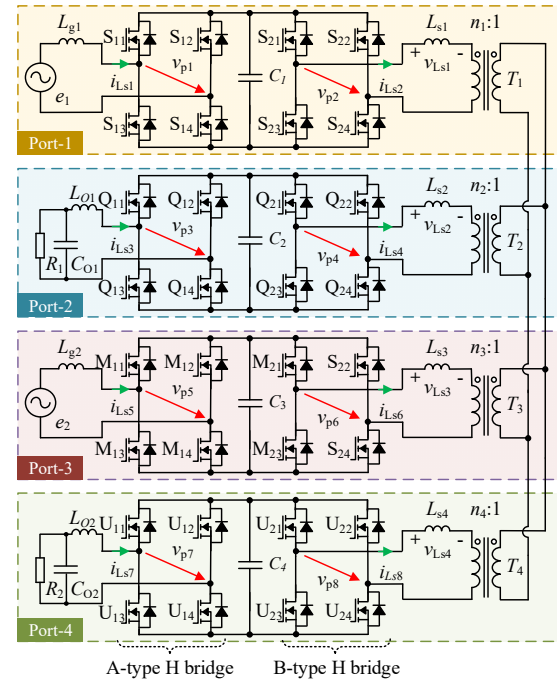


Fig. 9. The topology of studied MMAB converter.

TABLE I
EFFICIENCY COMPARISONS OF DAT AND COMMERCIAL SOFTWARE

	DAT Scheme	Commercial software
Simulation Scenario	0.6s low-voltage ride-through process	
Numerical algorithm	Taylor approximation	Runge-Kutta(4,5)
Step and order	variable-order	variable-step
Calculation time/s	0.6 (1:1)	23.5 (40:1)
Maximum step/s	20e-6	1e-3
Calculation Points	281657	335120
Relative error of i_{AC1}		5.24e-6
Relative error of v_{DC2}		3.47e-7
Relative error of v_{DC3}		4.56e-6
Relative error of i_{HFI}		1.64e-5

Note: The error calculation equation is $Error_{DAT} = \frac{\|y_{DAT} - y_{ref}\|_2}{\|y_{ref}\|_2}$, where $Error_{DAT}$ is the calculated relative error, y_{DAT} denotes the DAT scheme simulation results vector, y_{ref} denotes the reference commercial results vector, and $\|\cdot\|_2$ is the operator of the Euclidean norm.

HDL code toolbox.

The simulation scenario involves a low-voltage ride-through process, where the grid voltage drops to 0.65 p.u. at 0.2s and gradually recovers to the rated value at 0.3s. Port 1 is connected to the grid, and the other three ports are connected to loads. The simulation results are shown in Fig. 10. 10 (a)-(e) display the voltage or current of the four ports, while Fig. 10 (f) illustrates the high-frequency bus currents of Ports 1 and 2. Notably, the high-frequency current in Fig. 10 (f) represents the highest frequency (50 kHz) and most-coupled part of the entire converter, making it the most challenging part for accurate real-time simulation. These highly comparable results indicate that the proposed DAT scheme can attain the same accuracy level of commercial software in terms of the model and solving algorithm. Furthermore, detailed comparisons between the proposed DAT scheme and the commercial software are provided in Table I.

Further, the proposed DAT scheme is compared with the electromagnetic transient program (EMTP) scheme typically

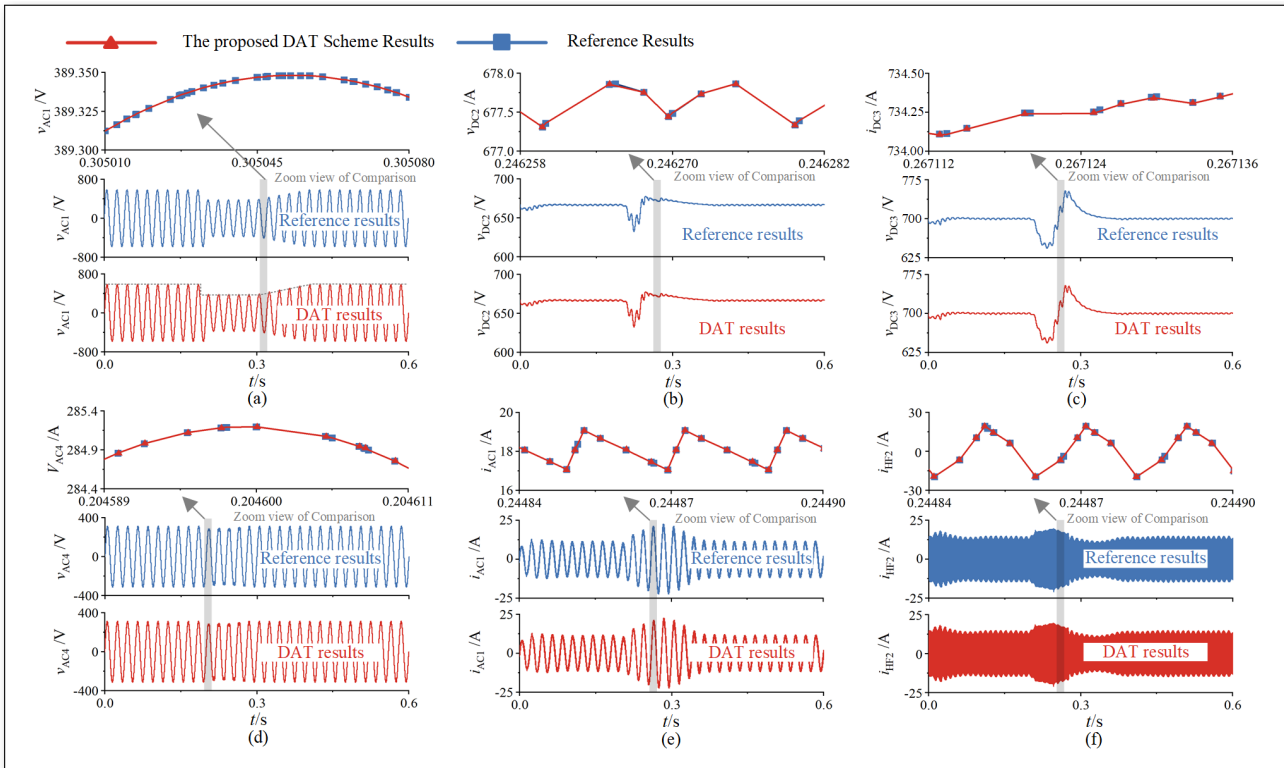


Fig. 10. Simulated results compared with commercial simulation software. The enlarge view is placed on the top of each figure. (a). Grid voltage v_{AC1} of Port 1. (b). Port 2 voltage v_{DC2} . (c). Port 3 voltage v_{DC3} . (d). Grid voltage v_{AC4} of Port 4. (e). The HFT current of Port 1. (f). The HFT current of Port 2.

used in conventional real-time simulators, and the commercial simulation results are used as the reference. To verify the simulation capability for the natural commutation condition, the port blocking process is chosen as the simulation scenario. Specifically, the DC bus voltages of the four ports are maintained at the rated value, and the B-type H-bridges of the four modules are set with phase shift ratios of 0, 0.1, 0.15, and 0.2, respectively. The EMTP scheme includes two options, one with consideration of diode behavior and the other without. Both EMTP schemes are implemented on the PL side of the MPSoC.

Fig. 11 (a) shows the simulation results of the control signals of the ports, the voltage of the high-frequency bus, and the high-frequency current of Ports 1 and 3. It can be observed that the three simulation schemes closely approximate the reference results when the system operates in full-controlled mode before the fault occurs. However, when Port 3 experiences a blocking process due to a fault at 0.01 s, it can only operate in a non-continuous conduction state relying on diodes. At this point, the EMTP scheme without diodes differs significantly from the reference results, while the other schemes continue to perform well.

Fig. 11 (b) illustrates an enlarged view of the blocking process in Fig. 11 (a). Although the EMTP scheme with diodes considers diode behavior, it still cannot accurately determine the timing of diode events. In contrast, the proposed DAT scheme precisely locates diode events within the required accuracy. The step-size adjustment of DAT scheme is based on switch and diode events, resulting only calculates five points during the time frame in Fig. 11.b. In contrast, the

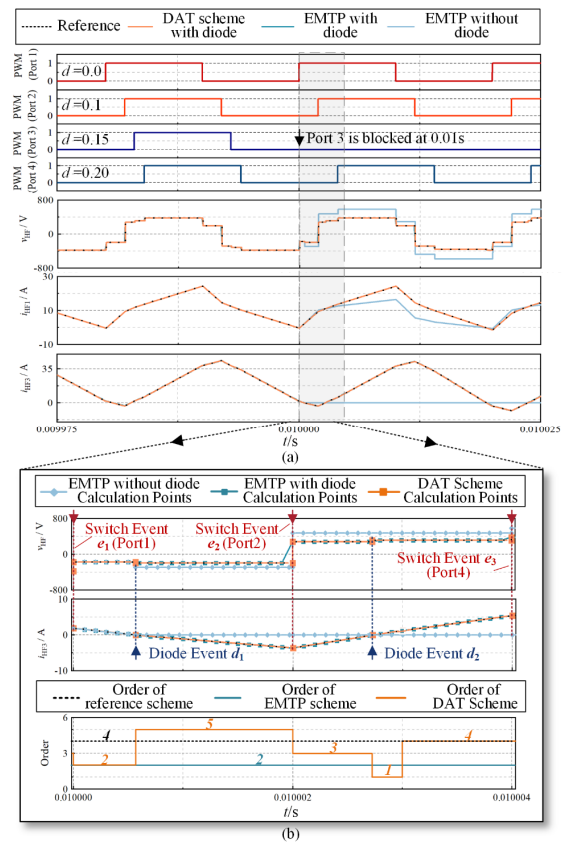


Fig. 11. Comparison with conventional fixed-step HIL simulation methods. (a). Control signals, HFT bus voltage v_{HF} , Port 1 HFT current i_{HF1} , and Port 3 HFT current i_{HF3} . (b). The enlarge view of Fig. (a).

EMTP schemes with a fixed step size of 100 ns generated 40 calculation points. Therefore, the DAT scheme requires fewer calculation points, reducing computational resources and simulating the natural commutation process more accurately.

C. Hardware Resource Utilization Comparison

To evaluate the performance of the proposed simulation task allocation, four computational tasks from the DAT scheme are allocated to the PS and PL sides. The scenario is consistent with Section V-B. Vitis is used for the PS side, while the code for the PL side is synthesized and implemented through Vivado following HLS. The PS-side APU runs at 1.5 GHz, the RPU clock frequency is 600 MHz, and the PL at 100 MHz. The implementation results can be found in Table II.

Compared to exclusive execution on the PL, a reduction of 66.3% in processing time and 79.8% and 67.3% in resource utilization is observed with the collaboration of PS and PL. These metrics demonstrate that the burden on the PL is effectively alleviated by the PS. Moreover, in comparison to computation solely on the PS, the collaborative approach with the PL handling data-intensive tasks results in a 54.4% reduction in PS time slot utilization. This indicates that task allocation and hardware-software co-computation optimize the entire simulation system's performance while lowering single computing resource usage.

The hardware latency of the entire simulation scheme is measured using AXI timers. Table III presents the timing measurements for the relevant data paths. For real-time performance, the RPU on the PS communicates with the PL through dedicated Tightly Coupled Memory (TCM). The total computation task takes 335 ns, while communication between the PS and PL requires 874 ns. The communication delay is usually fixed and difficult to reduce, depending on factors such as the size of the data transferred and the clock frequencies of the PS and PL. The communication delays have little impact on applications with simulation timesteps on the order of tens of microseconds. However, for ultra-high switching frequency applications (>100 kHz), communication delays become a bottleneck issue. In this case, they limit the maximum computation time for simulation tasks, subsequently constraining the simulation scale or accuracy. Furthermore, the detection tolerance and location method for diode events directly affect the communication frequency between the PS and PL, thereby influencing the overall communication delays and achieved acceleration.

This paper recommends that, in ultra-high switching frequency applications, diode detection tolerance should be appropriately reduced and the location method should be simplified. By doing so, communication delays can be decreased and overall acceleration can be improved.

Moreover, the hardware resource utilization comparison of the three schemes is provided in Table III. It can be observed that the DAT scheme utilizes both the ARM processor on the PS-side and the FPGA on the PL-side, while the EMTP schemes only utilize the PL-side resources. Therefore, the DAT scheme requires much less DSP blocks and BRAMs for computation and storage, respectively, compared to the EMTP

TABLE II
PERFORMANCE COMPARISON WITH DIFFERENT TASK ALLOCATION

No.	1	2	3
Task 1	PS	PS	PL
Task 2	PS	PL	PL
Task 3	PS	PS	PL
Task 4	PS	PL	PL
Time slot utilization (PS)	100%	45.59%	0%
Internal Cycle (PL)	0	1390	4130
LUT (PL)	0%	15444(13.2%)	76518(65.4%)
DSP48 (PL)	0%	319 (25.6%)	977(78.5%)
Total time/ μ s	58.7	13.9	41.3

TABLE III
HARDWARE LATENCY OF MPSOC

Start	END	Time
GPIO*	Converted data available in PL	10 ns
PL	Data in TCM of RPU	120 ns
PL	IRQ* interrupt to RPU	240 ns
RPU	Read 256 bit from TCM	44 ns
RPU	Task 1	25 ns
RPU	Data in TCM of RPU	57 ns
Data in TCM of RPU	PL	36 ns
PL	Task 2	128 ns
PL	Data in TCM of RPU	120 ns
PL	IRQ interrupt to RPU	240 ns
RPU	Read 256 bit from TCM	44 ns
RPU	Task 3	68 ns
RPU	Data in TCM of RPU	57 ns
Data in TCM of RPU	PL	36 ns
PL	Task 4	114 ns
PL	GPIO	10 ns

*GPIO and IRQ indicate General Purpose Input/Output and Interrupt Request respectively

TABLE IV
HARDWARE RESOURCE UTILIZATION OF DIFFERENT SCHEME

Resources	DAT	EMTP with diode	EMTP without diode
Time slot utilization	45.59%	0.00%	0.00%
DSP Blocks	319 (25.6%)	771 (61.8%)	713 (57.2%)
LUTs	15444 (13.2%)	44928 (38.4%)	34749 (29.7%)
BRAMs/Kb	1716.0 (6.3%)	13237.9 (48.6%)	10677.4 (39.2%)

TABLE V
HARDWARE RESOURCE UTILIZATION OF DIFFERENT HARDWARE

Resources	UltraScale+ MPSoC	UltraScale+ MPSoC (only PL)	Virtex 7 FPGA
Time slot utilization	45.59%	0%	0.00%
DSP Blocks	319 (25.6%)	977(78.5%)	1494 (53.4%)
LUTs	15444 (13.2%)	76518(65.4%)	79848 (16.4%)
BRAMs/Kb	1716.0 (6.3%)	6755.1 (24.8%)	7092.8 (19.1%)

scheme. In addition, due to the neglect of diode behavior, the resource utilization of the EMTP scheme without diodes is slightly lower than that of the EMTP scheme with diodes.

To further evaluate the advantages and effectiveness of the proposed method in different hardware structures, the proposed method is deployed on MPSoC's PS and PL (referred to as PS/PL scheme), MPSoC's PL (referred to as PL scheme) and FPGA respectively. The FPGA used in this paper is Xilinx's Virtex 7 485t (referred to as V7 scheme). The

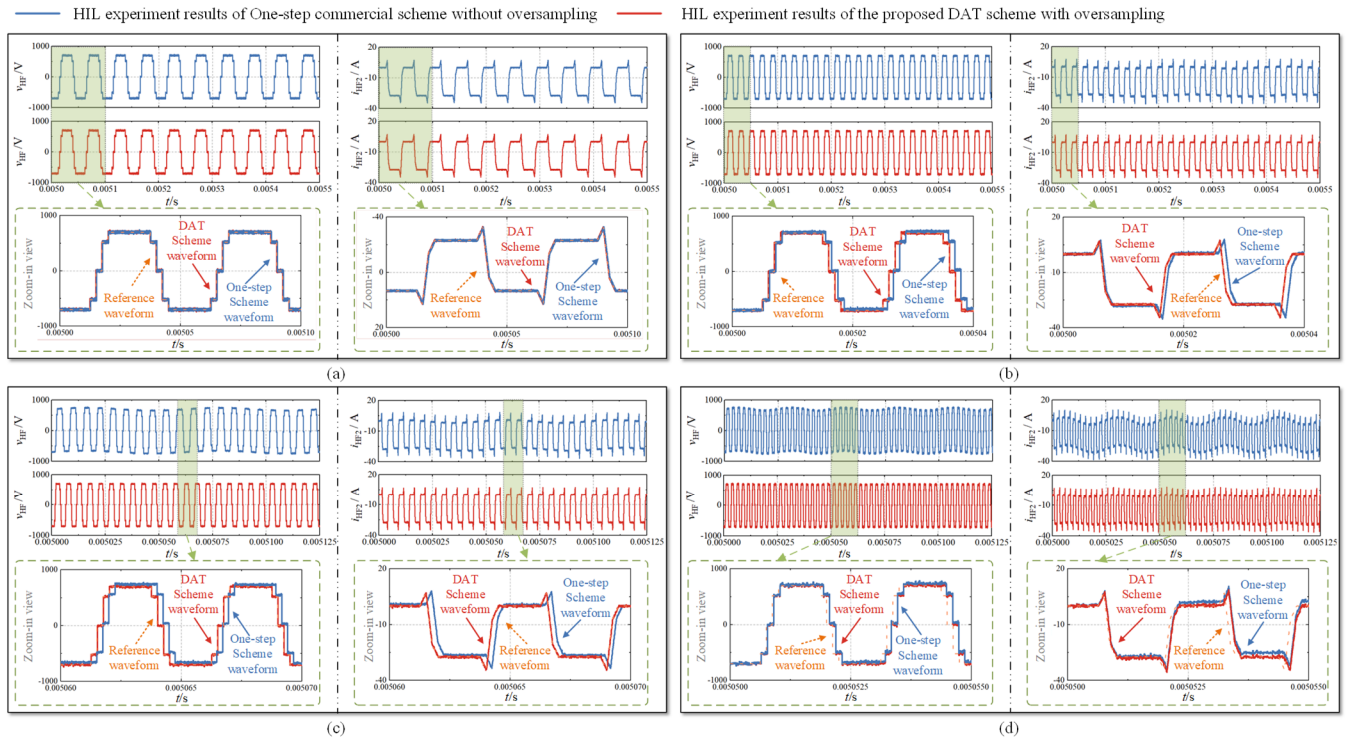


Fig. 12. Comparison with commercial real-time simulators at different switching frequencies. The enlarge view is placed on the bottom of each figure. (a). HFT bus voltage v_{HF} at 20 kHz. (b). HFT bus voltage v_{HF} at 50 kHz. (c). HFT bus voltage v_{HF} at 200 kHz. (d). HFT bus voltage v_{HF} at 400 kHz.

comparison results are shown in Table V. It can be seen that the PS/PL scheme has the lowest hardware utilization because PS offloads some computing to PL. For the PL scheme and V7 scheme, the PL scheme uses 34% fewer DSPs than the V7 scheme. This is due to the different chip performance. However, the hardware utilization of the V7 scheme is less than the PL scheme, because V7 is a complete FPGA chip, and its total hardware resources are naturally greater than MPSoC.

In summary, this section comprehensively evaluates the hardware performance of DAT from the aspects of computational task allocation, communication delay, different simulation schemes, and different hardware structures. From the evaluation indicators, the proposed DAT method is most suitable for running on MPSoC and cooperating on the PS and PL ends to maximize performance and acceleration.

D. Performance Comparison with Commercial HIL Simulator

The proposed DAT scheme and a commercial HIL simulator are both tested in HIL experiments to evaluate their overall performance. The same controller is tested on both real-time simulators, and it operates in open-loop control to avoid any influence on the performance evaluation results. The shift ratios for the four ports of the case study are 0, 0.05, 0.1, and 0.15, while other parameters remained unchanged.

To evaluate the advantages of the proposed SEO technique, comparisons between the proposed scheme, the commercial HIL simulator, and the reference are made at switching frequencies of 20 kHz, 50 kHz, 200 kHz, and 400 kHz. The DAT scheme utilizes the SEO technique with a switching event timing resolution of 5 ns, while the commercial HIL

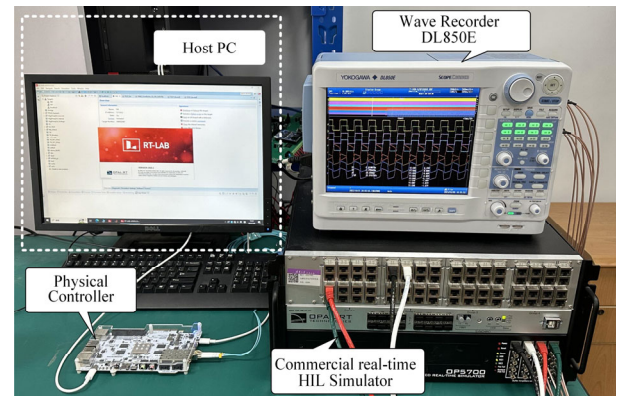


Fig. 13. HIL experiment with commercial HIL simulator.

simulator adopts a traditional one-step approach with a sampling step of 200 ns. Additionally, the DAT scheme synchronizes with the controller timing in symmetric modulation cases. Thus, the simulation step can be set to the switching period, e.g., at 400 kHz, the step is 2.5 μ s, with 1.35 μ s for computation and 1.15 μ s idle. In asymmetric modulation cases, using the smallest possible time step is crucial. In the studied scenario, simulating 2 μ s progress takes up to 1.35 μ s, considering communication delays, resulting in a 2 μ s time step. For smaller scale cases, a 1 μ s minimum time step is recommended, accounting for MPSoC communication latency and functional block delays. While the DAT scheme's time step is larger than commercial simulators, the switching sampling time step is significantly smaller, which is vital for high switching frequency applications.

Furthermore, for different switching frequencies, the tolerances for diode zero-crossing detection are set at 1e-6, 2e-5, 5e-4, and 1e-3s, in order to satisfy the corresponding

computation time constraints. Fig. 12 presents the simulation results of the high-frequency bus voltage v_{HF} and the high-frequency current i_{HF2} of Port 2, and provides an enlarged view of the comparison between different simulators. Table III provides a comprehensive comparison between the two simulations. The DAT simulator used for HIL experiments is shown in Fig. 7.a, while the commercial simulator is illustrated in Fig. 13.

1) At 20 kHz: As shown in Fig. 12(a), the simulation results of the proposed and commercial simulators are both very close to the reference waveform. This indicates that the sampling rates of both schemes are sufficient at 20 kHz.

2) At 50 kHz: The relative errors of v_{HF} and i_{HF2} of the proposed DAT simulator are $2.96e-3$ and $3.87e-3$, respectively, while the relative errors of the commercial HIL simulator are $2.45e-2$ and $3.37e-2$. Fig. 12(b) reveals a slight low-frequency oscillation in the simulation results of the commercial software, which is caused by the insufficient sampling rate and explained in detail in [37].

3) At 200 kHz: The simulation results of the commercial simulator exhibit significant oscillations, as is shown in Fig. 12 (c), and there is still a considerable relative error in Table III. In contrast, the sampling rate of the proposed DAT scheme is the highest frequency that can be achieved by the hardware used. Therefore, even at such a high switching frequency, the proposed DAT scheme can still achieve results that are very close to the reference waveform.

4) At 400 kHz: Fig. 12(d) shows severe distortion in the simulation results of the commercial HIL simulator, rendering it unusable. On the other hand, the proposed DAT scheme also exhibits slight oscillations due to insufficient sampling rate. The relative errors of v_{HF} and i_{HF2} in the commercial simulator are $3.42e-1$ and $5.69e-1$, which are 39.8 times and 62.4 times higher than those of the DAT simulator, respectively.

E. Fidelity validation with Power-level Experiment

To test the fidelity of the proposed scheme, power experiments were also conducted to validate and demonstrate the algorithm's performance in real-world scenarios. A prototype was built according to the topology depicted in Fig. 9, and the experimental platform setup is shown in Fig. 14. In the prototype, each port's Type B H-bridge is connected to power sources and loads with different voltage levels. The corresponding parameters are provided in Table A-II in the Appendix. The prototype employs a single-phase-shift control scheme and classic PI control. Additionally, we integrated the controller into the DAT simulation platform proposed in this paper to assess the fidelity of the simulation scheme.

Fig. 15 presents the high-frequency voltage and current waveforms of the MMAB converter. This is one of the most challenging cases for high-frequency real-time simulation, as these waveforms are highly sensitive to control signals. It can be observed that these waveforms are well-replicated in the DAT scheme. However, to meet real-time requirements, stray parameters, such as stray capacitances of high-frequency busbars, are not considered in the DAT scheme. These parameters are the main cause of the spikes in the experimental waveforms. Overall, the proposed DAT scheme

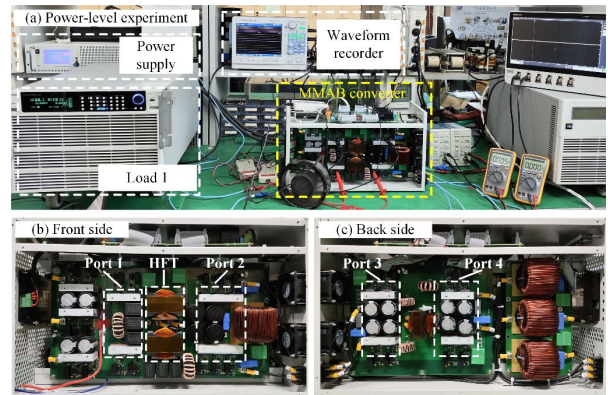


Fig. 14. The power-level experiment platform and laboratory prototype.

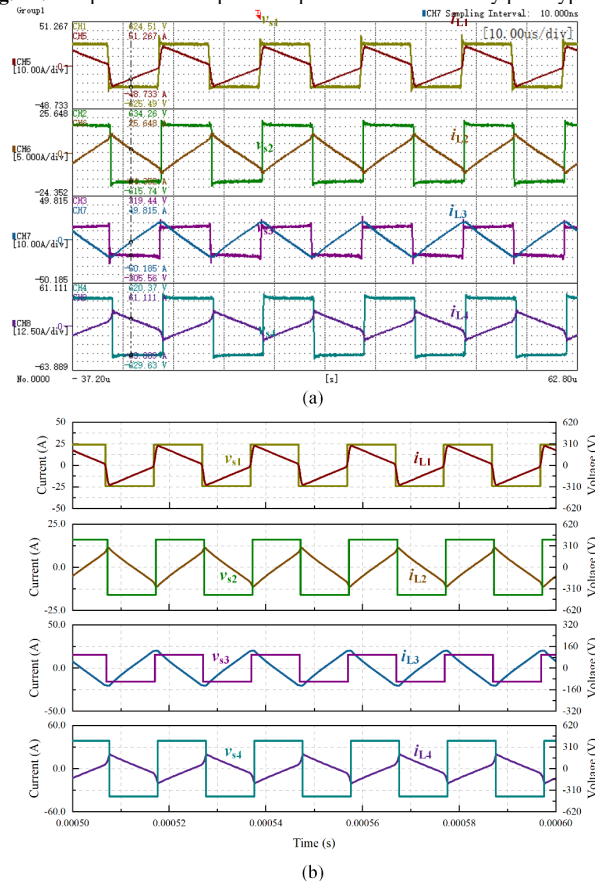


Fig. 15. Fidelity validation with Power-level Experiment. (a). Experiment results. (b). DAT simulation results.

offers high-fidelity simulation results. The experiments above validate the practical effectiveness of the proposed scheme.

F. Comprehensive Evaluation and Future work

In Table VI, a comprehensive comparison is made between the proposed simulator and the commercial HIL simulator. It is evident that the hardware architectures of the two simulators are not identical. The commercial simulator employs a fully FPGA-based architecture, while the proposed DAT simulator utilizes both FPGA and ARM processor for computation. The commercial simulator uses a fixed-step eHS scheme, while the proposed solver employs the variable-step DAT scheme. Through the combination of hardware and solving scheme, the proposed simulator requires only about 1/5 of the computing

TABLE VI
COMPREHENSIVE COMPARISON WITH COMMERCIAL HIL SIMULATOR

Terms	DAT HIL Simulator	Commercial HIL Simulator
Hardware	UltraScale® MPSoC XCZU5EV	Virtex 7 FPGA XC7VX485T
Solver	DAT	eHS
Simulation step	variable	fixed
Sampling step	5 ns (1:1)	200 ns (20:1)
DSP Blocks	319 (1:1)	1421 (4.45:1)
LUTs	15444(1:1)	57428 (3.71:1)
BRAMs/Kb	1716.0 (1:1)	18452.3 (10.75:1)
Relative error of v_{HF} at 20 kHz	2.64e-3 (1:1)	2.81e-3 (1.06:1)
Relative error of i_{HF2} at 20 kHz	3.36e-3 (1:1)	3.75e-3 (1.12:1)
Relative error of v_{HF} at 50 kHz	2.96e-3 (1:1)	2.45e-2 (8.27:1)
Relative error of i_{HF2} at 50 kHz	3.87e-3 (1:1)	3.37e-2(8.71:1)
Relative error of v_{HF} at 200 kHz	7.03e-3 (1:1)	1.28e-1 (18.2:1)
Relative error of i_{HF2} at 200 kHz	7.48e-3(1:1)	1.90e-1 (25.4:1)
Relative error of v_{HF} at 400 kHz	9.83e-3 (1:1)	3.42e-1 (34.8:1)
Relative error of i_{HF2} at 400 kHz	1.34e-2 (1:1)	5.69e-1 (42.4:1)

Note: The error calculation equation is $Error_{rel} = \|y_{sim} - y_{ref}\|_2 / \|y_{ref}\|_2$, where $Error_{rel}$ is the calculated relative error, y_{sim} denotes the simulator results vector, y_{ref} denotes the reference commercial results vector, and $\|\cdot\|_2$ is the operator of the Euclidean norm.

resources and 1/10 of the storage resources of the commercial simulator.

Moreover, the proposed simulator allows a sampling frequency up to 20 times higher than that of the commercial simulator. The use of a variable-order algorithm allows accurate computation of switch and diode events. Therefore, the proposed simulator can significantly improve the simulation accuracy of high-switching frequency applications. In particular, the best-case relative error is only about 1/42 of that of the commercial simulator in the studied case at a switching frequency of 400 kHz.

The excellent performance of the proposed DAT scheme is attributed to the consideration of sampling and computational tasks separately. This allows for constant and extremely small sampling steps, as well as flexible simulation scheme. However, these benefits do not come out of nowhere. The scheme requires the symmetrical modulation for the controller and relevant information to be provided. Furthermore, the proposed scheme currently only utilizes one core of the MPSoC, and does not take advantage of all available processors. In future work, parallel computing techniques among multiple processors and a collaborative solution between multiple processors and FPGA will be considered.

VI. CONCLUSION

This paper proposes an SEO technique and a DAT simulation scheme for real-time HIL simulations. The proposed schemes can accurately locate and calculate switching events and natural commutation processes in high switching frequency applications. Moreover, the MPSoC implementation of the proposed schemes is also presented, including computing tasks allocation, flexible step scheduling, and parallel acceleration of matrix calculation. The case study section presents a comprehensive comparison between the proposed DAT scheme, commercial simulation software,

commercial HIL simulator, and some typical HIL methods with a case study of a MMAB converter. The evaluation results demonstrate significant advantages of the proposed DAT scheme in terms of hardware resource utilization, simulation accuracy, and supported simulation range. The results demonstrate that the proposed scheme can improve the simulation accuracy by 42 times while reducing 9/10 of the computation memory compared with commercial HIL simulator. Overall, the proposed SEO and DAT schemes provide a promising solution for the design and test process of power converters in high switching frequency applications.

APPENDIX

The topology of the full bridge circuit is shown in Fig. 16. Due to space limitations, only one state space equation for the switching state is listed as,

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{2R_{on}}{L_1} & -\frac{1}{L_1} \\ \frac{1}{C_1} & -\frac{1}{RC_1} \end{bmatrix}}_{A_k} \begin{bmatrix} i_L \\ v_C \\ x \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{L_1} \\ 0 \\ B_k \end{bmatrix}}_{B_k} \begin{bmatrix} u_{in} \\ u \end{bmatrix} \quad (21)$$

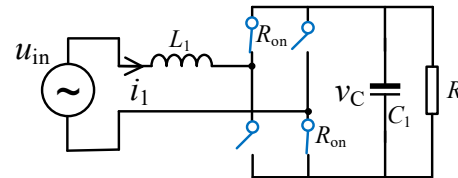


Fig. 16. The topology of a full bridge.

TABLE A-I
SYSTEM PARAMETERS OF STUDIED CASE

Circuit Parameter	Symbol	Value
Port Line Inductance	$L_{r1} \& L_{r2}$	2 μ H
Port Line Inductance	$L_{O1} \& L_{O2}$	15 μ H
DC Bus Capacitor	C_1, C_2, C_3, C_4	1 mF
HFT Leakage Inductance (20kHz)	$L_{S1}, L_{S2}, L_{S3}, L_{S4}$	96 μ H/64 μ H/64 μ H/96 μ H
HFT Leakage Inductance (50kHz)	$L_{S1}, L_{S2}, L_{S3}, L_{S4}$	38.4 μ H/25.6 μ H/25.6 μ H/38.4 μ H
HFT Leakage Inductance (200kHz)	$L_{S1}, L_{S2}, L_{S3}, L_{S4}$	9.6 μ H/6.4 μ H/6.4 μ H/9.6 μ H
HFT Leakage Inductance (400kHz)	$L_{S1}, L_{S2}, L_{S3}, L_{S4}$	4.8 μ H/3.2 μ H/3.2 μ H/4.8 μ H
HFT Turns Ratio	$n_r:1$	1:1
Switching Frequency	f_s	50 kHz
Port 1 Capacitor Rated Voltage	V_1	700
Port 2 Capacitor Rated Voltage	V_2	666.7
Port 3 Capacitor Rated Voltage	V_3	700
Port 4 Capacitor Rated Voltage	V_4	700

TABLE A-II
SYSTEM PARAMETERS OF POWER-LEVEL PROTOTYPE

Parameter	Symbol	Value
Input dc voltage of Port 1	V_1	300 V
Input dc voltage of Port 2	V_2	400 V
Input dc voltage of Port 3	V_3	100 V
Input dc voltage of Port 4	V_4	400 V
DC load in Port 2	R_2	360 Ω
Overall inductance in Port 1	L_{s1}	15 μ H
Overall inductance in Port 2	L_{s2}	20 μ H

>IEEE TRANSACTIONS ON TRANSPORTATION ELECTRIFICATION<

Overall inductance in Port 3	L_{s3}	8 μ H
Overall inductance in Port 4	L_{s4}	50 μ H
Turns ratio of Port 1	n_1	1:1
Turns ratio of Port 2	n_2	1:1
Turns ratio of Port 3	n_3	0.5:1
Turns ratio of Port 4	n_4	1:1
Switching frequency	f_s	50 kHz
Power semiconductors	$X_{21}\text{-}X_{24}(X=S,Q,M,U)$	C3M0021120K

REFERENCES

[1] A. Benigni *et al.*, "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.

[2] 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.

[3] H. Chen *et al.*, "Digital Twin Techniques for Power Electronics-Based Energy Conversion Systems: A Survey of Concepts, Application Scenarios, Future Challenges, and Trends," *IEEE Industrial Electronics Magazine*, pp. 2–18, 2022.

[4] F. Arrano-Vargas *et al.*, "Modular Design and Real-Time Simulators Toward Power System Digital Twins Implementation," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022.

[5] F. Li *et al.*, "Review of Real-time Simulation of Power Electronics," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 4, pp. 796–808, 2020.

[6] V. R. Dinavahi *et al.*, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," *IEEE Transactions on Power Delivery*, vol. 16, no. 4, pp. 775–781, Oct. 2001.

[7] A. Kiffe *et al.*, "Automated generation of a FPGA-based oversampling model of power electronic circuits," in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, Sep. 2012, p. DS3f.5-1-DS3f.5-8.

[8] E. Zamiri *et al.*, "Sub-harmonic oscillations attenuation in hardware-in-the-loop models using the Integration Oversampling Method," *International Journal of Electrical Power & Energy Systems*, vol. 144, p. 108568, Jan. 2023.

[9] H. Chalanger *et al.*, "A Direct Mapped Method for Accurate Modeling and Real-Time Simulation of High Switching Frequency Resonant Converters," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6348–6357, Jul. 2021.

[10] J. Allmeling *et al.*, "Accurate Real-Time Simulation of Converters with Frequent Current Commutation Using Sub-Step Events," in *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2021, pp. 1–6.

[11] Y. Wang *et al.*, "A Review of High Frequency Power Converters and Related Technologies," *IEEE Open Journal of the Industrial Electronics Society*, vol. 1, pp. 247–260, 2020.

[12] H. Chalanger *et al.*, "Methods for the Accurate Real-Time Simulation of High Frequency Power Converters," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021.

[13] M. E. Iranian *et al.*, "Real-Time FPGA-based HIL Emulator of Power Electronics Controllers using NI PXI for DFIG Studies," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pp. 1–1, 2020.

[14] T. Liang *et al.*, "Real-Time Device-Level Simulation of MMC-Based MVDC Traction Power System on MPSoC," *IEEE Transactions on Transportation Electrification*, vol. 4, no. 2, pp. 626–641, Jun. 2018.

[15] A. Boutros *et al.*, "FPGA Architecture: Principles and Progression," *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 4–29, 2021.

[16] C. R. D. Osório *et al.*, "Advancements on Real-Time Simulation for High Switching Frequency Power Electronics Applications (Invited Paper)," in *2021 21st International Symposium on Power Electronics (Ee)*, Oct. 2021, pp. 1–6.

[17] H. F. Blanchette *et al.*, "A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4555–4567, Dec. 2012.

[18] T. Ould-Bachir *et al.*, "CPU/FPGA-Based Real-Time Simulation of a Two-Terminal MMC-HVDC System," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 647–655, Apr. 2017.

[19] A. Hadizadeh *et al.*, "A Matrix-Inversion Technique for FPGA-Based Real-Time EMT Simulation of Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1224–1234, Feb. 2019.

[20] Z. Li *et al.*, "FPGA-based real-time simulation for EV station with multiple high-frequency chargers based on C-EMTP algorithm," *Protection and Control of Modern Power Systems*, vol. 5, no. 1, p. 27, Nov. 2020.

[21] S. Y. R. Hui *et al.*, "Generalised associated discrete circuit model for switching devices," *IEE Proceedings - Science, Measurement and Technology*, vol. 141, no. 1, pp. 57–64, Jan. 1994.

[22] K. Wang *et al.*, "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.

[23] Z. Li *et al.*, "A Discrete Small-Step Synthesis Real-Time Simulation Method for Power Converters," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021.

[24] Q. Mu *et al.*, "Improved ADC Model of Voltage-Source Converters in DC Grids," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 5738–5748, Nov. 2014.

[25] M. Milton *et al.*, "Latency Insertion Method Based Real-Time Simulation of Power Electronic Systems," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 7166–7177, Aug. 2018.

[26] T. Ould-Bachir *et al.*, "A Network Tearing Technique for FPGA-Based Real-Time Simulation of Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3409–3418, Jun. 2015.

[27] J. Zheng *et al.*, "An Event-Driven Parallel Acceleration Real-Time Simulation for Power Electronic Systems Without Simulation Distortion in Circuit Partitioning," *IEEE Transactions on Power Electronics*, vol. 37, no. 12, pp. 15626–15640, Dec. 2022.

[28] C. Liu *et al.*, "Real-Time Simulation of Power Electronic Systems Based on Predictive Behavior," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8044–8053, Sep. 2020.

[29] A. Benigni *et al.*, "Latency-Based Approach to the Simulation of Large Power Electronics Systems," *IEEE Transactions on Power Electronics*, vol. 29, no. 6, pp. 3201–3213, Jun. 2014.

[30] C. Liu *et al.*, "A Network Analysis Modeling Method of the Power Electronic Converter for Hardware-in-the-Loop Application," *IEEE Trans. Transp. Electric.*, vol. 5, no. 3, pp. 650–658, Sep. 2019.

[31] J. Bélanger *et al.*, "Validation of eHS FPGA reconfigurable low-latency electric and power electronic circuit solver," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, pp. 5418–5423.

[32] M. Yushkova *et al.*, "Oversampling Techniques to Improve the Accuracy of Hardware-in-The-Loop Switching Models," *IEEE Transactions on Power Electronics*, pp. 1–12, 2023.

[33] C. T *et al.*, "Time compensated models of switching elements for hardware in loop simulation," in *2016 IEEE Region 10 Conference (TENCON)*, Nov. 2016, pp. 831–836.

[34] J. Allmeling *et al.*, "Sub-cycle average models with integrated diodes for real-time simulation of power converters," in *2017 IEEE Southern Power Electronics Conference (SPEC)*, Dec. 2017, pp. 1–6.

[35] J. Zheng *et al.*, "An Event-Driven Real-Time Simulation for Power Electronics Systems Based on Discrete Hybrid Time-Step Algorithm," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 4809–4819, May 2023.

[36] E. Hairer *et al.*, *Solving Ordinary Differential Equations II*, vol. 14. Berlin, Heidelberg: Springer, 1996.

[37] E. Zamiri *et al.*, "Analysis of the aliasing effect caused in hardware-in-the-loop when reading PWM inputs of power converters," *International Journal of Electrical Power & Energy Systems*, vol. 136, p. 107678, Mar. 2022.



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

>IEEE TRANSACTIONS ON TRANSPORTATION ELECTRIFICATION<

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



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. Now, he is a postdoctoral researcher in Department of Electrical Engineering, Tsinghua University, Beijing. His current research interests include real-time simulation of power electronics.

converters, and artificial-intelligence-based design and control of power converters.



Di Mou (Member, IEEE) was born in Lichuan, Hubei Province, China, in 1994. He received the B.S. degree from the Three Gorge University, Yichang, China, in 2017, and the Ph.D. degree from the Chongqing University, Chongqing, China, in 2021, both in electrical engineering. He is currently a Postdoctoral Fellow with the Tsinghua University, Beijing, China. His research interests include multiport power electronic transformers.



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.



Weicheng Liu (Student member, IEEE) received the B. E. degree in electrical engineering from Southeast University, China, in 2018, and M.S degree in electrical engineering from Southeast University in 2021. He is currently working towards the Ph.D. degree in electrical engineering from Tsinghua University, China. His current research interests include power system simulation methodology and technique.



Han Xu (Student member, IEEE) received the B.S. degree in electrical engineering in 2021 from Tsinghua University, Beijing, China. Since 2021, he has been working toward the master degree in electrical engineering at the Department of Electrical Engineering, Tsinghua University, Beijing, China. His research interests include simulation of power electronic systems.



Haoyu Wang received the B.S. degree in electrical engineering and automation from Shanghai Jiao Tong University, Shanghai, China, in 2020, and the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2023. He is currently working toward the Ph.D. degree in electrical engineering at Columbia University in New York, NY. His current research interests include dc-dc converters, high-efficiency power