



Article Modeling of Power System Simulation Based on FRTDS

Bingda Zhang *, Ruizhao Hu, Sijia Tu, Jie Zhang, Xianglong Jin, Yun Guan and Junjie Zhu

The Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China; hu940319@tju.edu.cn (R.H.); tusijia@tju.edu.cn (S.T.); zj_ee@tju.edu.cn (J.Z.); 2017234016@tju.edu.cn (X.J.); gyun0601@tju.edu.cn (Y.G.); jjzhu@tju.edu.cn (J.Z.)

* Correspondence: bdzhang@tju.edu.cn; Tel.: +86-022-2740-4101

Received: 15 September 2018; Accepted: 12 October 2018; Published: 13 October 2018



Abstract: In order to expand the simulation scale of the real-time digital solver based on FPGA (FRTDS, FPGA: field-programmable gate array), the power system modeling process is optimized. The multi-valued parameter method is used to represent the external characteristics of the equipment. The methods of addressing the equivalent admittance and voltage coefficient of the interval unit are discussed in detail. The serial degree of the simulation script is effectively reduced. The disadvantageous effects of asymmetric elements and nonlinear elements on node elimination are analyzed. The elimination order of nodes is determined according to the minimum estimate of the execution time of the simulation script. According to the proposed method to reduce the serial degree and calculation time of the simulation script, software for generating an electromagnetic transient simulation script for power systems is developed. The effectiveness of the software is verified by an example.

Keywords: power system; real-time simulation; multi-valued parameter method; simulation script

1. Introduction

The analysis of power system simulations is the basis of the design and scheduling of power systems. Real-time simulation can be applied in hardware-in-the-loop experiments on secondary equipment and technical training for power system personnel [1].

Currently, devices for real-time digital simulation mainly include real-time digital simulators (RTDS) and real-time labs (RT-LAB), and these devices are all multi-core-based parallel processing systems. For these simulation devices, many scholars have carried out research on a coarse-grained parallel algorithm. In order to accomplish the secondary subnetting of power systems, each sub-network can be divided into multiple "groups" by matrix transformation, based on the natural decoupling of long-distance transmission lines to form multiple sub-networks [2]. The multi-area Thevenin equivalent (MATE) parallel algorithm was proposed so that the large-scale power system could be divided into multiple parallel processing sub-networks, based on the "diakoptics" and modified nodes analysis method [3–5]. A node splitting method, which improved the flexibility of subnetting, was proposed on the foundation of MATE method [6]. An alternating current/direct current split parallel algorithm, which lays the foundation for solving the electromagnetic transient real-time simulation of the high-voltage direct current transmission system, was proposed by means of the node splitting method and long-distance transmission line decoupling method [7].

A graphics processing unit (GPU) with a powerful parallel computing ability has been initially applied in a power system simulation. The node voltage equations could be solved in parallel by invoking the commercial software package of computed unified device architecture (CUDA) so that a simple electromagnetic transient simulation could be performed through the GPU [8,9].

The parallel algorithm based on single-instruction multiple data (SIMD) and a shared memory-based electromagnetic transient arithmetic level was proposed with the application of the GPU [10]. However, it is still very difficult to complete real-time simulations using the GPU, due to the access delay of the internal memory in the GPU and calculation efficiency of CUDA.

The field-programmable gate array (FPGA), which has the characteristics of high parallelism and a deep pipeline, has been applied in the real-time simulation of a power system. The real-time simulation of the active distribution network with a modular structure in FPGA was completed. Several key modules, whose design greatly shortens the calculation time of the simulation but requires researchers to master FPGA programming and complete knowledge of the power system, were introduced briefly [11,12]. The research team from the Key Laboratory of Smart Grid of the Ministry of Education in Tianjin University invented FRTDS to encapsulate commonly used expressions and functions in microprocessor cores and made a compiling software to convert the simulation scripts into orders for controlling the operation of the microprocessor cores [13]. Users only need to write simulation scripts according to the simulation calculation process and do not have to program the FPGA.

The modeling process of the power system simulation of FRTDS is embodied in the simulation script. The quality of the simulation script is an important factor in determining the scale of simulation in the case of a relatively mature hardware structure and the compiling software of FRTDS. Power system simulation script generation software is designed in this paper in order to promote the engineering application of FRTDS. This software sorts the simulation script automatically after receiving parameters for each equipment from users. The serial degree of the simulation script can be decreased greatly when the external characteristics of transformers, transmission lines, and synchronous generators are described through a multi-valued parameter method and interval units are described as comprehensive devices related to multi-valued equivalent admittance and a multi-valued voltage coefficient. A high-quality simulation script generation method is proposed considering the characteristics of the admittance matrix in node voltage equations and the transitivity of asymmetric elements and nonlinear elements in the node elimination process.

The structure of the paper is as follows. The structure of the FRTDS microprocessor cores and the functionality of the compiling software are introduced in Section 2. Section 3 describes the simulation model of each piece of equipment in the power system. Section 4 presents the multi-valued parameter simulation model of interval units. The generation process of the power system simulation scripts is shown in Section 5. The feasibility of software is verified in Section 6. Several pieces of experience in power system simulation modeling with FRTDS are listed in Section 7.

2. FRTDS

2.1. Microprocessor Core

The FRTDS real-time solver consists of multiple microprocessor cores, which exchange data with each other through the data sharing area, as shown in Figure 1. There is a data transmission circuit, comparison operation circuit, and arithmetic operation circuit inside the arithmetic component. One order is extracted by the orders processing unit and is written into the selection controller, the input data controller, and the output data controller after being decomposed at each clock tick. The selection controller and the output data controller determine which data storage unit the input and output ports of the computing circuit are connected to.



Figure 1. Structure of a microprocessor core.

The order is described as "control word + address table" format. The control word includes a selection control word and a data port word. Furthermore, the selection control word describes the state of the selection control ports in the calculation component, and the data port word describes the state of each data input/output port (0 is set as vacant state, and 1 is set as busy state). The address table shows the address of the data storage unit or data sharing area to which the data input/output ports in the busy state will be connected. Information of the same expression will appear in order at different clock ticks.

The parameters with a finite number of values are called multi-valued parameters, and the influencing words are used to describe the factors that affect the numerical change of multi-valued parameters. The influencing words are divided into internal influencing words and external influencing words. The way internal influencing words describe the multi-valued parameters is related to some variables in the simulation process, and the way external influencing words describe the multi-valued parameters is related to the status of some external equipment. The values of multi-valued parameters are stored in a data storage unit according to certain rules. The influencing words need to be stored separately; thus, the guide words of multi-valued parameters are obtained according to base addresses, internal influencing word addresses, external influencing word addresses and the decoding method. In the orders of the microprocessing cores, the addresses of the multi-value parameters are the addresses of their guide words. The orders processing unit obtains the corresponding influencing words according to the address of the internal and external influencing words of the guide word. The complete influencing words are obtained by the orders processing unit through the "or gate" operation. According to the decoding method of the guide words, the influencing words are transformed into offsets, which are added to the base addresses of the guide words to form real addresses of multi-valued parameters.

The decoding circuits in the orders processing unit, part of which can be solidified in the FPGA depending to the actual needs, are designed according to the reconfiguration concept. By the application of multi-valued parameters, the values of some parameters can be gained from tables, which greatly alleviates the burden of calculation on the components and effectively reduces the simulation time.

2.2. Compiling Software

The compiling software transforms the simulation scripts into orders. There are two steps: the directed acyclic graph (DAG) that describes the dependent relation of expressions is generated

based on the simulation scripts, and then the tasks are scheduled according to the DAG, and orders are generated.

The simulation scripts consist of the variable declaration body and expressions. The compiling software specifies 45 kinds of basic arithmetic expressions, which include intermediate variables, circulated variables, initial variables, and output variables. The intermediate variables refer to the temporary data in the calculation process. The circulated variables are the data transmitted from the data in the previous time step. The initial variables represent the data only read and not written in the simulation calculation process. The output variables stand for the simulation results to be output. The variable declaration body is responsible for the declaration of circulated variables, initial variables, and output variables.

Expressions are seen as tasks. If the left operand of task A equals the right operand of task B and task A is ahead of B in sequence in the simulation script, then A is the former-task of B. The DAG showing the dependent relation of tasks is generated according to the sequence of variables and location of expressions after all expressions are scanned by the compiling software. The clock tick, when the address of the input data in the first-level calculation is delivered to the input data controller, is regarded as the scheduled time of the task. Under the assumption of unlimited hardware resources, the DAG is read from top to bottom to calculate the earliest scheduled time of the tasks and the best final time for all tasks to be finished. Then, the latest scheduled time of the tasks can be calculated after DAG is read from bottom to top.

The tasks that can be scheduled are called ready tasks. The ready tasks are arranged according to the latest scheduled time priority strategy, which means that the earlier the latest scheduled time is, the sooner the task will be arranged. There are three issues to consider during the arrangement of a task: (1) whether the calculation components can undertake the task; (2) whether the input data required in the task can be obtained directly; and (3) whether there is conflict between the read ports and write ports in the relevant data storage area. The data need to be adjusted in advance when (2) and (3) are not satisfied. Three data adjustment methods are as follows: (1) the data source is moved to the appropriate data storage area directly; (2) the data are delivered to the input port of the calculation components in advance; (3) the data source is copied and sent to the appropriate data storage area temporarily. The location of data storage is reset without occupying the calculation component resources in method (1). With the utilization of the data latch function of the vacant input ports, data are arranged in advance, which consumes some calculation component resources of the data input ports, may be part of the input task, or may be the whole task in method (2). The data adjustment is accomplished through data transmission circuits, which consumes some resources of the data transmission channel for the calculation component and takes up storage area resources in method (3). The compiling software checks the feasibility of all methods and selects one method according to the resource-saving priority strategy when many methods are all feasible. The adjustment cost of a task being arranged in a certain microprocessor core is defined as follows:

$$\cos t = \sum_{i=1}^{s_2} \left(a_i \times t_i \right) + \sum_{i=1}^{s_3} \left(b_j \times T_j + c \right) \tag{1}$$

where s_2 is the number of adjusted data in method (2), a_i is the unit time cost of occupying data port i, t_i is the time of occupying data port i, s_3 is the number of adjusted data in method (3), b_j is the unit time cost of taking up storage area j (the data sharing area is bigger than the storage area in the core), T_j is the time of taking up storage area j, and c is the cost of occupying the resources of the data transmission channel.

At each clock tick, the task with the earliest scheduled time among all ready tasks is found. Its adjustment cost in all microprocessor cores is calculated. Then, the task is arranged in the microprocessor core with the lowest cost. This is repeated until all ready tasks are arranged or one is unable to arrange the remaining ready tasks. The number of orders generated by the compiling software represents the time consumed for FRTDS to calculate all expressions. This time is closely related to the parallelism degree and serial degree of the simulation script. If the parallelism degree of the expressions is enlarged, the serial degree usually declines. Thus, the best final time decreases, but the calculation amount may increase. Finally, the actual execution time will not be reduced. Therefore, generating a high-quality simulation script is the key to improving the simulation scale, according to the computing power of FRTDS, while taking into account the parallelism degree and calculation amount.

3. Power System Simulation Model

The real-time power system simulation platform constructed in this paper consists of three parts: the industrial control machine, secondary equipment and FRTDS. The industrial control machine completes simulation auxiliary tasks such as fault setting, load setting, misoperation judgment, and power system operation status monitoring. The secondary equipment includes real hardware devices such as exciters, switch cabinets, analog circuit breakers, and analog panels. The industrial control unit and secondary equipment are connected to the FRTDS peripheral interface to complete the hardware-in-the-loop experiment. The simulation operation of the mechanical system and electrical system, which is composed of synchronous generators, transformers, lines and other equipment, is completed in the real-time solver of FRTDS. In order to facilitate the processing of the synchronous generator model, the standard value is used in the simulation process. In the processing of the equipment simulation model, the dynamic element is equivalent to the adjoint circuit by the implicit Euler formula.

3.1. Transformer Model

The transformer simulation model is a controlled source model [14]. The adjoint circuit model of the three-phase double-winding transformer with Y-d1 connection mode is shown in Figure 2.



Figure 2. Electromagnetic transient model of three-phase double-winding transformer Y-d1 connection mode.

The characteristic equation of this adjoint circuit is

$$\mathbf{i}(t) = \mathbf{Y}\mathbf{u}(t) + \mathbf{J}(t - \Delta t)$$
⁽²⁾

where $\mathbf{i} = \begin{bmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 \end{bmatrix}^T$, $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{bmatrix}^T$, \mathbf{Y} is the node admittance matrix, and $J(t - \Delta t)$ is the linear combination of historical current sources.

Because of the saturation characteristics of transformer core, the inductance of the excitation branch is a nonlinear component. The basic magnetization curve of the core material is linearized piecewise. Then, the elements of Y related to G_m , the historical current source coefficients of the excitation branch and the residual current term in the historical current source are equal to different values in different parts of the curve. Through using the multi-valued parameter method to process these variables, only one set of internal influencing words are needed to complete the addressing of the relevant variables of the in-phase excitation branch. In addition, the influencing word can be represented by the state of the segmentation point (when the excitation current is greater than the segmentation point, it is equal to 1, otherwise to 0). In the addressing circuit, the number of influencing words which are equal to 1 is counted to obtain the offset.

3.2. Transmission Line Model

The transmission line simulation model is shown in Figure 3. One line is divided into two sub-lines. Every sub-line adopts the π -type lumped parameter model, where two short-circuit fault models and one disconnection fault model are inserted. The disconnection fault model consists of three two-valued conductances. The short-circuit fault model consists of three two-valued conductances and one grounding conductance. By changing the state of the two-valued conductances and setting the value of the grounding conductance, the model can simulate different types of short-circuit faults. By changing the length of the two sub-lines, it is possible to simulate faults at different locations on the transmission line.



Figure 3. π -type transmission line model with faults.

The values of the elements in the characteristic equation of the π -type lumped parameter model are related to the line length and cannot be processed by the multi-valued parameter method. Therefore, the self-admittance of node 4 in Figure 3 needs to be recalculated at each time step. The self-admittance can be divided into the increments of sub-line I and two fault models. The increment of two fault models is provided by two two-valued conductances, which can be processed by the multi-valued parameter method. To calculate the self-admittance of node 4, it is only necessary to add the increment of sub-line I to the multi-valued parameter. By extracting the increments which can be represented by the multi-valued parameters in the self-admittance, we can do one less addition when solving the self-admittance, which reduces the calculation amount and serial degree of the simulation script.

3.3. Synchronous Generator Model

In the synchronous generator model processing, the excitation voltage u_f and rotational speed ω are assumed to be constant. By connecting the flux linkage equation of the fourth-order model with the basic voltage equation and differentiating it, we can obtain

$$\boldsymbol{u}(t) = \boldsymbol{B}\boldsymbol{i}(t) + \boldsymbol{M}\boldsymbol{i}(t - \Delta t) \tag{3}$$

where $\boldsymbol{u} = \begin{bmatrix} u_d & u_q & u_0 & u_f & 0 & 0 \end{bmatrix}^T$, $\boldsymbol{i} = \begin{bmatrix} i_d & i_q & i_0 & i_f & i_D & i_g & i_Q \end{bmatrix}^T$. Since the *f* winding and the *D* winding in the states are decoupled from the

Since the f winding and the D winding in the stator are decoupled from the g winding and the Q winding in the rotor, lines 3, 4 and lines 5, 6 of Equation (3) are organized as

$$\mathbf{i}_{fD} = \mathbf{Y}_{fD} \cdot \begin{bmatrix} u_f & 0 \end{bmatrix}^{\mathrm{T}} + \mathbf{C}_{fD}\mathbf{i}_d + \mathbf{H}_{fD}\mathbf{i}_{fD}(t - \Delta t)$$
(4)

$$\mathbf{i}_{gQ} = \mathbf{C}_{gQ}\mathbf{i}_{q} + \mathbf{H}_{gQ}\mathbf{i}_{gQ}(t - \Delta t)$$
(5)

Substituting Equations (4) and (5) into (6), we can obtain

$$\dot{\mathbf{i}}_{dq0}(t) = \mathbf{Y}_{dq0} \mathbf{u}_{dq0}(t) + \mathbf{H}_{dq0} \dot{\mathbf{i}}_{dq0}(t - \Delta t)$$
(6)

The elements in the coefficient matrix of Equations (4) and (5), which are only related to the parameters of the generator, can be calculated and stored in advance. The calculation process of Y_{dq0} and H_{dq0} in Equation (6) are related to ω and u_f . Thus, the relevant calculation process must be included in the simulation script.

The electrical angle θ in a static reference frame, which is needed in Park's transformation, can be calculated by the electrical angle δ in the synchronous reference frame.

$$\theta(t) = \omega_N \cdot t + \delta(t) \tag{7}$$

The characteristic equation of the synchronous generator can be obtained by Park's inverse transformation of Equation (6).

$$\mathbf{i}_{abc}(t) = \mathbf{Y}_{abc}\mathbf{u}_{abc}(t) + \mathbf{H}_{abc}\mathbf{i}_{abc}(t - \Delta t)$$
(8)

3.4. Mechanical System

The mechanical system consists of a speed control system, a prime mover, and an equation set of shafting motion. The speed control system uses a power-frequency governor, which takes rotational speed ω and electromagnetic power P_e as input signals. P_e is calculated from the synchronous generator model:

$$P_e = u_d i_d + u_q i_q + 2u_0 i_0 (9)$$

The speed control system outputs the openings to the prime mover model. Taking a large steam turbine generator set as an example, the prime mover consists of four cylinders. The mechanical power of each cylinder is calculated by the prime mover model. Each cylinder rotor and generator rotor are processed into a concentrated rigid mass and an equation set of shafting motion is established. The rotational speed and synchronous electrical angle of the generator can be calculated by the equation set of shafting motion. After differentiating the equation set of shafting motion, we obtain

$$D\omega = T \tag{10}$$

$$\delta_i(t) = (\omega_i - 1)\Delta t + \delta_i(t - \Delta t), i = 1, 2, \dots, 5$$
(11)

where the elements of D are related to the inertia time constant, self-damping coefficient, mutual damping coefficient and torsional spring coefficient of each mass. ω is the rotational speed vector of each mass, and δ_i is the synchronous electrical angle of the mass *i*. ω_5 and δ_5 are the rotational speed and angle of the generator rotor, respectively. The elements of T are related to each cylinder's mechanical power, generator electromagnetic power, historical items of synchronous electrical angle and rotational speed.

D is a constant matrix, whose inverse matrix can be calculated and stored in advance. It can reduce the calculation amount and serial degree of the simulation script.

4. Multi-Valued Parameter Simulation Model of Interval Unit

The interval unit usually consists of circuit breakers, isolating switches and current transformers. Figure 4a shows the interval unit of the incoming lines connected to the double-bus, and Figure 4b is its circuit. G_{2g} , G_{3g} and G_{4g} are the grounding conductances, G_{34a} , G_{34} and G_{34c} are the equivalent conductances of the current transformer, while the rest are two-valued conductances. In order to set the value of the grounding conductance arbitrarily, the network is equivalented by setting nodes 1,5,6,2d,3d and 4d as boundary nodes. There are three independent sub-networks. Each sub-network contains only seven two-valued conductance before equivalence. The equivalent sub-network only has boundary nodes remaining, which reduces the dimension of the network equation and reduces the serial degree when solving the network equation.



Figure 4. (**a**) The interval unit of the incoming lines connected to the double-bus; (**b**) the circuit of the interval unit.

The equivalent admittance between the boundary nodes can be pre-stored as a multi-valued parameter. In order to ensure the network topology connectivity, the two-valued conductances' value is usually specified as $10^{-8}/10^8$ S. Thus, the multi-valued parameter's value is up to 128. It is stipulated that when the internal node cannot reach any boundary node, its node voltage is zero, and all the two-valued conductances in Figure 4b are specified as $0/10^8$ S. Therefore, the value of the multi-valued parameter will be greatly reduced [15].

Because the sub-network is a purely resistive network, the internal equivalent is achieved by performing a star-angle transformation with the internal node being the central node one by one. If the path from the internal node to the equivalent admittance related node is turned on (each two-valued conductance on the path is equal to 10^8 S), the star-angle transformation affects the value of the equivalent admittance, which is determined by the value of the boundary conductance (the two-valued conductance between the internal node and the connected boundary node). Figure 5 shows the circuit of the A-phase sub-network before the internal equivalent. The equivalent admittance $Y_{1a,2d}$ is taken as an example to illustrate the processing of the multi-valued parameter method. The paths 1a to 2d are numbered as (1), and the paths of the boundary nodes 3d, 4d, 5a, 6a to (1) (2a) are respectively numbered as (2), (3), (4) and (5). The on/off status of path (1) determines whether $Y_{1a,2d}$ is zero or not. The remaining paths represent the effect of boundary conductance on the equivalent conductance in

the star-angle transformation. The paths of star-angle transformation in the same time are classed into one group, then (2) is classed into one group and (3), (4), and (5) are classed into another group. Conducted paths in the various groups are ordered according to their number and then combined to create a sequence. According to this sequence, the specific values of $Y_{1a,2d}$ in each case are calculated and stored as multi-valued parameters in advance. $Y_{1a,2d}$ has nine kinds of values. It is only necessary to count the number of conducted paths in each group to calculate the offset of $Y_{1a,2d}$. The decoding circuit is shown in Figure 6.



Figure 5. The circuit of the A-phase sub-network.



Figure 6. The decoding circuit of $Y_{1a,2d}$.

All equivalent admittances can be addressed by the above method, and a general addressing formula is listed as follows:

$$D = D_0 + s_1 \times \sum_{i=2}^{n} \left[(s_i + 1) \times \prod_{j=i+1}^{n} (m_j + 1) \right]$$
(12)

where *D* is the actual address of the multi-valued parameter, D_0 is the base address, *n* is the number of groups after the path classification, s_i is the number of conducted paths in the *i*-th group, and m_j is the total number of the paths in the *j*-th group. When dealing with equivalent self-admittance addressing, there is only one self-admissive correlation node in path (1); besides this, s_1 is always equal to 1.

In order to calculate the current in the current transformer and judge whether the current in the circuit breaker is crossing zero, the internal node voltage needs to be calculated. The internal node voltage is equal to the sum of the product of each boundary node voltage and the corresponding voltage coefficient. The voltage coefficient is equal to the voltage value of the relevant internal node when the relevant boundary node is connected to the ideal voltage source and the other boundary nodes are grounded. The voltage coefficient can also be pre-stored by using the multi-valued parameter method. The on–off status of the grounded boundary node to the relevant boundary nodes' path

indicate whether the path is shunted, and so the addressing process of the voltage coefficient can be handled by a similar method of processing the equivalent admittance.

By utilizing the symmetry of the incoming interval circuit topology, the same decoding circuit can be applied in different multi-valued parameter addressing processes. In addition, only the position of each two-valued conductance in the influencing word needs to be adjusted. The addressing process of all multi-valued parameters in the double-bus incoming interval model can be accomplished with only 13 kinds of decoding circuits.

5. Power System Simulation Script Generation Process

5.1. Power System Simulation Script Framework

To ensure that the power system network equation is linear, the mechanical system is decoupled from the electrical system. An iterative solution is used to solve the mechanical system and the state of the nonlinear components in the electrical system. In order to ensure the real-time performance of the simulation, the network equation is only iteratively solved once.

The power system simulation script framework is shown in Figure 7. It is mainly divided into six parts: (1) the variable declaration body; (2) expressions for forming the characteristic equations of each piece of equipment; (3) expressions for forming the network equation, which is composed of the expressions for forming the node admittance matrix and the injection current vector; (4) expressions for solving the network equation; (5) expressions for calculating the internal electrical quantities of each equipment model; (6) the expressions for updating the output variables, which are calculated by the node voltage and converted into the actual values to output.



Figure 7. The decoding circuit of $Y_{1a,2d}$.

In parts (1), (2), (5) and (6) of the power system simulation script framework, the scripts of the same equipment are in the same form. Thus, the simulation model of the same device can be packaged into a class, which is added to the equipment library. The class includes the calculation process of converting the original parameters of the equipment into simulation parameters and simulation script templates (the declaration body template of simulation parameters, expression templates for forming the characteristic equation, expression templates for solving internal electrical quantities, and expression templates for updating output variables). By converting the original parameters into simulation parameters and declaring them in advance, the serial degree and calculation amount of the simulation script can be reduced.

The user only needs to provide information about type, number, original parameter, and associated node of each equipment in the network. And the object of the corresponding class can be generated. According to the equipment number and associated nodes, the simulation script templates can be filled in. And the "sub-script" of each equipment is filled into the corresponding part of the simulation script framework. At the same time, a node-equipment correspondence table and an adjacency matrix describing the network topology relationship are generated. The expressions in part (3) of the simulation script framework can be generated according to the node-equipment correspondence table. The adjacency matrix prepares for the subsequent solution of the network equation. Thus, the expressions of part (1), (2), (3), (5) and (6) of the power system simulation script framework have all been generated.

5.2. Expression Generation Technique for Solving the Network Equation

We use the Gaussian elimination method to solve the network equation. We update the expressions of the admittance matrix and injection current vector, and back-substitution expressions are generated in the process of eliminating nodes one by one in the network.

Using the sparsity of the node admittance matrix can reduce the computational burden of the expressions. When the node is being eliminated, all connected nodes of the elimination node are found according to the adjacency matrix. The updated expressions of the mutual admittance between the connected nodes, the self-admittance and injection current of the connected nodes are generated. In addition, the update expressions of the node voltage in the back-substitution process are generated. Then, the adjacency matrix is updated to connect the connected nodes.

The symmetric mutual admittance only needs to be stored once and updated once during the node elimination process. Using the symmetry of the node admittance matrix can omit many expressions. The mutual admittance between the nodes of the synchronous generator is asymmetrical, and the elimination of the node will transmit the asymmetry. The range of the asymmetry is recorded during the node elimination process, so that the symmetry of the mutual admittance can still be used to generate the expression outside the range.

Both the updated expressions and the back-substitution expressions need one-step division. In the computational component of FRTDS, the pipeline of the division operation is much longer than the pipeline of the addition and multiplication operations. Before generating the updated expressions, the reciprocal of the self-admittance of the elimination node is calculated and stored. Thus, the division operation in the subsequent expressions can be converted into a multiplication operation, which shortens the serial degree of the network equation solving process. The expression of calculating the reciprocal of self-admittance of the elimination node is

$$D_{ii} = 1/Y_{ii} \tag{13}$$

where *i* is the number of the node to be eliminated.

The update expression of the node admittance is

$$Y_{jk} = Y_{jk} - Y_{ji} \times Y_{ik} \times D_{ii}$$
⁽¹⁴⁾

where *j* and *k* are the numbers of the node connected to *i*. If j = k, Expression (14) is the updated expression of the self-admittance. If $j \neq k$, Expression (15) is the updated expression of mutual admittance.

The updated expression of the node injection current is

$$I_j = I_j - Y_{ji} \times I_i \times D_{ii} \tag{15}$$

The back-substitution expression is

$$U_i = (I_i - \sum_{j \in V_i} U_j \times Y_{ij}) \times D_{ii}$$
(16)

In the process of node elimination, due to the influence of nonlinear factors, the values of some elements in the network equation when they are solved for the first time are different from those in the iterative solution process. Only these elements need to be recalculated in the iterative solution process, which can avoid resolving the whole network equation and reduce the calculation amount. By recording the association range of the nonlinear factors, the expressions in the iterative solution process can be generated while in the first solution process.

The associated nodes of the nonlinear components are marked as nonlinear nodes, and the remaining nodes are marked as linear nodes. When a nonlinear node is eliminated, the connected linear node is marked as an affected node. When the affected node is eliminated, the nonlinear factor is also transmitted to the linear node, which will be marked. In the process of solving the network equation for the first time, only the updated expressions generated when the nonlinear nodes and the affected nodes are eliminated need to be added to the iterative solution process.

In order to inherit the values of the variables in updated expressions between the two solving processes, the variable names in expressions (14)–(16) need to be modified. We change the variable name Y_{ij} , representing a node admittance to Y_{t_ij} , and change the variable name I_i , representing an injection current to I_{t_i} , where the subscript *t* represents the number of times that value is changed in the process of node elimination.

When the original nonlinear elements (the admittance and node injection current directly related to nonlinear components) are affected by nonlinear factors, in order to introduce nonlinear values, it is necessary to add expressions of extracting and introducing element increments to the simulation script. We add expressions of extracting the element increments in the first solution process as follows:

$$Cum_Y_{t_ij} = Y_{t_ij} - Y_{t_0_ij}$$
⁽¹⁷⁾

$$Cum_{I_{t}i} = I_{ti} - I_{t_{0}i}$$
(18)

where t_0 is the number of times that the element was last changed by nonlinear factors. If the element is first affected by a nonlinear factor, then $t_0 = 0$, which represents the value of the element in the original network equation.

We add expressions of introducing the element increments in the iterative solution process as follows:

$$Y_{t_{ij}} = Y_{t_{0}ij} + Cum_{Y_{t_{ij}}}$$
(19)

$$I_{t_{-i}} = I_{t_{0-i}} + Cum_{-}I_{t_{-i}}$$
(20)

Nonlinear factors can be divided into two categories: the first is the change of the state of the nonlinear component, and the second is the transfer of the nonlinear value when the nonlinear element is updated. When the original nonlinear element is affected by the first type of nonlinear factors, the expressions of extracting element increments are added to the simulation script before this node is eliminated; when it is affected by the second type of factor, this means that a new nonlinear value

is introduced in the other node elimination process, and expressions of introducing the element increments are added before the updated expression of this element.

By adding the expressions of processing increments when the original nonlinear element is affected by the nonlinear factors, only two additive expressions are used to replace the updated expressions of the element when continuous linear nodes are eliminated, which further reduces the serial degree and calculation amount of the iterative solution process.

5.3. Optimal Elimination Order of Nodes

The elimination order of nodes affects the efficiency of solving the network equation. This is mainly reflected in three aspects: (1) the number of new-born elements in the node elimination process is influenced by the elimination order of nodes; (2) nodes that are not connected to each other can be eliminated in parallel, and the elimination order of nodes affects the parallel elimination; and (3) the elimination order of synchronous generator nodes and nonlinear nodes affects the association range of asymmetric elements and nonlinear elements. The calculation amount of the simulation script in the network equation solving stage is affected by (1) and (3), and the parallelism degree is affected by (2). Therefore, finding the optimal elimination order of nodes is the key means to improving the quality of the simulation script.

The optimal node elimination order problem is a kind of NP-hard combinatorial optimization problem. It is similar to the traveling salesman problem (TSP). The feasible solution to the problem is an arrangement of all node numbers, and the neighborhood information of the feasible solution affects the objective function value.

The arrangement of nodes in a feasible solution is used as an elimination order to generate a simulation script. If the execution time of the simulation script is directly used as the optimization target, the solution speed of the optimal node elimination order problem will be slow due to the long execution time of the compiling software. When the compiler software is scheduling tasks, 80% of the time will be spent on data adjustment due to issues (2) and (3). If we ignore the read/write conflict problem of the data storage area (issue (3)), and only use the method of transferring data between microprocessor cores to solve the problem of whether the input data can be obtained (issue (2)), then the cost calculation only needs to consider the number of data transfers between cores and the solution speed is improved. In this way, the calculated time is an estimate of the actual execution time by considering the parallel computing capability of the computing components, which is slightly less than the actual execution time. Using the estimation of the simulation script execution time as the objective function value will improve the solution speed of the optimal node elimination order problem.

TSP is a classic combinatorial optimization problem. Many scholars focus on using heuristic algorithms to solve this problem. The genetic algorithm (GA) has strong global search ability and performs well in solving TSP. The crossover operator and mutation operator play a decisive role in the performance of GA. The existing crossover operators and mutation operators are comprehensively summarized, in which the order crossover (OX) and the displacement mutation (DM) perform best in solving TSP [16]. Therefore, the GA of OX and DM is used to solve the optimal node elimination order problem.

6. Case Study

The Xilinx Virtex-7 VC709 FPGA (Xilinx Company, San Jose, CA, USA) is adopted to build a real-time solver for FRTDS. The real-time solver consists of eight microprocessor cores whose clock frequency is 200 MHz. The power system simulation script generation software is compiled on the QT C++ platform, according to the generation process of the power system simulation script. On the real-time simulation platform for the power system, the performance of the simulation script is verified by the Institute of Electrical and Electronics Engineers (IEEE) 14-bus system with five synchronous generators, shown in Figure 8. In the simulation model of the IEEE 14-bus system, the interval unit is connected to both ends of each line. The GEC-300 exciter (Jisi Electric Co. Ltd, Beijing, China) is

used to provide the excitation signal of G1. The excitation systems of other generators use the same mathematical model as the exciter, whose calculations are completed in the real-time solver.



Figure 8. Institute of Electrical and Electronics Engineers (IEEE) 14-bus system.

In order to verify the effectiveness of the internal equivalent method of the interval unit, two kinds of interval unit models are set in the equipment library: one is the model adopting the internal equivalent method, and the other is the original model. Table 1 shows the comparison of the number of simulation nodes and the multi-valued parameter storage and simulation time before and after using the internal equivalent method. As we can see in Table 1, the additional data storage is very small and the simulation efficiency can be significantly improved after applying the internal equivalent method.

Method	Simulation Node	Multi-valued Parameter Storage (kb)	Simulation Time (µs)
Before	308	794	44.865
After	128	1135	38.658

Table 1. Comparison of using the internal equivalent method.

In the iterative solution process of the network equation, two generation methods of expressions are set. Method 1 iteratively updates some of the elements by recoding the association range of nonlinear factors. Method 2 re-solves the overall network equation. Table 2 shows the comparison of the calculation amount and simulation time of the simulation script generated by using the two methods. As we can see, the calculation amount of the iterative solution process is substantially reduced and the simulation efficiency is improved by recording the association range of nonlinear factors.

 Table 2. Comparison of two generation methods of expressions.

Method	Calculation Amount	Simulation Time (µs)
1	28,589	38.658
2	39,989	54.893

In the power system simulation script generation software, three methods to determine the elimination order of nodes are set. Method 1 is the minimum degree algorithm, which considers the least calculation amount, method 2 is the maximum independent set algorithm, which considers the maximum parallelism degree, and method 3 is the genetic algorithm. Table 3 shows the comparison of the simulation times for the three methods. It can be seen that the elimination order of nodes has a great influence on the simulation efficiency. Using the GA to determine the elimination order of nodes can effectively expand the simulation scale.

Method	Simulation Time (µs)	
1	51.968	
2	47.968	
3	38.658	

Table 3. Comparison of simulation time for three methods.

In order to verify the correctness of the simulation script, the simulation model with the same parameters is built on a power system computer aided design (PSCAD). In the operation mode that all of the buses are put into, a three-phase short-circuit fault command occurs at bus I at 10 s through the industrial control machine, and the fault is cleared after 0.1 s. Figure 9 shows the waveform of G1 terminal current, the voltage of bus 1 and the G1 speed, respectively, on FRTDS and PSCAD.



Figure 9. Cont.





Figure 9. (a) Three-phase terminal current of G1; (b) voltage of bus 1; (c) G1 speed.

It can be seen from Figure 9 that the amplitude of the G1 terminal current increases sharply after the three-phase short-circuit fault occurs at 10 s. The voltage of bus 1 drops rapidly, and the exciter starts a strong excitation regulation to prevent the terminal voltage from dropping rapidly. As the electromagnetic power of the synchronous generator drops suddenly, the speed rises and oscillates under the regulation of the speed control system. After the fault is removed, the voltage of bus 1 is quickly restored to the rated value under the strong excitation regulation and the speed is gradually stabilized at the rated speed. The error between the waveforms of the FRTDS simulation and PSCAD simulation is less than 3%; thus, the simulation precision is high.

7. Conclusions

- According to the characteristics of FRTDS, the modeling process of the power system simulation is optimized after considering the serial degree and calculation amount of the simulation script, which effectively improves the simulation efficiency and expands the simulation scale;
- (2) The multi-valued parameter method is used to represent some coefficients in the external characteristic equations of each device, which effectively reduces the burden on the computing component. The internal equivalent method is used for the interval units. The path method is used to analyze the star-angle transformation process of the interval element model. A general addressing method of multi-value equivalent admittance and the multi-valued voltage coefficient is designed to greatly reduce the dimension of the network equation. The serial degree of the simulation script is effectively reduced;
- (3) In the solution stage of the network equation, the correlation range of the asymmetric elements is analyzed. Thus, the expressions can still be generated outside the range by using the symmetry of the admittance matrix. The transitivity of nonlinear elements in the process of node elimination is analyzed. The expressions of processing the increments of the original nonlinear elements are added, which reduces the computation amount of the iterative solution process;
- (4) The elimination order of nodes is determined according to the shortest execution time of the simulation script, which effectively improves the quality of the simulation script. The optimal node elimination order problem is equivalent to the combination optimization problem and is solved by the genetic algorithm. The estimation of the simulation script execution time is proposed as the objective function value, which improves the solution speed;

(5) The power system simulation script generation software is developed based on the proposed method of reducing the serial degree and calculation amount. Users only need to provide relevant information of each piece of equipment to complete the modeling, which further improves the utilization efficiency of FRTDS and lays the foundation for graphical modeling.

Author Contributions: Conceptualization, B.Z.; methodology, B.Z. and R.H.; software, R.H.; validation, R.H., J.Z. (Jie Zhang); formal analysis, B.Z.; investigation, R.H. and J.Z. (Junjie Zhu); resources, B.Z.; data curation, Y.G. and X.J.; writing—original draft preparation, R.H.; writing—review and editing, B.Z. and S.T.; visualization, S.T.; supervision, B.Z.; project administration, B.Z.; funding acquisition, B.Z.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51477114).

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

References

- Tian, F.; Li, Y.; Zhou, X.; Wu, Z.; Zhang, X. Advanced Digital Power System Simulator. *Power Syst. Technol.* 2008, 32, 17–22. [CrossRef]
- 2. Luo, G. Research Development of Parallel Algorithm for Power System Transient Real-Time Simulation. *Guangdong Electr. Power* **2014**, *12*, 62–69. [CrossRef]
- Marti, J.; Linares, L.; Calvino, J.; Dommel, H.; Lin, J. OVNI: An Object Approach to Real-Time Power System Simulators. In Proceedings of the International Conference on Power System Technology, Beijing, China, 18–21 August 1998.
- 4. Marti, J.; Linares, L.; Hollman, F. OVNI: Integrated Software/Hardware Solution for Real-Time Simulation of Large Power Systems. In Proceedings of the 14th PSCC, Sevilla, Spain, 24–28 June 2002.
- 5. Ho, C.W.; Ruehli, A.; Brennan, P. The Modified Nodal Approach to Network Analysis. *IEEE Trans. Circuits Syst.* **1975**, 22, 504–509. [CrossRef]
- 6. Yue, C.; Zhou, X.; Li, R. Study of Parallel Approaches to Power System Electromagnetic Transient Real-Time Simulation. *Proc. CSEE* **2004**, *24*, 5–11. [CrossRef]
- 7. Tian, F.; Zhou, X. Partition and Parallel Method for Digital Electromagnetic Transient Simulation of AC/DC Power System. *Proc. CSEE* 2011, *31*, 1–7. [CrossRef]
- 8. Chen, L.; Chen, Y.; Xu, Y.; Mei, S. Feasibility Study of GPU Based Electromagnetic Transient Simulation. *Power Syst. Prot. Control* **2013**, *41*, 107–112. [CrossRef]
- Jayanta, K.; Fung, W.K.; Aniruddha, M.; Shaahin, F. Simulation of Large-scale Electrical Power Networks on Graphics Processing Units. In Proceedings of the IEEE Electrical Power and Energy Conference, Winnipeg, MB, Canada, 3–5 October 2011.
- 10. Xu, Y.; Chen, Y.; Chen, L.; Ren, Z. Fast Electromagnetic Transient Simulation Method for PWM Converters Based on Averaging Theory. *Autom. Electr. Power Syst.* **2013**, *37*, 58–64. [CrossRef]
- 11. Matar, M.; Iravani, R. Massively Parallel Implementation of AC Machine Models for FPGA-Based Real-Time Simulation of Electromagnetic Transients. *IEEE Trans. Power Deliv.* **2011**, *26*, 830–840. [CrossRef]
- 12. Liu, J.; Dinavahi, V. A Real-Time Nonlinear Hysteretic Power Transformer Transient Model on FPGA. *IEEE Trans. Ind. Electr.* **2014**, *61*, 3587–3597. [CrossRef]
- 13. Zhang, B.; Fu, S.; Jin, Z.; Hu, R. A Novel FPGA-Based Real-Time Simulator for Micro-Grids. *Energies* **2017**, *10*, 1239. [CrossRef]
- 14. Zhu, Y. The Study on Electromagnetic Simulation Model and Algorithm of Power Transformers. Master's Thesis, Tianjin University, Tianjin, China, December 2011.
- 15. Zhang, B.; Wu, Y.; Jin, Z.; Wang, Y. A real-time digital solver for smart substation based on orders. *Energies* **2017**, *10*, 1795. [CrossRef]
- 16. Larranage, P.; Kuijpers, C.; Murga, R.; Inza, I.; Dizdarevic, S. Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artif. Intell. Rev.* **1999**, *13*, 129–170. [CrossRef]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).