Article

Multi-Rate Parallel Real-Time Simulation Method for Doubly Fed Wind Power Systems Based on FPGA–CPU

Guangrao Yang, Yahui Li, Zhenghang Hao, Zhuo Chen *, Puxiang He and Jing Zhang ©

College of Electrical Engineering, Guizhou University, Guiyang 550025, China
* Correspondence: zchen1@gzu.edu.cn

Abstract: A multi-rate parallel real-time simulation method based on FPGA–CPU is studied to realize the asynchronous co-simulation of the converter of doubly fed wind power systems with the wind turbine and external power grid. The doubly fed wind power system is partitioned by simulation step length, and the partitioned small-step-length data are processed using integral homogenization. For large-step data, an improved delay-compensated linear interpolation method combined with Newton interpolation is proposed for processing. The general small time-step (GST) model method is used to implement the FPGA modeling of the small-step converter, and resource optimization is achieved through timing time-division multiplexing. Asynchronous parallel co-simulation of a doubly fed wind power system is implemented on an FPGA–CPU co-simulation platform. Among them, the FPGA realizes the development of the converter HDL with a small step of 1 µs, while the CPU completes the simulation of the wind turbine and power grid synchronously with a large step of 50 µs. Finally, by comparing with MATLAB/Simulink offline simulation and analyzing the error, it is concluded that the simulation accuracy of the improved method in this paper is higher than that of the un-interpolated parallel simulation, which verifies the real-time performance and accuracy of the modeling and improved method in this paper.

Keywords: doubly fed wind power system; field-programmable gate array (FPGA); real-time simulation; multi-rate parallel simulation

1. Introduction

As one of the main methods of wind power generation, the transient real-time simulation technology of the doubly fed wind power system has gradually become the research focus of scholars in various countries [1,2]. Compared with the traditional offline simulation, the real-time simulation can reflect the operation mechanism of the model more accurately, better reflect the dynamic characteristics of the model, and make it closer to the actual situation of system operation [3]. Therefore, it is necessary to deeply understand the grid-connected characteristics and transient characteristics of wind power systems through efficient and accurate real-time simulation [4–6].

As an emerging and rapidly developing modern digital simulation technology, the digital twin has been widely used in industrial manufacturing [7,8]. Based on the digital simulation platform, this technology builds a mapping relationship between the real space and the virtual space and achieves the purpose of manipulating the actual equipment or system through the state feedback of the virtual space [9–11]. However, the scale of power systems is much larger, and the control systems are more complex than those of industrial manufacturing, so digital twin technology has not yet been fully implemented in power systems [12]. In addition, digital twins are mostly implemented based on machine learning (ML) [13,14], and ML involves complex algorithms, which adds an extra burden to the already complex doubly fed wind power system. Therefore, the use of real-time simulators for major simulation tasks remains a necessary task under current technology.
At present, commercial real-time simulation machines represented by RTDS, RT-LAB and HYPERSONIC have been widely used in the fields of control of doubly fed wind power systems [15], modeling of photovoltaic power plants [16], and model partitioning of power systems [17]. These simulators represent the CPU-based development of real-time simulation models, limited by the serial simulation structure of the CPU, and the simulation step length of such processors can only reach tens of microseconds in most cases [18]. However, the typical power electronic converter switching frequency has been as high as 10~200 kHz [19], and the development of high-frequency converter technology has brought great challenges to the real-time simulation of power electronics. To reduce the simulation step, RTDS and RT-LAB have introduced field-programmable gate arrays (FPGAs) as peripheral devices to undertake the simulation of high-frequency power electronics devices [20]. Compared to commercial simulators, FPGA has a fully configurable parallel hardware structure, distributed memory structure and pipeline structure [21]. The highly parallel numerical computation enables FPGA to perform microsecond and even nanosecond simulations. In addition, FPGA has the advantages of small size, low price, rich I/O interfaces, and short development cycle, and it has gradually become one of the main devices for real-time simulation [22].

However, for large systems containing high-frequency power electronics, if the whole system is simulated in small steps, the simulation scale will be severely limited [23]. Therefore, multi-rate co-simulation with multiple processors has gradually become the trend in real-time simulation platforms [24,25]. Ref. [26] proposed a multi-FPGA real-time simulation architecture, using multiple FPGAs as parallel computing units for real-time simulation, which effectively increases the simulation resources and reduces the simulation step size. Ref. [27] combines the advantages of the strong computational power of FPGA and the flexibility of RTDS and uses FPGA as the external expansion device of RTDS to realize the parallel real-time simulation of FPGA-RTDS. In multi-rate co-simulation, the interactive processing of parallel data is particularly important. The authors of [28] proposed a real-time simulation method based on a CPU-FPGA heterogeneous platform, which allocates different simulation tasks to different processors and greatly reduces the simulation burden of a single processor. However, this method directly takes the result of the last simulation step on the opposite side as the input during the simulation process, without further considering the timing error during the multi-rate decoupled parallel simulation. In [29], the authors proposed a multi-rate real-time simulation method based on Norton equivalence. The data interaction between asynchronous length systems is realized by linear extrapolation, and the multi-rate parallel simulation of FPGA-RTDS is achieved. However, the method does not consider the compensation of the predicted data on both sides, and there is a certain simulation error. In [30], the authors used an improved linear interpolation algorithm combined with Lagrange interpolation to achieve parallel interaction of asynchronous long data, which greatly improved the simulation accuracy based on the literature [29]. Unfortunately, the method did not process small-step data, and the Lagrange interpolation algorithm is computationally intensive and tends to occupy a large number of FPGA resources.

Based on the above research theories, this paper takes a doubly fed wind power system as an example and divides the system into two simulation subsystems with different step lengths based on the continuous–discrete model separation (CDMS) method. To reduce the simulation error, the partitioned small-step data are processed using integral average, and on this basis, an improved linear interpolation algorithm combined with Newton interpolation is used to realize the parallel interaction of the asynchronous step-length data. The general small time-step (GST) model is used to model the back-to-back converters in the doubly fed wind power system, and the model is optimized based on the time-sharing multiplexing principle. The asynchronous parallel real-time simulation of the whole doubly fed wind power system is performed by the FPGA–CPU platform. In particular, the hardware description language (HDL) of the converter is developed on the FPGA in small steps of 1 µs, and the 50 µs simulation of the wind turbine and the external power
grid is performed simultaneously in the CPU. By comparing with MATLAB/Simulink offline simulation results and analyzing the simulation errors, the real-time capability and accuracy of the modeling and improved methods in this paper are verified.

The remainder of this paper is organized as follows. Section 2 introduces the basic structure of the doubly fed wind power system. Section 3 performs model partitioning of the doubly fed wind power system and completes parallel processing of asynchronous length data on this basis. Section 4 implements the FPGA modeling of the back-to-back converter and the optimization of its hardware resources. Section 5 completes the FPGA–CPU asynchronous parallel real-time simulation and compares and analyzes the simulation results. Section 6 concludes the paper.

2. Basic Structure of Doubly Fed Wind Power System

As shown in Figure 1, The doubly fed wind power system consists of four parts: the turbine, the power grid, the back-to-back converter, and the control system. The turbine drives the rotor of the induction motor through a gearbox, and the rotor side is connected to the grid through a back-to-back converter, while the stator side is directly connected to the grid.

![Figure 1. The fundamental structure of a doubly fed wind power system.](image-url)

In the figure, parameters $u_{ref \_g}$ and $u_{ref \_r}$ represent the modulating voltages from the control system on the grid side and rotor side, respectively, and PWM1 and PWM2 are the trigger pulses of the converter on the grid side and rotor side. To understand the basic structure of the system in more detail, the back-to-back converter is listed separately. As shown in Figure 2, it consists of three sets of insulated-gate bipolar transistors (IGBTs) that are completely symmetrical about the DC bus capacitance. In this case, the modulating waves on both sides are compared with the high-frequency carrier waveform to generate three pulse signals, $g_{(1,3,5)}$ and $g'_{(1,3,5)}$, respectively. Considering the trigger pulses of the upper and lower bridge arms of each group of converters as ideal complementary PWM waves, the control signals of the whole back-to-back converters can be obtained by inverting the three-way signals by bit. In the figure, C indicates the DC bus capacitor, $g_{1}$ indicates the trigger pulse of the upper bridge arm of the first converter, and $g_{6}$ indicates the trigger pulse of the lower bridge arm of the second converter.
Therefore, a multi-rate co-simulation method can be used to simulate the whole system, i.e., different simulation steps for different time constants of the simulation subsystem, i.e., different simulation steps for different time constants of the simulation subsystem [31].

### 3. Multi-Rate Parallel Simulation of Doubly Fed Wind Power System

Due to the difference in simulation performance, in this paper, the simulation step for the back-to-back converter part is set to $\Delta t$, while the other parts are simulated with a large step of $\Delta T$, $\Delta T$ is an integer multiple of $\Delta t$.

#### 3.1. Multi-Rate Parallel Decoupling of Doubly Fed Wind Power System

For different step simulation systems, realizing multi-rate decoupled simulation of the system can reflect the high-frequency transient processes while effectively improving the simulation computational efficiency. A continuous–discrete model separation (CDMS) method is proposed in [32], as shown in Figure 3.

![Figure 3. CDMS modeling block diagram.](image)

The main idea is to use an ideal current source to replace the discrete circuit model in the system, and then separate the switching devices from the whole system. The switching model $G(V_{SW})$ is used to describe the voltage–current relationship of the switching device, and its calculated current $I(V_{SW})$ is used as an input to the circuit model equation $F(V_{SRC}, I_{SRC}, I_{SW}, X)$. $F(V_{SRC}, I_{SRC}, I_{SW}, X)$ then calculates the voltage of the switching device and uses this voltage as the input to the equation $G(V_{SW})$.

In this paper, we improve the CDMS to decouple the doubly fed wind power system from the multi-rate simulation. As shown in Figure 4, the entire doubly fed wind power system is divided into two subsystems according to the simulation step, namely, the large-step turbine and power grid subsystem 1 and the small-step back-to-back converter and control module subsystem 2. The data interaction between the subsystem interfaces is...
achieved using controlled sources. In particular, the grid and rotor sides of the turbine receive the phase voltages from the back-to-back converter and use them as the control signal for the controlled voltage source on both sides. The back-to-back converter receives the three-phase currents from the grid side and rotor side and uses them as the control signals of the controlled current sources, thus realizing the decoupling of the whole system at different steps.

During real-time simulation, large- and small-step data need to be exchanged synchronously. However, the small-step data changes several times in a large-step cycle, and the data is passed out in the last small step before the large-step update. This results in the loss of many small-step data, which affects the simulation accuracy. In addition, the inherent delay in data transmission causes the data received on the large-step side to lag by one synchronous clock cycle, increasing the simulation error. Therefore, in this paper, the parallel data on the large- and small-step sides are processed for error compensation.

3.2. Parallel Processing of Data with Different Step Lengths

During real-time simulation, large- and small-step data need to be exchanged synchronously. However, the small-step data changes several times in a large-step cycle, and the data is passed out in the last small step before the large-step update. This results in the loss of many small-step data, which affects the simulation accuracy. In addition, the inherent delay in data transmission causes the data received on the large-step side to lag by one synchronous clock cycle, increasing the simulation error. Therefore, in this paper, the parallel data on the large- and small-step sides are processed for error compensation.

3.2.1. Small-Step PWM Equalization

Based on the principle of area equivalence, the data on the small-step sides are processed by integration averaging. The data output from subsystem 2 can all be considered as discrete small-step PWM signals of equal width and unequal amplitude, so the integration averaging process can be considered as an amplitude averaging process in combination with the time average method (TAM) [33].

**Figure 4.** Principle diagram of decoupling system with different step lengths.

In the above decoupling process, the phase voltage data such as \( v_{ab,g} \) received by subsystem 1 will be discrete small-step PWM signals due to the influence of the PWM wave of the control module, while the current data such as \( i_{abc,g} \) received by subsystem 2 will be continuous large-step signals. Because of the difference in simulation steps, the simulation process will lead to delays in large-step data and loss of small-step data, which will seriously affect the simulation accuracy, so it is necessary to compensate for the simulation errors.

**Figure 4.** Principle diagram of decoupling system with different step lengths.

In the above decoupling process, the phase voltage data such as \( v_{ab,g} \) received by subsystem 1 will be discrete small-step PWM signals due to the influence of the PWM wave of the control module, while the current data such as \( i_{abc,g} \) received by subsystem 2 will be continuous large-step signals. Because of the difference in simulation steps, the simulation process will lead to delays in large-step data and loss of small-step data, which will seriously affect the simulation accuracy, so it is necessary to compensate for the simulation errors.

**Figure 4.** Principle diagram of decoupling system with different step lengths.

In the above decoupling process, the phase voltage data such as \( v_{ab,g} \) received by subsystem 1 will be discrete small-step PWM signals due to the influence of the PWM wave of the control module, while the current data such as \( i_{abc,g} \) received by subsystem 2 will be continuous large-step signals. Because of the difference in simulation steps, the simulation process will lead to delays in large-step data and loss of small-step data, which will seriously affect the simulation accuracy, so it is necessary to compensate for the simulation errors.

**Figure 4.** Principle diagram of decoupling system with different step lengths.
Subsystem 2 sends $\bar{v}$ ahead of time, and subsystem 1 uses this value in the next large step for simulation. The above process is repeated in each $\Delta T$ thereafter to obtain the mean value of the small-step effect for subsystem 2, which indirectly reflects the current state of the small-step model. A series of equal-width and unequal-amplitude pulse data is obtained on the large-step side, thus enabling asynchronous data interaction.

### 3.2.2. Large-Step Prediction and Delay Compensation

Although the small-step PWM averaging can reflect the operation of the small-step model very well, it also brings a corresponding transmission delay.

As shown in Figure 6, in the real-time simulation, subsystem 1 samples and sends data at the beginning of each large-step cycle, which requires subsystem 2 to send out the small-step averages before the sampling point. Assuming that the data is sent in advance $T_f$, generally speaking, $T_f$ is larger than the simulation step $\Delta t$ of subsystem 2. As can be known from the previous subsection, the small-step averaging is performed in the last $\Delta t$ of each $\Delta T$, which makes the small-step averaging in the current cycle delayed by one $\Delta T$ before it is transmitted to the large-step subsystem, thus causing simulation errors.

![Figure 5: Small-step PWM effect averaging schematic.](image)

![Figure 6: Large-step data delay schematic.](image)
To address the above errors, this paper uses the three-point interpolation method to predict and compensate for the large-step data.

\[
i_{est}(t) = \frac{5}{4}i(t - \Delta T) + \frac{1}{2}i(t - 2\Delta T) - \frac{3}{4}i(t - 3\Delta T) \quad (2)
\]

\[
i_{est}(t + \Delta T) = \frac{5}{4}i(t) + \frac{1}{2}i(t - \Delta T) - \frac{3}{4}i(t - 2\Delta T) \quad (3)
\]

where \(i_{est}(t), i_{est}(t + \Delta T)\) are the predicted values of current at the present moment and the next moment, respectively. The three-point interpolation method has some errors in the prediction of data, and this paper uses error delay compensation to correct the prediction values. Since the current is a periodically varying sine wave, the prediction error of the current moment can be delayed by one present period to compensate for the prediction value of the next moment. Assuming that the initial value is zero, the current prediction correction value can be expressed as Equation (4) in the complex frequency domain.

\[
i_{cor1}(z) = z i_{est}(z) + z^{-(\lambda - 1)}(i(z) - i_{est}(z)) \quad (4)
\]

where \(i_{cor1}(z)\) is the predicted correction value for the next moment, and \(\lambda\) is the value of latency. In the case of back-to-back converters, the received data delay is influenced by the operating frequency of the system on the grid and rotor sides, as shown in Equation (5).

\[
\lambda = \begin{cases} 
\frac{1}{\Delta T} & \text{grid side} \\
\frac{1}{s\Delta T} & \text{rotor side}
\end{cases}
\]

where \(f\) is the operating frequency of the grid side system, and \(s\) is the slip ratio. After the large-step prediction and compensation, the data synchronization process between large and small steps can be realized. As expressed in Figure 6, the large-step data sent by subsystem 1 at the moment \(t_2\) is the corrected value \(i_{cor1}(t_2)\) at the present moment. Subsystem 2 finishes receiving the corrected values at the moment \((t_2 + T_1)\) and uses them for the small-step simulation between \((t_2 + T_1 \sim t_4 + T_1)\), and finishes the averaging process of the small steps in the last \(\Delta T\) of this \(\Delta T\). Subsystem 2 sends out the mean value \(\tau_2\) earlier at moment \(t_5\), and subsystem 1 finishes receiving it at moment \(t_6\). \(\tau_2\) is the result of \(i_{cor1}(t_2)\)'s participation in the small-step subsystem 2 simulation, i.e., the simultaneous interaction of data of different steps in the previous \(\Delta T\) is achieved at moment \(t_6\).

### 3.2.3. Small-Step Data Interpolation

The simulation accuracy of the system is greatly improved after the small-step averaging process, large-step prediction, and compensation operations. However, the small-step system receives the large-step data and holds it until the arrival of the next large-step data. This leads to the small-step module performing the same calculation several times, which wastes simulation resources and does not take advantage of the small-step simulation. In view of this, this paper uses the Newton interpolation method to correct the small-step data. Compared with other interpolation algorithms, Newton interpolation is less computationally intensive, easy to program and more suitable for real-time simulation [34]. The Newton interpolation polynomial is shown in Equation (6) [35].

\[
f(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \cdots + \sum_{n=0}^{\lambda} f(x_0, x_1, \ldots, x_n)(x - x_0)(x - x_1) \cdots (x - x_n) \quad (6)
\]

where \(f(x_0, \ldots, x_n)\) is the nth order difference quotient of the function \(f\). Since the small-step subsystem uses a fixed-step simulation, the Newtonian pre-interpolation formula for equidistant nodes can be used to interpolate the large-step data. Considering that the interpolated function has sinusoidal characteristics and the computational load of the real-time simulation should not be too large, a two-order polynomial is chosen to implement
the interpolation. Before interpolation, it is also necessary to extend the known interval, and by combining Equations (3) and (4), it is obtained that:

\[ i_{\text{est}}(t + 2\Delta T) = \frac{5}{4} i_{\text{cor}}(t + \Delta T) + \frac{1}{2} i(t) - \frac{3}{4} i(t - \Delta T) \]  

(7)

\[ i_{\text{cor}}(z) = z^2 i_{\text{est}}(z) + z^{-(l-2)}(i(z) - i_{\text{est}}(z)) \]  

(8)

In Equation (8), \( i_{\text{cor}}(z) \) is the predicted corrected value for the next two moments. From this, the two-order Newtonian pre-interpolation formula for large-step currents is obtained as:

\[ i_2(t_1 + \delta \Delta T) = i(t_1) + [i_{\text{cor}1}(t_1 + \Delta T) - i(t_1)] \delta \Delta T + \frac{i_{\text{cor}2}(t_1 + 2\Delta T) - 2 i_{\text{cor}1}(t_1 + \Delta T) - i(t_1)}{2!} \delta \Delta T (\delta \Delta T - 1) \]  

(9)

where \( i_2(t_1 + \delta \Delta T) \) denotes the multiple small-step current values obtained using quadratic Newton pre-interpolations after the \( t_1 \) moment.

\[ \delta = (1 + h \frac{\Delta t}{\Delta T}) h = (1, 2, \ldots, \frac{\Delta T}{\Delta T}) \]  

(10)

The principle of small-step interpolation is shown in Figure 7.

![Figure 7. Newton interpolation with small steps.](image)

The solid black line of subsystem 1 indicates the actual value of the large-step current, and the corrected values \( i_{\text{cor}1}(t_2) \) and \( i_{\text{cor}2}(t_2) \) at moments \( (t_2 + \Delta T) \) and \( (t_2 + 2\Delta T) \) are obtained by three-point interpolation. The corrected values are advanced one sampling step for the interpolation condition, i.e., \( i_{\text{cor}1}(t_2) \) is advanced to moment \( t_2 \) and \( i_{\text{cor}2}(t_2) \) is advanced to moment \( t_3 \). Newton interpolation is performed on the interval \([t_2 \sim t_3] \), and the interpolation results are applied to the simulation of the small-step subsystem. This enables the linearization of the large-step data and improves the simulation accuracy.

4. FPGA-Based Back-to-Back Converter Modeling and Hardware Resource Optimization

After parallel processing of the asynchronous data as described above, the system simulation tasks need to be distributed. In this paper, the small-step simulation of the back-to-back converter is implemented using FPGA, so the converter needs to be modeled for its HDL development. Since FPGA is highly parallel and has limited logic resources, the same parts of the converters can be simplified to save resources.

4.1. Common Switching Device Modeling Methods

At present, modeling methods for switching devices can be broadly classified into two types: mechanistic models and behavioral models [36]. The mechanistic model focuses on
the internal transient characteristics of the switching device, while the behavioral model focuses on the external dynamic characteristics of the switching device. The latter is more suitable for FPGA real-time simulation. The behavioral models can be generally divided into three types: the average-value model method, the two-value conductance model method, and the switching function method [37].

The average-value model method averages one cycle of the switching device and replaces the actual value output with the average value. This modeling method does not take into account the high-frequency characteristics of the switching device and has low accuracy. The switching function method lists the basic equations of the simulation based on the physical characteristics of the switching device and the circuit topology. This method is more accurate, but the relationship between the system state variables needs to be known, and it is more difficult to model complex circuits. The two-value conductance method equates the on/off state of the switching device to a large/small resistance, respectively, which is simple to model and easy to implement. However, it wastes a lot of time and resources in updating and inverting the system matrix, and the method also causes the system equations to be pathological, resulting in numerical instability [38]. Therefore, some scholars have proposed the associated discrete circuit (ADC) method on this basis, which is to equate the on/off state of the switching device to small inductance/small capacitance, respectively. Using the numerical integration algorithm, the discretized branch circuit is equated to the parallel form of the conductor and the historical current source [39]. By calculating the parameters of each node, the overall transient simulation equation of the system is obtained.

\[ Gu = i \]  

(11)

where \( G \) is the conductance matrix, which is kept constant during the calculation, and \( u \) and \( i \) denote the voltage and current of each node, respectively. This method has a simple modeling idea and does not need to consider the operation state of the whole circuit, but the simulation accuracy is easily affected by the choice of parameters, and virtual power loss will be generated [18]. Based on this method, Ref. [40] proposes a fixed-admittance modeling method for power electronic converters using response-matching, that is, the general small time-step (GST) model method [41]. This method aims to eliminate transient errors and no longer relies on specific physical circuits and numerical integration methods, and can accurately simulate ideal switching characteristics through simple settings. Table 1 gives a comparison of the performance of various switching device modeling approaches.

<table>
<thead>
<tr>
<th>Modeling Method</th>
<th>Modeling Basis</th>
<th>Complexity</th>
<th>High-Frequency Characteristics</th>
<th>Simulation Accuracy</th>
<th>Generality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average-value model</td>
<td>Periodic averages instead of actual values</td>
<td>Moderate</td>
<td>No</td>
<td>Low</td>
<td>Moderate</td>
</tr>
<tr>
<td>Switching function method</td>
<td>Circuit topology</td>
<td>Complex</td>
<td>Yes</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Two-value conductance method</td>
<td>On/off states are equivalent to different resistances</td>
<td>Simple</td>
<td>Yes</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Associated discrete circuit (ADC)</td>
<td>On/off state equivalent to inductor/capacitor</td>
<td>Moderate</td>
<td>Yes</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>General small time-step (GST)</td>
<td>On/off state equivalent to on/off element</td>
<td>Simple</td>
<td>Yes</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

In summary, to obtain better simulation accuracy and wider application scenarios, the GST model method is chosen for modeling, HDL development, resource optimization, and experimental verification of the back-to-back converters.
4.2. Modeling of Back-to-Back Converters

4.2.1. The GST Model

In the GST model based on response-matching, the on/off state of the switching device is no longer equated to a small inductor/small capacitor, but the on/off of the switching device is replaced by the closing/breaking element, respectively. The closing/breaking element is equated as a parallel form of conductor and historical current source, as shown in Figure 8.

\[
\begin{align*}
\text{Switch on} & : \quad Y_{SW_{on}} \text{ and } i(t) + u(t) = i_{his_{on}} \\
\text{Switch off} & : \quad Y_{SW_{off}} \text{ and } i(t) \quad \text{with } i_{his_{off}} \\
\end{align*}
\]

Figure 8. The GST model.

To ensure the constant conductance characteristics of the model, the Dommel equivalent conductance when the switch is on and off should be equal, differing only in the historical current source, i.e., \(Y_{SW_{on}} = Y_{SW_{off}} = Y_{SW}\), which takes the following values:

\[
\frac{\Delta t}{L_f} \ll Y_{SW} \ll \frac{C_f}{\Delta t}
\]  

(12)

where \(L_f\) and \(C_f\) denote the filter inductor and DC capacitance, respectively. Therefore, the relationship between the historical current source and the voltage and current of the previous step can be expressed as shown in Equation (13).

\[
\begin{align*}
\begin{cases}
    i_{his_{on}}(t) = a_{on}Y_{SW}u(t - \Delta t) + \beta_{on}i(t - \Delta t) \\
    i_{his_{off}}(t) = a_{off}Y_{SW}u(t - \Delta t) + \beta_{off}i(t - \Delta t)
\end{cases}
\end{align*}
\]

(13)

In the modeling process, the voltage coefficient \(a\) and current coefficient \(\beta\) are chosen as the optimal damping parameters for the fixed-admittance model of the two-level converter.

\[
\begin{align*}
\begin{cases}
    a_{on} = -1 + \sqrt{2} & a_{off} = 1 \\
    \beta_{on} = -1 & \beta_{off} = 1 + \sqrt{2}
\end{cases}
\end{align*}
\]

(14)

4.2.2. Converters Modeling Based on GST Model

Based on the above GST model, the back-to-back converter in the doubly fed wind power system is modeled. The back-to-back converter consists of a DC bus capacitor and two groups of symmetrical IGBTs, and the GST model is used to replace each IGBT and DC capacitor to obtain the simulation topology shown in Figure 9.
In the figure, whether the historical current source is \( i_{\text{his\_on}} \) or \( i_{\text{his\_off}} \) will depend on whether the switching device is on or off, i.e., determined by the trigger pulse of the IGBT. Through the relationship between voltage and current, combined with Equations (11) and (13), the 13th-order iterative equation of the back-to-back converter can be obtained.

\[
\begin{align*}
\begin{bmatrix}
Y_{sw} & Y_{sw}Y^T & -E & 0 & 0 \\
Y_{sw} & 0 & E & 0 & Y
\end{bmatrix}
\begin{bmatrix}
V_{g}^{n+1} \\
V_{r}^{n+1}
\end{bmatrix}
+ \begin{bmatrix}
Y_{dc} & Y_{dc}Y & 0 & E & 0 \\
E & 0 & Y
\end{bmatrix}
\begin{bmatrix}
I_{g}^{n+1} \\
I_{r}^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
I_{\text{his\_lg}} \\
I_{\text{his\_hr}}
\end{bmatrix}
\end{align*}
\]

(15)

where \( E \) is the third-order unit matrix, \( Y = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} \) is the coefficient matrix of the current; \( Y_c \) is the equivalent conductance of the DC capacitor after discretization by the trapezoidal integration method, \( v_{dc}^{n+1} \) and \( i_{\text{his\_c}} \) are its voltage and historical current sources; \( V_{g}^{n+1} \) and \( V_{r}^{n+1} \) are the node voltages of the grid/rotor side of the back-to-back converter, \( I_{g}^{n+1} \) and \( I_{r}^{n+1} \) are the input currents of the upper bridge arm of the grid/rotor side; \( I_{\text{his\_lg}} \) and \( I_{\text{his\_hr}} \) are the historical current sources of the upper bridge arm of the grid/rotor side; and the expressions of the voltage and current matrices are shown in Equation (16).

\[
\begin{align*}
V_{g}^{n+1} &= \begin{bmatrix} v_{g1}^{n+1} & v_{g2}^{n+1} & v_{g3}^{n+1} \end{bmatrix}^T \\
V_{r}^{n+1} &= \begin{bmatrix} v_{r4}^{n+1} & v_{r5}^{n+1} & v_{r6}^{n+1} \end{bmatrix}^T \\
I_{\text{his\_lg}} &= I_{abc\_g} + I_{\text{his\_dg}} \\
I_{\text{his\_hr}} &= I_{abc\_r} + I_{\text{his\_dr}}
\end{align*}
\]

(16)

where \( v_{g}^{n+1} = v_{r}^{n+1} = v_{k}^{n+1} \) (\( k = 1 \cdots 6 \)) are the node voltages and upper bridge arm branch currents, \( I_{abc\_g}, I_{abc\_r} \) are the three-phase input currents on the grid/rotor side, and \( I_{\text{his\_dg}}, I_{\text{his\_dr}} \) are the historical current sources on the lower bridge arm on the both sides.

From Equation (15) to (16), it can be seen that the system conductance matrix remains constant during the modeling of the back-to-back converter using the GST model method. The modeling only requires the trigger signals of each switching device and the input currents on both sides. The trigger signal is used to control the change in the historical current source and is generated by the modulating waveform \( u_{\text{ref}} \). This modeling method does not require knowing the logical relationship between the system state variables, and is simple and general, which is very suitable for HDL development on FPGA. In addition, according to the symmetry of the back-to-back converter, it can also be optimized for FPGA resources.
4.3. Optimization of FPGA Resources for Back-to-Back Converter

4.3.1. Model Optimization Basis

Since the mathematical models of the grid side and the rotor side of the back-to-back converter are the same, the two sides of the converter can be combined into one for modeling and simulation by time-sharing. Considering the clock periods of the large and small-step systems as $\Delta T$ and $\Delta t$, respectively, the asynchronous simulation timing of the model can be obtained as shown in Figure 10.

![Figure 10. System and converter operation timing diagram.](image)

The system starts simulation at the moment $t_0$. The periods $t_0$ to $t_1$ and $t_1$ to $t_2$ are the operating intervals of the grid side and rotor side converters, respectively. When the grid side is working, the rotor side data is maintained as the simulation result of the previous step, and the same for the rotor side, and the length of the operating interval for each side converter is $\Delta t$. When the simulation time reaches $t_3$, the whole system completes a large-step simulation, and the two converters alternately perform $n/2$ calculations, $n = \Delta T/\Delta t$. The process is repeated in each subsequent large-step cycle, thus reducing the number of back-to-back converters by half for optimization purposes.

4.3.2. Resource Optimization of the Model

From Equation (15), it can be seen that the admittance matrix of the system has symmetry, and Equation (15) can be simplified to a seventh-order iterative equation based on the above optimization principle.

$$
\begin{bmatrix}
Y_{sw}E & Y_{sw}Y^T & -E \\
Y_{sw}E & 0 & E \\
0 & Y_c & Y
\end{bmatrix}
\begin{bmatrix}
V^{n+1} \\
I^{n+1}_r \\
I^{n+1}_g
\end{bmatrix} =
\begin{bmatrix}
I_{his_lgr} \\
I_{his_lgr} \\
I_{his_cd}
\end{bmatrix}
$$

(17)

where $V^{n+1}$, $I^{n+1}_r$, and $I_{his_lgr}$ are results of alternating $V^{n+1}_g$ and $V^{n+1}_r$, $I^{n+1}_r$ and $I^{n+1}_g$, $I_{his_lg}$ and $I_{his_lr}$ in Equation (16) in time sequence, respectively, and $I_{his_cd}$ is the sum of the upper bridge arm branch current on the working side within the previous $\Delta t$ and the DC capacitor history currents. The expression is shown in Equation (18).

$$
i_{his_cd} = i_{his_c} + i_{1(4)}^{n+1} + i_{2(5)}^{n+1} + i_{3(6)}^{n+1}
$$

(18)

where $i_{1(4)}^{n+1}$ denotes the branch current of bridge arm 1 or the branch current of bridge arm 4, the value of which is selected by the alternate timing.
The block diagram of hardware resource optimization for the back-to-back converter can be obtained as shown in Figure 11, where the inputs are currents and pulse modulated voltages on the grid/rotor side, respectively. Subsystem 1 obtains corrected data such as $u_{ref,gcor}$ by predicting and compensating for the large-step data. Subsystem 2 performs Newton interpolation on the corrected data to obtain small-step data such as $u_{ref,gN}$. The multiplexer realizes the time-sharing input of data, and the data on both sides are time-sharing multiplexed with a PWM generator and simplified converter model. The output value of the converter is used as the history term parameter of the model after delay. The data distributor distributes the data such as $v_{n+1}$ to the grid side and rotor side in time order, and the node voltages on both sides are passed through the subtractor to obtain the phase voltages of each phase. The output phase voltages $v_{ab,g}, v_{bc,g}, v_{ab,r}, v_{bc,r}$ and the DC capacitor voltage $v_{dc}$ are transformed from small to large step by small-step integral averaging process and participate in the simulation of subsystem 1 with a step size of $\Delta T$. As a result, the number of converters and PWM generators is reduced by half, which reduces the consumption of FPGA resources. In addition, the model built by the non-interpolated GST model can also be optimized for FPGA resources in the same way, and the optimized model is used for simulation in this paper.

![Figure 11. Block diagram of FPGA resource optimization for back-to-back converters.](image)

5. Experimental Verification

5.1. Construction of Experimental Platform

Based on the converter model built using the GST method and the hardware resource optimization theory of FPGA, this paper selects HDL for 1 $\mu$s FPGA development of subsystem 1 and completes 50 $\mu$s simulation of subsystem 2 using CPU synchronously. FPGA is Kintex-7 series xc7k325t from Xilinx, package type and pin count is ffg676, speed level is $-2$, and the $100$ MHz simulation main clock is generated through a phase-locked loop (PLL). The simulation CPU is the Universal Real-time Experimental Platform (UREP) independently developed by our group, which has the capability of real-time digital simulation and semi-physical simulation of the full electromagnetic transient state of 10,000-node level grid-scale and 1 million kilowatt-level level new energy and energy storage equipment. The system composition of the experimental platform is shown in Figure 12.

Among them, the PC implements the generation of the hardware description language of the back-to-back converter and the initialization of the UREP model. The UREP and the FPGA realize the data transfer in large and small steps via Ethernet. In addition, the PC monitors in real time the signals to be observed for the system in the UREP. The FPGA–UREP simulation physical platform is shown in Figure 13.
writes these data into the asynchronous FIFO at the PHY operating frequency (125 MHz). (Port Physical Layer) receives data from the UREP through the UDP receiving module and for UDP sending data is shown in Figure 14. The actual data sent is UDP data, as shown for UDP sending data is shown in Figure 14. The actual data sent is UDP data, as shown in the light red box in the flow chart. The process of receiving data is basically the same as sending, the difference is that the receiving process adds a data bit counter, which is not repeated here.

The block diagram of FPGA and UREP asynchronous co-simulation is shown in Figure 15. After both UREP and FPGA are started, UREP first sends an ARP (Address Resolution Protocol) request to the FPGA. The FPGA makes an ARP reply and links to UREP, The UREP first sends an ARP request to the FPGA, and the FPGA makes an ARP reply and links to the UREP, subsequently establishing communication between the two. The UREP runs subsystem 1 in 50 µs steps and sends out the data obtained from the operation, such as \( u_{ref_{3v}} \) through the UDP sending module. The FPGA data receiving/sending chip PHY (Port Physical Layer) receives data from the UREP through the UDP receiving module and writes these data into the asynchronous FIFO1 at the PHY operating frequency (125 MHz).
When the data is received, all the data in FIFO1 is read out at the same time using the main frequency operation (100 MHz) of FPGA to ensure the synchronization of data. The readout data is assigned to the back-to-back converter as input and run in subsystem 2 with a 1 μs step. When the simulation counter reaches 45 μs, the FPGA writes the $v_{dc}$ and other small-step data obtained from the run to the asynchronous FIFO2 one by one at its main frequency. The UREP receives data from the FPGA while sending out other data at the next 50 μs, and uses the received data for the simulation of subsystem 1. The above process is repeated in each $\Delta T$ thereafter, thus achieving asynchronous parallel co-simulation of FPGA–UREP.

**Figure 14.** Flow chart showing UDP sending data.

**Figure 15.** FPGA–UREP asynchronous parallel simulation framework.
5.3. Analysis of Co-Simulation Results

In this paper, the rated operating voltage of the external power grid is 120 kV, which is reduced to the rated voltage of 575 V of the wind turbine after passing through multiple transformers. The number of wind turbines is 6. The wind speed is constant at 15 m/s, the rated output power of the single machine is 1.5 MVA, and the power factor is 0.9. The DC bus capacitance of the back-to-back converter is 60 mF, and its stable working voltage is 1150 V.

5.3.1. Simulation Accuracy Analysis

To verify the real-time traceability of the interpolation and optimization method of this paper on the transient process of the system, the following two scenarios are set up in the external power grid and the wind turbine connection point, respectively.

Scenario A: A voltage loss fault occurs in the external power grid at the moment of 1.5 s, the voltage drop is 0.8 times the rated voltage, and the fault duration is 0.05 s, the simulation results obtained are shown in Figure 16.

Scenario B: A phase-to-ground short-circuit in phase A occurs at the wind turbine connection point at the moment of 2 s, and the fault duration is 0.01 s. The simulation results obtained are shown in Figure 17.

Figure 16. Comparison of simulation waveforms in scenario A: (a) Rotor-side current waveform; (b) Connection point current waveform; (c) DC capacitance-voltage waveform; (d) P, Q waveform.

Scenario B: A phase-to-ground short-circuit in phase A occurs at the wind turbine connection point at the moment of 2 s, and the fault duration is 0.01 s. The simulation results obtained are shown in Figure 17.
In Figures 16 and 17, P represents active power and Q represents reactive power. The symbol Simulink shows the offline simulation results obtained by using MATLAB/Simulink library component IGBT modeling with a simulation step of 5 µs; GST and GST-N2 denote the FPGA–CPU real-time simulation results obtained using the GST model method modeling and the prediction compensation combined with Newton interpolation, respectively. Taking the simulation data within 0.2 s before and after the fault moment as steady-state and transient samples, respectively, we compared the Euclidean norm errors [42] of the rotor side A-phase current $i_{ar}$, grid side A-phase current $i_{ag}$, and active power P with the offline simulation results, as shown in Table 2.

**Table 2.** Comparison of paradigm errors in different simulation scenarios.

<table>
<thead>
<tr>
<th>Simulation Scenarios</th>
<th>Modeling Method</th>
<th>$i_{ar}$ (%)</th>
<th>$i_{ag}$ (%)</th>
<th>P (%)</th>
<th>$i_{ar}$ (%)</th>
<th>$i_{ag}$ (%)</th>
<th>P (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario A</td>
<td>GST</td>
<td>1.22</td>
<td>0.69</td>
<td>0.12</td>
<td>4.61</td>
<td>2.33</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>GST-N2</td>
<td>1.20</td>
<td>0.62</td>
<td>0.08</td>
<td>2.21</td>
<td>0.67</td>
<td>0.08</td>
</tr>
<tr>
<td>Scenario B</td>
<td>GST</td>
<td>1.06</td>
<td>0.67</td>
<td>0.08</td>
<td>9.47</td>
<td>3.13</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>GST-N2</td>
<td>1.03</td>
<td>0.63</td>
<td>0.05</td>
<td>2.54</td>
<td>0.93</td>
<td>0.10</td>
</tr>
</tbody>
</table>

From Figures 16 and 17 and Table 2, it can be seen that both modeling methods can track the offline simulation results well in the steady state. When a fault occurs,
the transient paradigm error modeled by the GST-N2 interpolation algorithm is reduced by more than half compared to that modeled by the un-interpolated GST, i.e., GST-N2 is closer to the offline simulation. In addition, the model established by the GST method appears to gradually offset the offline simulation after the occurrence of the fault. This verifies the reliability and accuracy of the interpolation algorithm in this paper, i.e., it has high simulation accuracy.

5.3.2. Simulation Time Consumption Analysis

During the FPGA–UREP co-simulation, the first communication consumes 0.28 µs each for ARP request and reply, and then the link is established. In each 50 µs large-step cycle, the UDP receiving time is 0.86 µs and sending time is 0.74 µs on the FPGA side under GST modeling, and the UDP receiving time is 1.5 µs and sending time is 0.74 µs under the GST-N2 interpolation algorithm. The sending and receiving time under both modeling methods does not exceed 5 µs, which leaves sufficient margin for the physical layer transmission of data, thus verifying the reasonableness of sending data 5 µs ahead of time in this paper. Ignoring the initial compilation time for offline simulation and the link time for real-time simulation, the total simulation time for both modeling methods is shown in Table 3. From the table, we can see that the real-time simulation time of FPGA–UREP is much smaller than that of offline simulation, which greatly improves the simulation efficiency.

<table>
<thead>
<tr>
<th>Simulation Methods</th>
<th>Modeling Method</th>
<th>Simulation Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off-line Simulation</td>
<td>GST</td>
<td>41.94</td>
</tr>
<tr>
<td></td>
<td>GST-N2</td>
<td>85.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>116.64</td>
</tr>
<tr>
<td>Real-time Simulation</td>
<td>GST</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>GST-N2</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.5</td>
</tr>
</tbody>
</table>

Table 3. Comparison of simulation time consumption.

In summary, the FPGA–UREP co-simulation system established in this paper has very good real-time simulation characteristics. Compared with the un-interpolated model, the model built by predictive compensation combined with Newton interpolation algorithm greatly improves the simulation accuracy while maintaining a high simulation efficiency and high feasibility.

6. Conclusions

In this paper, we proposed an asynchronous parallel real-time simulation method for doubly fed wind power systems based on FPGA–CPU. The main work and contributions are as follows:

(1) Based on the CDMS, we performed an asynchronous length model segmentation of the doubly fed wind power system, which enables the decoupling of the small-step back-to-back converter from the large-step turbine and the grid.

(2) The timing and error problems in the decoupling calculation are analyzed, and the PWM data of the small-step side are processed using integral averaging. A prediction delay compensation method was proposed for linear prediction and correction of the data on the large-step side, and the error compensation of the small-step data was realized based on this method combined with Newton interpolation.

(3) The GST mode method is used to model the back-to-back converter in FPGA, and the hardware resources are optimized based on the time-sharing multiplexing principle.

(4) Using multiple processors for different simulation tasks, the real-time simulation platform of the doubly fed wind power system based on FPGA–CPU is built. In particular, the HDL development of the back-to-back converter is carried out on the FPGA with a small step of 1 µs, and the 50 µs simulation of the wind turbine and the external grid is completed.
simultaneously in the real-time simulator UREP. By comparing with MATLAB/Simulink offline simulation results, combined with the analysis of errors and comparison of simulation time consumption, the real-time performance and accuracy of the modeling and improved method in this paper are verified.

The proposed improved method solves the latency of multi-processor asynchronous parallel co-simulation and improves the accuracy of co-simulation. Among them, the asynchronous data parallel processing method can be used as a reference for real-time simulation of various rigid systems in power systems. For example, the large inertia link in the power system can be simulated in the CPU, while the high-frequency power electronics are run in the FPGA, and the communication between them can be realized by the interface processing method proposed in this paper. As an important part of the doubly fed wind power system, the back-to-back converter, with its GST model and hardware resource optimization, is unique to doubly fed wind power systems, which is the focus of this paper. In addition, the real-time simulation in this paper can provide a mapping for the stable operation, fault detection, and control methods of the real-world doubly fed wind power system. The real-time simulation can effectively avoid the operation risk of the real system, reduce the system operation and maintenance cost.

Author Contributions: Conceptualization, G.Y. and Z.H.; methodology, G.Y. and Y.L.; software, G.Y. and P.H.; validation, Y.L., Z.H. and Z.C.; formal analysis, Y.L.; investigation, Z.C.; resources, Z.H. and Z.C.; data curation, G.Y. and P.H.; writing—original draft preparation, G.Y. and Y.L.; writing—review and editing, Z.C.; visualization, Z.H.; supervision and project administration, Z.H. and Z.C.; funding acquisition, Z.H., Z.C. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Guizhou Provincial Education Department “Research Center of New Power System and Digital Technology Engineering” (QJJ [2022]043).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
