Received: 29 November 2021 | Revised: 31 March 2022 | Accepted: 25 April 2022

DOI: 10.1049/rpg2.12500

IET Renewable Power Generation

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Asynchronous multi-rate method of real-time simulation for active distribution networks

Hao Fu[1] | Peng Li[1] | Xiaopeng Fu[1] | Zhiying Wang[1] | Jinyue Yan[2] | Jianzhong Wu[3] | Chengshan Wang[1]

[1]Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin, China

[2]School of Business, Society, and Engineering, Mälardalen University, Västerås, Sweden

[3]Institute of Energy, School of Engineering, Cardiff University, Cardiff, UK

Correspondence
Xiaopeng Fu, Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China
Email: fuxiaopeng@tju.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 51961135101, U1866207, 51807131

Abstract

The real-time simulation of active distribution networks (ADNs) can provide an accurate insight into transient behaviours, but faces challenges in simulation efficiency and flexibility brought by larger system scales and wider time-scale ranges. This paper presents an asynchronous multi-rate (AMR) method and design for the real-time simulation of large-scale ADNs. In the proposed method, the entire ADN was decoupled into different subsystems according to accuracy requirements, and optimized time-steps were allocated to each subsystem to realize a fully distributed simulation. This not only alleviated the time-step coordination problem existing in multi-rate real-time simulations, but also enhanced the flexible expansion capabilities of the real-time simulator. To realize the AMR real-time simulation, a multi-rate interfacing method, synchronization mechanism, and data communication strategy are proposed in this paper, and their hardware design is also presented in detail. A modified IEEE 123-node system with photovoltaics and wind turbine generators was simulated on a 3 field-programmable gate arrays (FPGAs)-based AMR real-time simulator. The real-time results were captured by the oscilloscope and verified with PSCAD/EMTDC, which demonstrated the superiority in simulation flexibility and accuracy compared with the synchronous multi-rate (SMR) method.

1 | INTRODUCTION

With the increasing penetration of distributed generators (DGs) and energy storage systems, power distribution systems have evolved from passive networks to active distribution networks (ADNs) [1, 2]. Although this transition has paved the way for more sustainable power grids, it poses challenges to the secure and stable operation of power systems [3]. Many concerns associated with the DGs, such as strong uncertainties [4], power oscillations [5], and overvoltage issues [6], have emerged, thus making it necessary to devise effective simulation methods and tools to enhance the understanding of these transients. Real-time electromagnetic transient (EMT) simulations can accurately reproduce the complex dynamic behaviours of ADNs at the pace of real-world clock time [7]. This advantage enables real-time simulators with the ability of hardware-in-the-loop validations [8], which can perform closed-loop tests on the external controllers and protective devices [9–11]. However, the integration of a large number of DGs enlarges the simulation scale of the ADNs, leading to an excessive computational burden [12]. To reproduce these transients with a high fidelity, a simulation time-step at the microsecond level, generally determined by the fastest dynamics, is required [13]. The conflict between the computational burden and real-time requirements poses a significant challenge to the real-time simulation of ADNs.

Multi-rate (MR) simulation, modelling decoupled subnetworks with different integration time-steps, has been considered as an efficient method for the real-time simulation of large-scale systems [14–20]. The synchronous multi-rate (SMR) algorithms, which support an extendable simulation of interconnected subsystems with integer multiples of time-steps, are presently the most established MR algorithms embedded in real-time simulators. An SMR algorithm combining superstep and substep strategies was proposed to enhance modelling capabilities in the powerful real-time simulator, RTDS [17]. Another SMR algorithm based on the Stubline decoupling method was presented for the real-time simulation of two-terminal MMC-HVDC grids, where the entire system was split into three subsystems and processed on one CPU and two field-programmable gate arrays (FPGAs) with different time-steps [18]. Based on the transmission line model, an SMR simulation architecture for ADNs was proposed, and the real-time simulation of a modified IEEE 33-node system with photovoltaic (PV) and battery units was realized on a 4-FPGA-based real-time simulator [19]. Although the above research has realized the SMR real-time simulation of different systems, it cannot be ignored that certain prerequisites must be met for applying these algorithms to the real-time simulation. Namely, two balances must be achieved between the hardware resources and simulation requirements as follows: one is the coordination between the calculation resources and simulation accuracy according to the requirements of integer multiples of time-steps; and the other is the coordination between the communication resources and the number of decoupled subsystems for the extendable simulation. However, balanced coordination is sometimes difficult to achieve owing to critical real-time requirements. This is because, for real-time simulators, the calculation resources and solution speed are two conflicting constraints. With limited calculation resources, it is difficult to determine an appropriate time-step for each decoupled subsystem to simultaneously achieve both high simulation accuracy and time-step coordination with other subsystems, especially for the SMR real-time simulation with large numbers of subsystems. This problem may be alleviated by further decomposing the subnetworks and allocating abundant hardware resources [18, 19]; however, this would increase the computational cost and communication burden [21].

To address the above issues, a growing interest has arisen in the asynchronous multi-rate (AMR) concept [22], which allows a distributed MR simulation with arbitrary time-step relationships. The AMR offline simulation has been implemented in [22], in which an asynchronous Multi-Area Thevenin Equivalent-multirate (A-MATE-multirate) algorithm is proposed for solving subsystems with different time-steps. Nevertheless, the current research on AMR simulation is still in the preliminary stage, and few explorations have been carried out from a real-time simulation perspective. The communication delay among subsystems and the demand of fast calculation in the real-time simulation both put forward higher requirements on the AMR algorithm. Therefore, further research is required for developing an AMR simulation framework and corresponding algorithms suitable for real-time simulations. In this paper, we have proposed an AMR real-time simulation framework for the first time. Its salient features include the following.

1) *Versatility and high fidelity*: Difficulty in time-step coordination among subsystems was avoided, enabling more versatile MR real-time simulations. In addition, time-steps that best matched the time-scale of the subsystems could be used in the AMR simulation so that the internal transients could be reproduced with a high fidelity.
2) *Flexible plug-and-play (PnP)*: As a fully distributed simulation, the AMR method allowed the subsystems to self-update without the need for simultaneous solutions. This high autonomy enabled the real-time simulator to flexibly expand its simulation scale through plug-and-play of the underlying hardware.

To enhance the AMR real-time simulation, this paper presents a novel AMR algorithm in which a general interpolation formula was integrated for the fast calculation of interface variables. Moreover, the real-time constraint of the propagation delay was considered to ensure real-time interaction. With the proposed algorithm, an extendable AMR real-time simulation platform was developed based on multiple FPGAs. The major contributions of this study are summarized as follows:

1) Based on the Bergeron line model, this paper proposes an AMR algorithm considering real-time constraints for the flexible real-time simulation of ADNs. The proposed algorithm can perform MR simulations with arbitrary time-step relationships, allowing subsystems to use the best simulation time-step for the optimum performance on accuracy and hardware resource utilization.
2) An AMR real-time simulation architecture supporting PnP modelling is presented. The AMR interfacing method, synchronization mechanism, and data communication scheme have been integrated into the multi-FPGA-based AMR simulation platform for facilitating the asynchronous operation.

The remainder of this paper is organized as follows. Section 2 introduces the proposed AMR algorithm and the real-time simulation framework. Section 3 describes a detailed hardware design of the FPGA-based AMR real-time simulation platform. Numerical case study is presented in Section 4, demonstrating the effectiveness of the proposed method and design, and finally conclusions are made in Section 5.

2 | ASYNCHRONOUS MULTI-RATE SIMULATION METHOD

2.1 | Introduction to AMR simulation

Based on the numerical solution relationship among subsystems, the MR simulation algorithms are classified as SMR [14] and AMR [22] simulations. Figure 1 shows these two types of MR simulations of a sample network, where the entire network is decoupled into three subsystems, and the most suitable time-step sizes for their simulation scales are represented by $\Delta \hat{t}_m$, $m = 1, 2, 3$, as shown in Figure 1a. The time-steps of Subsystem m in the SMR and AMR simulations are denoted by $\Delta t'_m$ and Δt_m. Here, $\{\Delta t'_1/\Delta t'_2,\ \Delta t'_2/\Delta t'_3\} \subseteq \mathbb{Z}^+$, and $\{\Delta t_1/\Delta t_2, \Delta t_2/\Delta t_3\} \subseteq \mathbb{R}^+$.

The obvious difference between two algorithms is the freedom in the time-step configuration of MR subsystems. For the AMR simulation in Figure 1c, a high degree of freedom allows the subsystems to select the optimized time-step according to

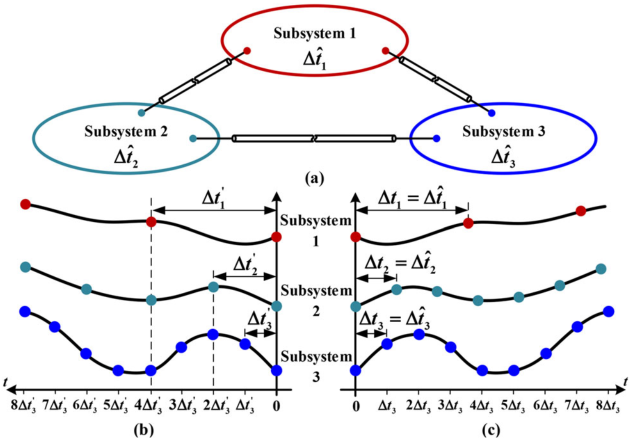

Subsystem 1
$\Delta\hat{t}_1$
Subsystem 2
$\Delta\hat{t}_2$
Subsystem 3
$\Delta\hat{t}_3$
(a)
$\Delta t_1'$
$\Delta t_2'$
$\Delta t_3'$
Subsystem 1
Subsystem 2
Subsystem 3
$\Delta t_1=\Delta\hat{t}_1$
$\Delta t_2=\Delta\hat{t}_2$
$\Delta t_3=\Delta\hat{t}_3$
t
$8\Delta t_3'$ $7\Delta t_3'$ $6\Delta t_3'$ $5\Delta t_3'$ $4\Delta t_3'$ $3\Delta t_3'$ $2\Delta t_3'$ $\Delta t_3'$ 0
0 Δt_3 $2\Delta t_3$ $3\Delta t_3$ $4\Delta t_3$ $5\Delta t_3$ $6\Delta t_3$ $7\Delta t_3$ $8\Delta t_3$
(b)
(c)

FIGURE 1 Comparison of the two types of MR simulations. (a) Sample network. (b) SMR simulation. (c) AMR simulation

the requirements of the simulation accuracy and efficiency, that is, $\Delta t_m = \Delta\hat{t}_m$ for Subsystem m. However, this freedom cannot be guaranteed for the SMR simulation because of the requirement of integer multiples of time-steps, which mainly arises from the simultaneous and partial simultaneous solutions of the simulated system [14]. As shown in Figure 1b, the SMR algorithms perform a simultaneous solution for the entire network at the time of main synchronization, that is, the time corresponding to the integer multiples of $\Delta t_1'$, and a partial simultaneous solution for the interlinked subsystems 2 and 3 at the time of sub-synchronization [23], that is, the time corresponding to the integer multiples of $\Delta t_2'$.

Compared with the SMR simulation, the AMR simulation is an implementation of fully distributed self-updating of multiple subsystems. The solution of each subsystem is carried out separately from other subsystems, and the entire network is not solved simultaneously [22]. Although these subsystems are also updated simultaneously at the common multiple of their time-steps, their updating processes are independent, and these coinciding points are not different from the other solution points in the AMR simulation.

2.2 | Network decomposition

Based on the nodal analysis method, the network equation of a typical ADN for time-discretized EMT simulations can be expressed as follows:

$$\boldsymbol{G}_{\mathrm{ADN}}\boldsymbol{v}_{\mathrm{ADN}}(t) = \boldsymbol{i}_{\mathrm{ADN}}(t) - \hat{\boldsymbol{i}}_{\mathrm{ADN}}(t-\Delta t_{\mathrm{ADN}}) \tag{1}$$

With various decoupling methods, the entire ADN can be decoupled into different subsystems for MR simulations. In this study, the Bergeron line model (BLM), which is known to be stable due to its passivity [24], was employed for a preliminary exploration of AMR simulations. The equivalent circuit of this interface model is shown in Figure 2; τ_{mn}, Z_{mn}, and R_{mn} are the propagation delay, surge impedance, and lumped resistance, respectively, of the BLMs between Subsystems m and n. Since the length of the interface lines usually does not match with the propagation delay, it is necessary to insert split nodes into the long lines or adjust the capacitance of the short lines for a matching length. Thus, the matching length can be calculated using the following equation.

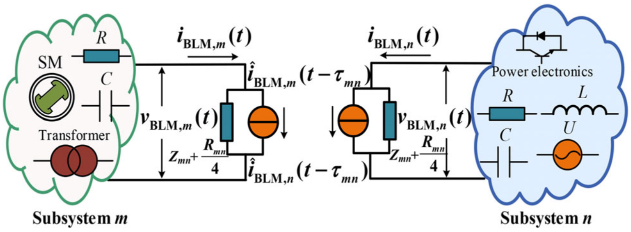

SM
R
C
Transformer
$\boldsymbol{i}_{\mathrm{BLM},m}(t)$
$\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{mn})$
$\boldsymbol{v}_{\mathrm{BLM},m}(t)$
$Z_{mn}+\frac{R_{mn}}{4}$
$\hat{\boldsymbol{i}}_{\mathrm{BLM},n}(t-\tau_{mn})$
$\boldsymbol{i}_{\mathrm{BLM},n}(t)$
$\boldsymbol{v}_{\mathrm{BLM},n}(t)$
$Z_{mn}+\frac{R_{mn}}{4}$
Power electronics
R
L
C
U
Subsystem m
Subsystem n

FIGURE 2 Decoupling of the ADN using BLM

$$d_{mn} = \tau_{mn}/\sqrt{L_{mn}'C_{mn}'} \tag{2}$$

Assuming that the entire ADN was decoupled into M subsystems, different integration time-steps could be used to solve these subsystems according to their inherent timescales. Thereafter, the network equations for the decoupled subsystems were derived from Equation (1) and are given by

$$\begin{cases} \boldsymbol{G}_1\boldsymbol{v}_1(t) = \boldsymbol{i}_1(t) - \hat{\boldsymbol{i}}_1(t-\Delta t_1) - \hat{\boldsymbol{i}}_{\mathrm{BLM},1}(t-\boldsymbol{\tau}_1) \\ \vdots \\ \boldsymbol{G}_m\boldsymbol{v}_m(t) = \boldsymbol{i}_m(t) - \hat{\boldsymbol{i}}_m(t-\Delta t_m) - \hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\boldsymbol{\tau}_m) \\ \vdots \\ \boldsymbol{G}_M\boldsymbol{v}_M(t) = \boldsymbol{i}_M(t) - \hat{\boldsymbol{i}}_M(t-\Delta t_M) - \hat{\boldsymbol{i}}_{\mathrm{BLM},M}(t-\boldsymbol{\tau}_M) \end{cases} \tag{3}$$

where the current vector $\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\boldsymbol{\tau}_m) = [\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{m1})\cdots\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{mn})\cdots\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{mM})]^T$, $n \neq m$, represents the history current source of all BLMs inside the subsystem m; $\hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{mn})$ is the history current source vector of the BLMs between subsystems m and n. It should be noted that the dimensions of $\boldsymbol{i}_m$, $\hat{\boldsymbol{i}}_m$, and $\hat{\boldsymbol{i}}_{\mathrm{BLM},m}$ were expanded so that they were consistent with that of $\boldsymbol{G}_m$, where $m \leq M$ and $m \in \mathbb{Z}^+$. The history current sources of the BLM between subsystems m and n are given by the following equations.

$$\begin{cases} \hat{\boldsymbol{i}}_{\mathrm{BLM},m}(t-\tau_{mn}) = \boldsymbol{i}_m'(t-\tau_{mn}) + \boldsymbol{i}_m''(t-\tau_{mn}) \\ \hat{\boldsymbol{i}}_{\mathrm{BLM},n}(t-\tau_{nm}) = \boldsymbol{i}_n'(t-\tau_{nm}) + \boldsymbol{i}_n''(t-\tau_{nm}) \end{cases} \tag{4}$$

$$\begin{cases} \boldsymbol{i}_m'(t-\tau_{mn}) = \boldsymbol{D}_t^1\boldsymbol{v}_{\mathrm{BLM},m}(t-\tau_{mn}) + \boldsymbol{D}_t^3\boldsymbol{i}_{\mathrm{BLM},m}(t-\tau_{mn}) \\ \boldsymbol{i}_m''(t-\tau_{mn}) = \boldsymbol{D}_t^2\boldsymbol{v}_{\mathrm{BLM},n}(t-\tau_{mn}) + \boldsymbol{D}_t^4\boldsymbol{i}_{\mathrm{BLM},n}(t-\tau_{mn}) \\ \boldsymbol{i}_n'(t-\tau_{nm}) = \boldsymbol{D}_t^1\boldsymbol{v}_{\mathrm{BLM},n}(t-\tau_{nm}) + \boldsymbol{D}_t^3\boldsymbol{i}_{\mathrm{BLM},n}(t-\tau_{nm}) \\ \boldsymbol{i}_n''(t-\tau_{nm}) = \boldsymbol{D}_t^2\boldsymbol{v}_{\mathrm{BLM},m}(t-\tau_{nm}) + \boldsymbol{D}_t^4\boldsymbol{i}_{\mathrm{BLM},m}(t-\tau_{nm}) \end{cases} \tag{5}$$

Here, both τ_{mn} and τ_{nm} denote the propagation delay of the BLM between subsystems m and n, thus, $\tau_{mn} = \tau_{nm}$; $\boldsymbol{i}'_m(t - \tau_{mn})$ and $\boldsymbol{i}'_n(t - \tau_{nm})$ denote the variables without interaction of the BLMs inside subsystems m and n, respectively; $\boldsymbol{i}''_m(t - \tau_{mn})$ represents the interface variables transferred from the subsystem n to m, and vice versa is represented by the vector $\boldsymbol{i}''_n(t - \tau_{nm})$; $\boldsymbol{D}_t^1$, $\boldsymbol{D}_t^2$, $\boldsymbol{D}_t^3$, and $\boldsymbol{D}_t^4$ are the coefficient vectors, whose expressions can be found in [19] and are not included due to space limitations.

From Equations (4) and (5), the interface variables and variables without interaction of all BLMs inside the subsystem m can be expressed by the vectors $\boldsymbol{i}''_m(t - \tau_m) = [\boldsymbol{i}''_m(t - \tau_{m1}) \cdots \boldsymbol{i}''_m(t - \tau_{mn}) \cdots \boldsymbol{i}''_m(t - \tau_{mM})]^T$ and $\boldsymbol{i}'_m(t - \tau_m) = [\boldsymbol{i}'_m(t - \tau_{m1}) \cdots \boldsymbol{i}'_m(t - \tau_{mn}) \cdots \boldsymbol{i}'_m(t - \tau_{mM})]^T$, respectively. On this basis, Equation (3) can be rearranged as

$$\begin{cases} \boldsymbol{G}_1 \boldsymbol{v}_1(t) = \boldsymbol{i}_1(t) - \hat{\boldsymbol{i}}_1(t - \Delta t_1) - \boldsymbol{i}'_1(t - \boldsymbol{\tau}_1) - \boldsymbol{i}''_1(t - \boldsymbol{\tau}_1) \\ \vdots \\ \boldsymbol{G}_m \boldsymbol{v}_m(t) = \boldsymbol{i}_m(t) - \hat{\boldsymbol{i}}_m(t - \Delta t_m) - \boldsymbol{i}'_m(t - \boldsymbol{\tau}_m) - \boldsymbol{i}''_m(t - \boldsymbol{\tau}_m) \\ \vdots \\ \boldsymbol{G}_M \boldsymbol{v}_M(t) = \boldsymbol{i}_M(t) - \hat{\boldsymbol{i}}_M(t - \Delta t_M) - \boldsymbol{i}'_M(t - \boldsymbol{\tau}_M) - \boldsymbol{i}''_M(t - \boldsymbol{\tau}_M) \end{cases} \quad (6)$$

As can be seen from Equation (6), when solving the subsystem m, the vector $\boldsymbol{i}'_m$ is updated with the time-step of Δt_m, whereas the vector $\boldsymbol{i}''_m$ is updated with different time-steps. Therefore, the solution of each subsystem requires the interface variables from the neighbouring subsystems updated with different time-steps, which calls for an effective interfacing method for the multi-rate simulation.

2.3 | AMR interfacing method

The interface variables with time-delays need to be accurately estimated to represent the effects of the neighbouring subsystems. In the SMR simulation, to extract the interface variables, the propagation delay τ_{mn} was set as the time-step of the slow subsystem Δt_m artificially, where $\Delta t_m / \Delta t_n \in \mathbb{Z}^+$ [19]. And the interface variables of the subsystems were averaged or interpolated to obtain their time-delay values. However, since Δt_m and Δt_n in the AMR simulation do not satisfy the integer multiple relationship, the average of the fast interface variables may not accurately represent their contribution to the slow subsystem. Therefore, linear interpolation was employed for both fast and slow interface variables in the AMR simulation; it simultaneously acted as a low-pass filter and removed the imaging and aliasing errors introduced by the sampling rate conversions [22]. The interpolated interface variables with time-delays were calculated using Equation (7).

$$\begin{cases} \boldsymbol{i}''_m(t - \tau_{mn}) = \boldsymbol{i}''_m\left(t - \hat{t}'\right) + \dfrac{\boldsymbol{i}''_m\left(t - \hat{t}''\right) - \boldsymbol{i}''_m\left(t - \hat{t}'\right)}{\Delta t_m}\left(\hat{t}' - \tau_{mn}\right) \\ \boldsymbol{i}''_n(t - \tau_{nm}) = \boldsymbol{i}''_n\left(t - t'\right) + \dfrac{\boldsymbol{i}''_n(t - t'') - \boldsymbol{i}''_n(t - t')}{\Delta t_n}\left(t' - \tau_{nm}\right) \end{cases} \quad (7)$$

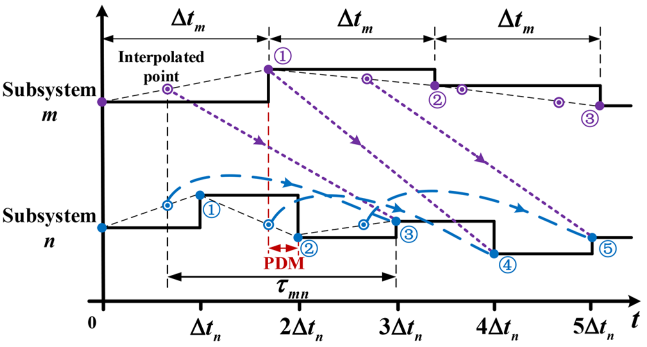


FIGURE 3 Schematic of the AMR interfacing method

Here, $t' = \lceil \frac{\tau_{nm}}{\Delta t_n} \rceil \cdot \Delta t_n$, $t'' = \lfloor \frac{\tau_{nm}}{\Delta t_n} \rfloor \cdot \Delta t_n$, $\hat{t}' = \lceil \frac{\tau_{mn}}{\Delta t_m} \rceil \cdot \Delta t_m$, and $\hat{t}'' = \lfloor \frac{\tau_{mn}}{\Delta t_m} \rfloor \cdot \Delta t_m$; $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ represent the rounding up and rounding down operations, respectively.

When applying the proposed interfacing method for AMR real-time simulations, the propagation delay is an important factor affecting real-time performance to consider. Due to the real-time requirement, the interpolated interface variables should be obtained before using them to solve the neighbouring subsystems, as shown in Figure 3. Note that the interpolation can only be performed after the interface variables at $\lceil \frac{t - \tau_{mn}}{\Delta t_m} \rceil \cdot \Delta t_m$ and $\lfloor \frac{t - \tau_{mn}}{\Delta t_m} \rfloor \cdot \Delta t_m$ have been solved. This implies that the interval from the interpolation to the use of the interpolated values, that is, the propagation delay margin (PDM), should be greater than 0 for both the slow and fast subsystems. This constraint on the propagation delay can be expressed as

$$\begin{cases} (k_n - 1)\, \Delta t_n > \left\lceil \dfrac{k_n \cdot \Delta t_n - \tau_{mn}}{\Delta t_m} \right\rceil \Delta t_m, \forall k_n = 2, 3, \cdots, J \\ (k_m - 1)\, \Delta t_m > \left\lceil \dfrac{k_m \cdot \Delta t_m - \tau_{mn}}{\Delta t_n} \right\rceil \Delta t_n, \forall k_m = 2, 3, \cdots, H \end{cases} \quad (8)$$

where J and H are the numbers of Δt_n and Δt_m within the least common multiple of Δt_n and Δt_m , respectively. A minimum propagation delay satisfying the inequality constraints would be required in the proposed algorithm to achieve high accuracy.

2.4 | Framework of the AMR simulation

Based on the proposed AMR algorithm, a framework of the AMR simulation is presented in this section. The real-time simulation of an ADN partitioned into two subsystems—Subsystem m for the slow network and Subsystem n for the fast network—was employed to illustrate the proposed framework, as shown in Figure 4. The time-steps of the subsystems m and n are denoted as Δt_m and Δt_n, where $\Delta t_m / \Delta t_n \in \mathbb{R}^+$. The calculation of each subsystem would begin with the variable initialization, and then the subsystems m and n would perform their EMT simulations simultaneously under the synchronization mechanism, which was designed to eliminate the natural frequency deviation of crystal oscillators on different emulator hardware. Within a

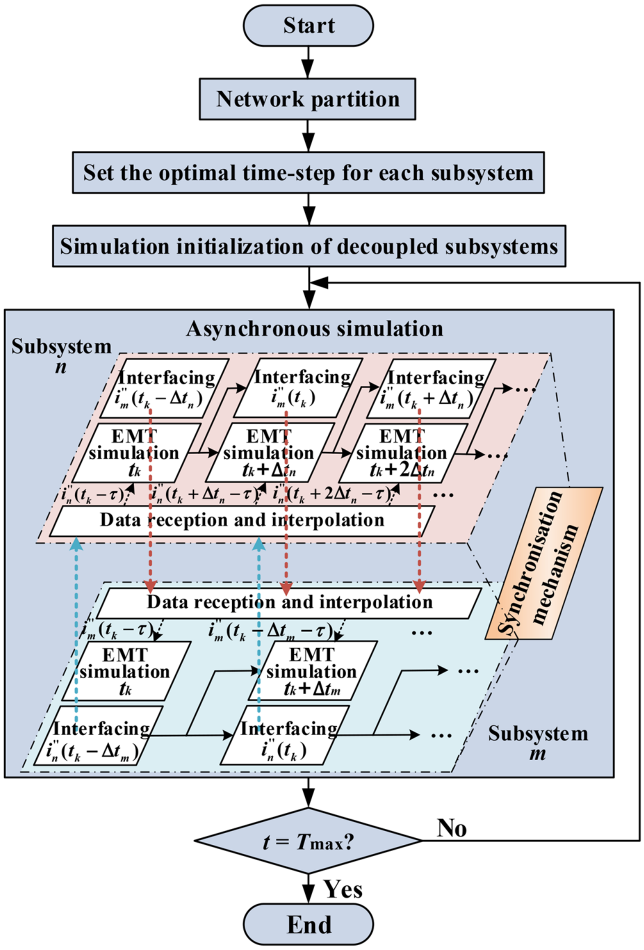

Start
Network partition
Set the optimal time-step for each subsystem
Simulation initialization of decoupled subsystems
Asynchronous simulation
Subsystem n
Interfacing $i''_m(t_k-\Delta t_n)$
Interfacing $i''_m(t_k)$
Interfacing $i''_m(t_k+\Delta t_n)$
EMT simulation t_k
EMT simulation $t_k+\Delta t_n$
EMT simulation $t_k+2\Delta t_n$
$i''_n(t_k-\tau)$
$i''_n(t_k+\Delta t_n-\tau)$
$i''_n(t_k+2\Delta t_n-\tau)$
Data reception and interpolation
Synchronisation mechanism
Data reception and interpolation
$i''_m(t_k-\tau)$
$i''_m(t_k+\Delta t_m-\tau)$
EMT simulation t_k
EMT simulation $t_k+\Delta t_m$
Interfacing $i''_n(t_k-\Delta t_m)$
Interfacing $i''_n(t_k)$
Subsystem m
$t = T$max?
No
Yes
End

FIGURE 4 Framework of the AMR real-time simulation

time-step, the history current sources $\hat{\boldsymbol{i}}_m(\hat{\boldsymbol{i}}_n)$, interpolated interface variables $\boldsymbol{i}''_m(\boldsymbol{i}''_n)$, and BLM variables without interaction $\boldsymbol{i}'_m(\boldsymbol{i}'_n)$ were required to be read from the memories for updating Equation (6). Thereafter, each subsystem would solve the transmission line current $\boldsymbol{i}_{\mathrm{BLM},m}(\boldsymbol{i}_{\mathrm{BLM},n})$ and terminal voltage $\boldsymbol{v}_{\mathrm{BLM},m}(\boldsymbol{v}_{\mathrm{BLM},n})$ based on the updated nodal voltage $\boldsymbol{v}_m(\boldsymbol{v}_n)$. The parallel communication mechanism, that is, implementing the simulation and communication in parallel, was applied in the AMR simulation to avoid the additional time consumption associated with data storage. Under this mechanism, each subsystem would send out the interface variable $\boldsymbol{i}''_m(\boldsymbol{i}''_n)$ after completing a time-step solution, and the neighbouring subsystem would perform the EMT simulation while its data reception and prediction module received the interface variables and calculated the interpolation points. Such decoupled iterations were to be continued in parallel until the termination time $T_{\max}$ was reached; then, the real-time simulator would output the simulation results.

The data reception and interpolation module shown in Figure 4 was designed to receive the interface variables, calculate the interpolation points, and update the interpolation information. Taking the subsystem m as an example, the interpolation information would include the time-step number p_{mn} and the interpolation coefficient c_m. These two variables were formulated as Equations (9) and (10), where mod{·} represents a modulo operation.

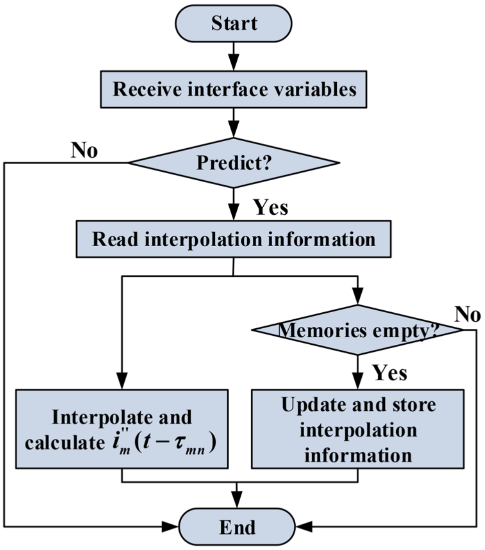

Start
Receive interface variables
Predict?
No
Yes
Read interpolation information
Memories empty?
No
Yes
Interpolate and calculate $i''_m(t-\tau_{mn})$
Update and store interpolation information
End

FIGURE 5 Flow chart of the data reception and interpolation module

$$p_{mn} = \left\lceil \frac{k_m \cdot \Delta t_m - \tau_{mn}}{\Delta t_n} \right\rceil \tag{9}$$

$$c_m = \mathrm{mod}\left(\frac{k_m \cdot \Delta t_m - \tau_{mn}}{\Delta t_n} \right) / \Delta t_n \tag{10}$$

To reduce time consumption, a synchronous update strategy was integrated into the data reception and interpolation module, as shown in Figure 5. In this strategy, the received interface variables would be accessed to calculate the interpolation points with the interpolation information if a prediction were needed, and then the approximated data would be stored for the following EMT simulation. Simultaneously, this strategy would check whether the interpolation information needs to be updated, and this update was to be implemented using Equations (9) and (10), if necessary.

3 | COMPREHENSIVE AMR SIMULATION IMPLEMENTATION

In this section, a multi-FPGA-based AMR real-time simulation platform is presented, and its overall architecture and function modules are introduced. In addition, the hardware implementations of the AMR interface, synchronization mechanism, and data communication are also presented.

3.1 | AMR real-time simulation platform

Considering the openness and flexibility, we developed an AMR real-time simulation platform, as depicted in Figure 6,

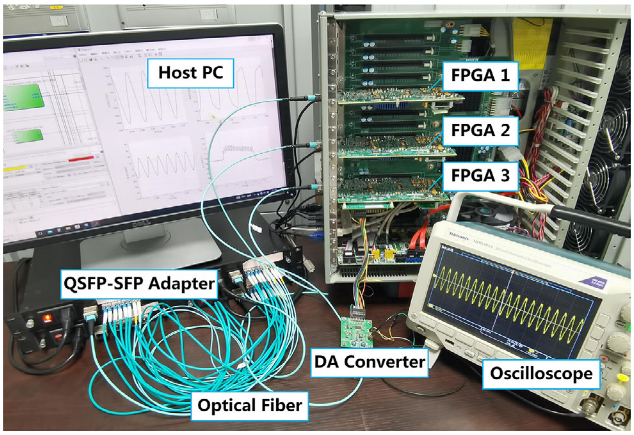


FIGURE 6 Hardware platform of the FPGA-based AMR real-time simulator

instead of using commercial simulators. This platform mainly consisted of a host PC, three FPGA modules, a QSFP-SFP adapter, and other external devices. Here, the Intel Stratix® V 5SGSMD5K2F40C2N FPGA was used, and each FPGA development board was equipped with four 2.5 Gbps full-duplex optical fiber communication channels that enabled flexible connection among the FPGAs. The simulation of the test cases in the case studies was performed on this platform.

The functional structure of the hardware platform is shown in Figure 7, which included the application and operation programs. The application program on the host PC was designed to accomplish topology identification, system partitioning, parameter generation, and result display [25]. The operation program, supported by three FPGAs, was designed to implement the AMR real-time simulation. Multiple function modules have been developed in the operation program, including i) a global control module for resetting, initializing, and starting simulations, ii) an electrical system module for simulating various electrical components, iii) a control system module for modelling the DGs and their controllers, and iv) an I/O adapter for ensuring bidirectional transmission of interactive data among the FPGAs. It should be noted that each FPGA was equipped with an independent operation program to simulate one subsystem with its time-step.

3.2 | Hardware design of AMR interface

In the developed real-time simulation platform, the AMR interface was regarded as an electrical component and was simulated in the electrical system module, as shown in Figure 7. Here, the subsystem m in the ADN shown in Figure 2 was employed as an example to illustrate the hardware design of the AMR interface.

In the AMR interface module, the branch current of the BLM $\boldsymbol{i}_{\mathrm{BLM},m}$ was first calculated based on the updated terminal voltage $\boldsymbol{v}_{\mathrm{BLM},m}$, after which the interface variable i_n'' and the variable without interaction i_m' could be solved using Equation (5). It should be noted that the simulation of the interface elements was implemented in the form of a pipeline. Therefore, a time-delay link was designed for the $\boldsymbol{v}_{\mathrm{BLM},m}$ in the solution process to ensure clock alignment, and all stored coefficient vectors, including $\boldsymbol{D}_t^1$, $\boldsymbol{D}_t^2$, $\boldsymbol{D}_t^3$, and $\boldsymbol{D}_t^4$, were continuously read from the ROM unit in the form of a pipeline. Once the interface variable i_n'' was solved, it would be immediately sent to the subsystem n via the I/O adapter. However, since the variable $i_m'(t-\Delta t_m)$ corresponded to the integer multiples of Δt_m, they would be temporarily stored instead of being used directly. Simultaneously, the values of i_m' at $t-\hat{t}'$ and $t-\hat{t}''$ were read from the RAM unit and sent to the interpolation unit for calculating $i_m'(t-\tau_{mn})$. Similarly, the received interface variable $i_m''(t-\Delta t_n)$ sent by the subsystem n corresponded to the integer multiples of Δt_n; therefore, these data were also cached in the dedicated RAM unit. Since the interpolation points of i_m'' were not distributed in each time-step of the subsystem n, the interface variables $i_m''(t-\hat{t}')$ and $i_m''(t-\hat{t}'')$ corresponding to the solution time-step of the subsystem m were accessed with the interpolation information p_{mn} and sent to the data reception and prediction submodule for calculating $i_m''(t-\tau_{mn})$ using Equation (7). The summation value of $i_m'(t-\tau_{mn})$ and $i_m''(t-\tau_{mn})$ was then sent to the linear equation solver for the calculation of nodal voltages. Once the terminal voltages of the BLMs were obtained, they would be sent back to the AMR interface module for the next time-step.

3.3 | Synchronization mechanism

Although the subsystems were solved asynchronously with different time-steps, the simulation clock of each subsystem in the AMR simulation still needed to be synchronized. This is because these simulation clocks were generated by separate oscillators on different FPGAs, and the intrinsic clock skew was hardly eliminated [26]. Accumulated clock deviations may cause the interface variables to be unavailable before being required by the neighbouring subsystems, thus reducing the simulation accuracy or even yielding incorrect results.

In this study, a synchronization mechanism was proposed for the AMR simulation, as depicted in Figure 8. In the multi-FPGA system, any one of the FPGAs could be selected as the master FPGA. Here, the FPGA m was chosen as the master, and was designed to provide the synchronization signal *sync_sig* with the interval $\Delta t_{\min}$ for other FPGAs. The value of $\Delta t_{\min}$ is configurable and is typically set to the greatest common divisor of the time-steps of all subsystems. It should be noted that the signal *sta_sync_sig* in Figure 8 is a special synchronization signal used to start a time-step of subsystems. Each subsystem was to start its entire simulation process only when it received the first *sta_sync_sig* sent by the master. Given the transmission delay of the synchronization signals, the simulation start time for the master FPGA must be delayed by the same amount of time.

For the AMR simulation of a large number of subsystems, the limited communication channels may make it impossible for the master FPGA to connect directly with all the other FPGAs. In this case, the FPGAs connected to the master FPGA should provide synchronization signals for the remaining FPGAs. As

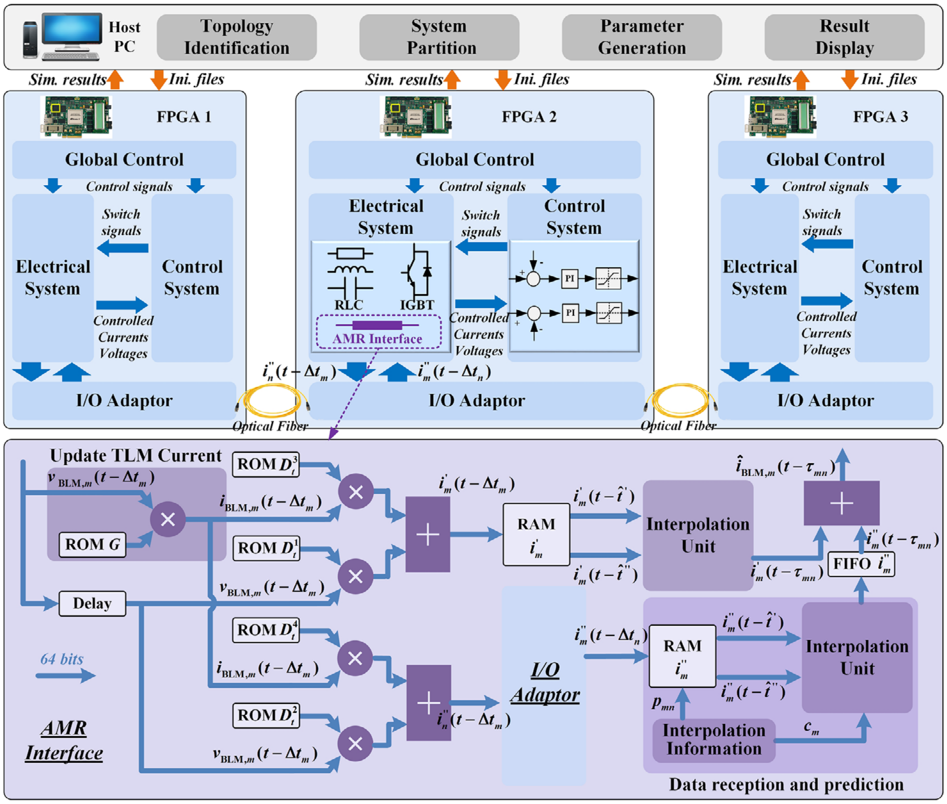


FIGURE 7 Functional structure of the hardware platform

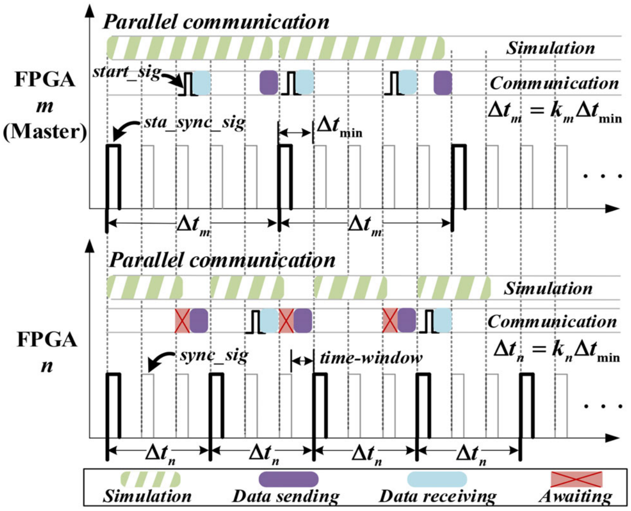


FIGURE 8 Synchronization mechanism and data communication strategy

a result, the simulation start times of the master and bridge FPGAs must be delayed accordingly to ensure the simultaneous simulation start of all subsystems.

3.4 | Data communication strategy

To enhance the real-time performance of the AMR real-time simulations, a data communication strategy was designed to achieve high-speed and high-fidelity communication among FPGAs. The proposed data communication strategy included parallel communication and data interaction mechanisms.

In the parallel communication mechanism, each subsystem implements the simulation and communication in parallel to avoid the additional time cost associated with data storage, as depicted in Figure 8. Under this mechanism, the subsystem would send out the interface variables once completing a time-step solution, and the neighbouring subsystem would keep performing the EMT simulation while its data reception and prediction module received the interface variables, predicted and stored the interpolation points.

A data communication mechanism was developed to realize transceiver channel multiplexing and accurate data transmission and reception. As mentioned in Section 3.3, a synchronization signal was generated and transmitted between the master and the remaining FPGAs for clock benchmarking. To conserve transceiver channel resources, this signal shared the same transceiver channel with the interface variables. A time-window-based time-sharing multiplexing technology was designed to avoid confusion between the synchronization signals and interface variables, as shown in Figure 8. Specifically, the sending moment of the interface variables should be limited within the time-window, which is defined as the period between two consecutive synchronization signals. If the solved interface variables could complete the data-sending process within the time-window, they would be sent out immediately after being calculated. Otherwise, they had to wait for the next time-window.

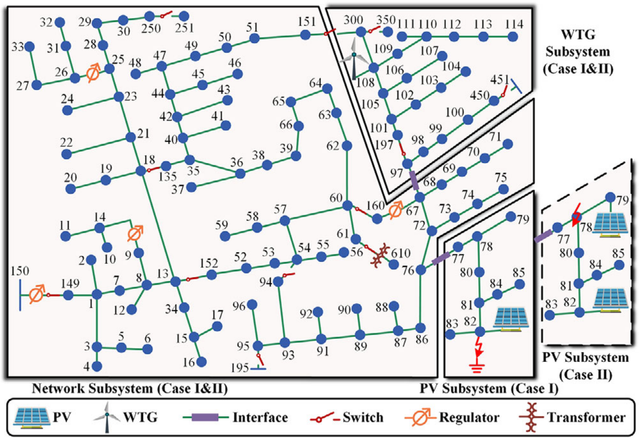


FIGURE 9 Partitioning of the modified IEEE 123-node system

In addition to sending out the interface variables, the receiving process played a key part in the data interaction. In contrast to the SMR simulation, here the receiving moment of the interface variables was not relatively fixed within the time-step of the receiving subsystem, and this would make it impossible to determine a time within the time-step for receiving the data. To accurately receive the interface variables, a dedicated message was added to the header of the interface data packet. After receiving the dedicated message, the subsystem would generate a signal to start receiving the interface data.

4 | NUMERICAL TEST AND VALIDATION

4.1 | Test case

In this section, a large-scale ADN modified from the IEEE 123-node system was employed for the case studies. As illustrated in Figure 9, the test system has two configurations, which correspond to different stages of the project development, one with a PV unit and a permanent magnet synchronous generator (PMSG)-based wind energy conversion system (WECS), and another with two PV units and a PMSG-based WECS. Detailed parameters of the IEEE 123-node network can be found in [27], and the structure and parameters of the PV unit and PMSG-based WECS are provided in [19] and [28], respectively. We considered different transient events in the two test cases to evaluate the performance of the proposed AMR method. A Phase-A-ground fault at the grid connection point of the PV unit and a Phase A-B fault at the upstream feeder of PV units were considered in Cases I and II, respectively, which both occurred at 3.0 s and were cleared after 0.3 s. Another simulation scenario of the wind velocity decreasing from 12 m/s to 10 m/s at 4.2 s was imposed to the PMSG in both two cases. To study the transients of the grid-connected PVs under different faults, the power electronic components in the PV units were represented by the associated discrete circuit (ADC) model [29] for accurately reflecting the internal switching processes, while the back-to-back converter in the PMSG-based WECS was represented by the switching-function model [30].

Considering the modelling capabilities of a single FPGA and the different transient research needs, the entire system was decomposed into three subsystems: the network, PV, and wind turbine generator (WTG) subsystems. The PV and WTG subsystems were decoupled from the network subsystem at Lines 76–77 and Lines 67–97, respectively. The real-time simulation of the test system was implemented on the simulation platform described in Section 3.1, and each subsystem was simulated on a single FPGA.

4.2 | Multi-rate simulation

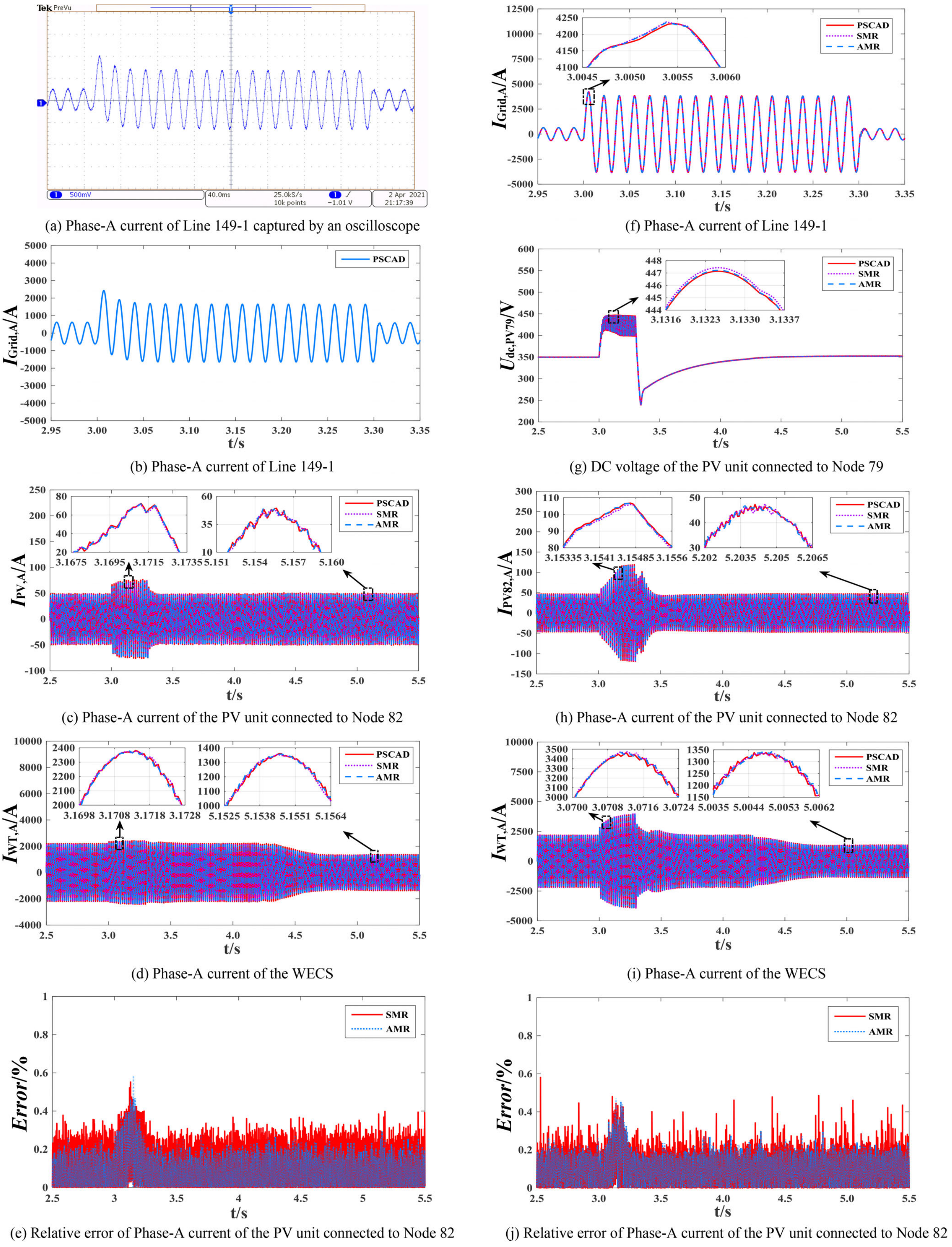
To achieve an accurate real-time simulation of the test system, the minimum time-steps of the network, WTG, and PV subsystems in the two test cases were set to 22, 7, and 3 μs, respectively. This was the limit for the Stratix$^{®}$ V Edition DSP development board, as any means to speed up the simulation of each subsystem would require more hardware resources than on-board resources. At the same time, the SMR real-time simulations of the two test cases were also implemented based on the designed interface to demonstrate the universality of the proposed method, and the simulation time-steps of the three subsystems were set to 24, 8, and 4 μs, respectively.

The minimum propagation delay satisfying the constraint (8) was applied to the AMR simulations to enhance their stability. The propagation delay between the network and WTG subsystems was set to 29 μs and that between the network and PV subsystems was set to 26 μs. For the SMR simulation, the propagation delay was set to the step size value of the network subsystem, which was 24 μs.

4.3 | Accuracy validation

To validate the proposed method and hardware design, the AMR and SMR simulation results of the two test cases were captured and compared with PSCAD/EMTDC. It is worth mentioning that the test system was simulated in this commercial simulation tool with a single time-step of 3 μs.

Figure 10 shows the dynamics of the two test cases, where the individual feeder current, currents of the PMSG-based WECS and PV units, prior, during and immediately after the transient events were displayed. The Phase-A current waveforms of Line 149-1 in Cases I and II are shown in Figures 10a and 10f, respectively. It is worth noting that Figure 10a was captured by a oscilloscope, and its corresponding behaviour simulated in the PSCAD/EMTDC was shown in Figure 10b. From the above results, we can see that the feeder current increased to three to five times the rated value during the faults, which is of great value for the formulation of protection schemes. The Phase-A output current and DC voltage waveforms of PV units under the phase-to-phase and single-phase ground fault are displayed in Figures 10c, g, and h. It can be seen that the currents of the PV units were distorted during the faults, and recovered after



(a) Phase-A current of Line 149-1 captured by an oscilloscope

(f) Phase-A current of Line 149-1

(b) Phase-A current of Line 149-1

(g) DC voltage of the PV unit connected to Node 79

(c) Phase-A current of the PV unit connected to Node 82

(h) Phase-A current of the PV unit connected to Node 82

(d) Phase-A current of the WECS

(i) Phase-A current of the WECS

(e) Relative error of Phase-A current of the PV unit connected to Node 82

(j) Relative error of Phase-A current of the PV unit connected to Node 82

FIGURE 10 MR simulation results of the test system: (a)–(e) Case I; (f)–(j) Case II

TABLE 1 Resource utilization for the 3-FPGA-based hardware design

Hardware design	ALMs (172,600)	DSP blocks (1590)	RAM blocks (39 Mbits)
Network subsystem (Cases I and II)	95%	24%	82%
WTG subsystem (Cases I and II)	98%	17%	33%
PV subsystem (Case I)	92%	14%	15%
PV subsystem (Case II)	93%	32%	47%

the clearance. As the wind velocity decreased, the current of the WECSs in the two cases both dropped from the 2.20-kA peak to the 1.33-kA peak, as depicted in Figures 10d and 10i. Figures 10e and 10j show the relative error of the Phase-A current of the PV unit at Node 82 with the PSCAD/EMTDC as a reference.

As can be seen from Figure 10, the simulation results of the AMR method closely coincided with the reference waveforms and the relative error of the AMR simulation was smaller than that of the SMR simulation. This verified the accuracy of the proposed method and the correctness of the hardware design. In contrast to PSCAD/EMTDC, the errors of the FPGA-based simulator incorporate the following aspects: (i) The single-precision floating-point data format was applied in the modelling of the PV and PMSG to decrease the hardware resource. (ii) Interpolation prediction was applied in the AMR interfacing method, which may bring errors. (iii) This paper adopted the ADC and SF model to represent power electronic circuits, while the binary resistance model was used in PSCAD/EMTDC for modelling switches.

4.4 | Resource utilization

The hardware resource utilization of the three subsystems was presented in Table 1. The FPGA resources mainly included adaptive logic modules (ALMs), DSP blocks, and RAM blocks. The ALMs, consisting of lookup tables (LUTs) and registers, were used to implement a variety of different combinatorial and sequential logical functions, and they were the most heavily consumed in the real-time simulations based on floating-point numbers.

As shown in Table 1, each subsystem occupied almost all of the ALMs on the FPGA, which indicated that it was not feasible to shorten the time-step using a higher degree of computational parallelism. On the other hand, the computational resources utilized by the WTG and PV subsystems were 95–97% of the network subsystem, even though their simulation scale was 13–37% of the network subsystem. This is because the modelling of the PVs and PMSGs, as well as their controllers, was relatively complicated, and their simulation required more hardware resources than that of conventional power grids. At the same time, the logical resource utilization of the PV subsystems in the two cases was almost the same, although the PV subsystem in Case II integrated two PV units. This is because a spatial-temporal parallel architecture is applied in the PV subsystem to improve the simulation capability.

5 | CONCLUSION

In this paper, a novel AMR method is presented to improve the simulation efficiency and flexibility of large-scale ADNs. It partitions the entire ADN into several subsystems according to the accuracy requirements, and provides a fully distributed simulation of subsystems with any arbitrary time-steps. To realize the AMR real-time simulation, the multi-rate interfacing method considering the real-time requirements, as well as the synchronization mechanism and data communication strategy, is proposed. And their hardware design has been further developed and embedded in the extendable real-time simulation platform to enhance the plug-and-play capabilities.

To illustrate the proposed method and design, two modified IEEE 123-node systems considering different transient scenarios have been simulated on the platform. The test cases are decoupled into three subsystems, and the optimized time-steps are allocated to them, thus alleviating the time-step coordination problem. The real-time simulation results of both the BLM-based AMR and SMR method are captured and compared with PSCAD/EMTDC. It is found that the relative error of the AMR simulation is smaller than that of the SMR simulation, and both them are less than 0.6% when the time-step of slow subsystems is less than 8 times that of fast subsystems. This verifies the accuracy of the proposed method as well as the correctness of hardware design. In the future, the stability of AMR simulation methods is an area of further investigation.

NOMENCLATURE

Variables

$\boldsymbol{i}_{\mathrm{ADN}}, \boldsymbol{i}_m$	current source vectors of the branches inside the active distribution network (ADN) and inside Subsystem m
$\hat{\mathbf{i}}_{\mathrm{ADN}}, \hat{\mathbf{i}}_m$	history current source vectors of the branches inside the ADN and inside Subsystem m
$\hat{\mathbf{i}}_{\mathrm{BLM},m}, \boldsymbol{i}_{\mathrm{BLM},m}$	history current source vector and branch current vector of the Bergeron line model (BLM) inside Subsystem m
k_{fs}	time-step ratio of the slow and fast subsystem in the SMR simulation
k_m	time-step number of Subsystem m
p_{mn}	time-step number of Subsystem n for the interpolation of interface variables required for the simulation of Subsystem m
$\boldsymbol{v}_{\mathrm{ADN}}, \boldsymbol{v}_m$	nodal voltage vectors of the electrical subsystem of ADN and of Subsystem m
$\boldsymbol{v}_{\mathrm{BLM},m}$	terminal voltage vector of the BLM inside Subsystem m
$x_{\mathrm{f}}, x_{\mathrm{f,avg}}$	fast-varying variable and its average in the SMR simulation

Parameters

$\boldsymbol{\tau}_m$ propagation delay vector of all transmission lines between Subsystem m and its adjacent subsystems

τ_{mn} propagation delay of transmission lines between Subsystems m and n

Δt_{ADN} time-step size of ADN

Δt_m, Δt_n time-step sizes of Subsystems m and n in the AMR simulation

$\Delta t'_m$ time-step size of Subsystem m in the SMR simulation

$\Delta \hat{t}_m$ the most suitable time-step size of Subsystem m for its simulation scale

c_m interpolation coefficient of interface variables required for the simulation of Subsystem m

C'_{mn} capacitance per unit length of the transmission line between Subsystems m and n

d_{mn} length of the transmission line between Subsystems m and n

$\boldsymbol{G}_{\mathrm{ADN}}, \boldsymbol{G}_m$ admittance matrices of the electrical subsystem of the ADN and of Subsystem m

K_{fs} time-step number of the fast subsystems within that of the slow subsystems

K_m, K_n ratios of the time-step sizes of Subsystems m and n to the interval of synchronization signals

L'_{mn} inductance per unit length of the transmission line between Subsystems m and n

M number of subsystems decoupled from the ADN

CONFLICT OF INTEREST

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled: "Asynchronous Multi-rate Method of Real-time Simulation for Active Distribution Networks".

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Xiaopeng Fu https://orcid.org/0000-0001-6056-7103
Zhiying Wang https://orcid.org/0000-0001-6932-6998

REFERENCES

1. Liu, W., Gu, W., Li, P., et al: Non-iterative semi-implicit integration method for active distribution networks with a high penetration of distributed generations. IEEE Trans. Power Syst. 36(1), 438–450 (2020)
2. Ji, H., Wang, C., Li, P., et al: Robust operation of soft open points in active distribution networks with high penetration of photovoltaic integration. IEEE Trans. Sustain. Energy 10(1), 280–289 (2019)
3. Fu, X., Wang, C., Li, P., et al: Exponential integration algorithm for large-scale wind farm simulation with Krylov subspace acceleration. Appl. Energy 254, 113692, (2019)
4. Jing, R., Wang, X., Zhao, Y., et al.: Planning urban energy systems adapting to extreme weather. Adv. Appl. Energy 3, 100053 (2021)
5. Tian, X., Chi, Y., Li, Y., et al.: Coordinated damping optimization control of sub-synchronous oscillation for DFIG and SVG. CSEE J. Power Energy Syst. 7(1), 140–149 (2021)
6. Sharna, V., Aziz, S., Haque, M., et al.: Effects of high solar photovoltaic penetration on distribution feeders and the economic impact. Renew. Sustain. Energy Rev. 131, 110021 (2020)
7. Guillaud, X., Faruque, M., Teninge, A., et al. "Applications of real-time simulation technologies in power and energy systems. IEEE Power Energy Technol. Syst. J. 2(3), 103–115 (2015)
8. Newaz, J.O., Faruque, M.: Controller hardware-in-the-loop validation of a graph search based energy management strategy for grid-connected distributed energy resources. IEEE Trans. Energy Convers. 35(1), 520–528 (2019)
9. Chandak, S., Rout, P.: Seamless transition of microgrid between islanded and grid-connected mode of operation. IET Energy Syst. Integr. 3(1), 60–72, (2021)
10. Talha, M., Raihan, S., Rahim, N.: A grid-tied PV inverter with sag-severity-independent low-voltage ride through, reactive power support, and islanding protection. J. Mod. Power Syst. Clean Energy 9(6), 1300–1311, (2021)
11. Reddy, V.R., Sreeraj, E.S.: Grid voltage sensor-less protection scheme for one cycle-controlled single-phase photovoltaic inverter systems. CSEE J. Power Energy Syst. (2021), Early Access
12. Xu, J., Wang, K., Wu, P., et al: FPGA-based sub-microsecond-level real-time simulation for microgrids with a network-decoupled algorithm. IEEE Trans. Power Deliv. 35(2), 987–997 (2020)
13. Xu, J., Wang, K., Wu, P., et al: FPGA-based sub-microsecond-level real-time simulation of solid-state transformer with a switching frequency of 50kHz. IEEE J. Emerg. Sel. Topics Power Electron. 9(4), 4212–4224 (2020)
14. Moreira, F., Marti, J., Zaneta, L., et al: Multirate simulations with simultaneous-solution using direct integration methods in a partitioned network environment. IEEE Trans. Circuits Syst. I. 53(12), 2765–2778 (2006)
15. Galván, V., Martí, J., Dommel, H., et al.: MATE multirate modelling of power electronic converters with mixed integration rules. In: Proc. PSCC 2016, pp. 1–7 (2016)
16. Li, Y., Shu, D., Shi, F., et al: A multi-rate co-simulation of combined phasor-domain and time-domain models for large-scale wind farms. IEEE Trans. Energy Convers. 35(1), 324–334 (2020)
17. Multi-rate simulation. https://www.rtds.com/technology/simulation-algorithm/, [accessed 15 May 2021]
18. Zhai, X., Lin, C., Gregoire, L.: Multi-rate real-time simulation of modular multilevel converter for HVDC grids application. In: 43rd Annual Conference of the IEEE Industrial Electronics Society(IECON), Beijing, China, pp. 1325–1330 (2017)
19. Wang, Z., Wang, C., Li, P., et al: Extendable multirate real-time simulation of active distribution networks based on field programmable gate arrays. Appl. Energy 228, 2422–2436 (2018)
20. Duan, T., Shen, Z., Dinavahi, V.: Multi-rate mixed-solver for realtime nonlinear electromagnetic transient emulation of AC/DC networks on FPGA-MPSoC architecture. IEEE Power Energy Technol. Syst. J. 6(4), 183–194 (2019)
21. Kyesswa, M., Murray, A., Schmurr, P., et al: Impact of grid partitioning algorithms on combined distributed AC optimal power flow and parallel dynamic power grid simulation. IET Gener. Transm. Distrib. 14(25), 6133–6141 (2020)
22. Galvan-Sanchez, V., Marti, J., Banuelos-Cabral, E., et al: An asynchronous MATE-multirate method for the modeling of electric power systems. Electr. Eng. 103, 993–1007 (2021)
23. Han, J., Dong, Y., Miao, S., et al: Multi-rate electromagnetic transient parallel simulation of power system based on MATE. High Voltage Eng. 45(6), 1857–1865 (2019)
24. Tremblay, O., Rimorov, D., Gagnon, R., et al: A multi-time-step transmission line interface for power hardware-in-the-loop simulators. IEEE Trans. Energy Convers. 35(1), 539–548 (2020)

25. Li, P., Wang, Z., Wang, C., et al: Synchronization mechanism and interfaces design of multi-FPGA-based real-time simulator for microgrids. IET Gener. Transmiss. Distrib. 11(12), 3088–3096 (2017)
26. Zhu, K., Wong, D.F.: Clock skew minimization during FPGA placement. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 16(4), 376–385 (1997)
27. Kersting, W.: Radial distribution test feeders. IEEE Trans. Power Syst. 6(3), 975–985 (1991)
28. Wu, W., Li, P., Fu, X., et al. GPU-based power converter transient simulation with matrix exponential integration and memory management. Int. J. Electr. Power Energy Syst. 122, 106186 (2020)
29. Maguire, T., Giesbrecht, J.: Small time-step (<2μ Sec) VSC model for the real-time digital simulator. In: International Conference on Power System Transients (IPST), Montreal, Canada, pp. 1–6 (2005)
30. Hoang, L., Gilbert, S., Richard, G., et al: Real-time simulation of PWM power converters in a doubly-fed induction generator using switching-function-based models. In: Proc. IECON 2006, pp. 1878–1883 (2006)

How to cite this article: Fu, H., Li, P., Fu, X., Wang, Z., Yan, J., Wu, J., Wang, C.: Asynchronous multi-rate method of real-time simulation for active distribution networks. IET Renew. Power Gener. 1–12 (2022). https://doi.org/10.1049/rpg2.12500