Parallel Simulation of The Electromechanical Transient Phenomena of Power System via Modern Software Development Technique

Yu XIE¹, Zhigang WU ¹*, Erzhuan ZHOU ²

(¹South China University of Technology, Guangzhou 510650, China; ²China Electric Power Research Institute, Beijing 100192, China

*Corresponding author: epzgwu@scut.edu.cn)

Abstract: The simulation of power system electromechanical transient phenomena is highly dependent on the knowledge and model of special domain. For this reason, traditional parallelization of the simulation must be coupled with concrete algorithm and code. In this paper, a parallelized electromechanical transient simulation program is proposed based on the open-source simulation engine called InterPSS and general open object-oriented modelling framework. The correctness and effectiveness of this framework have been verified by a large-scale simulation case.

1. Introduction

To construct a new power system with high penetrance of renewable energy, numerical simulation techniques of the system with sufficient accuracy and efficiency are required urgently, and hence, appropriate parallel computing should be implemented via modern high-performance computing techniques. The parallel computing of electromechanical transient (EMT) simulation is one of the important types of simulation tasks, which always requires specific domain knowledge.

There are three types of parallel algorithm used in EMT simulation: spatial parallel algorithms, temporal parallel algorithms, and spatial-temporal parallel algorithms. The main idea of spatial parallel algorithms is to divide the network into different partitions and define sub-tasks for each partition to simulate the corresponding part of EMT model decoupling within a single time step, finally coordinate the sub-results to get the whole result of this time step [1-3].

Temporal parallel algorithm, on the contrary, simulate the entire electrical power network without any partitioning, but try to calculate simultaneously for different time steps to speed up the calculation. Most of the proposed algorithms try to finish the job with the convergence property similar with Newton method, and sometime special task-scheduling methods such as pipeline processing technique is utilized to solve the variable coupling relationship between adjacent integration time steps [4-9].

Both spatial parallel algorithm and time parallel algorithm have limitations. If the spatial parallel algorithm uses too many subnets, the communication burden between subnets increases, which reduces the computational efficiency. The performance of the time parallel algorithm highly depends on the serial convergence of the algorithm, and its speedup tends to be saturated with the increase of the parallel scale. To further improve the technical efficiency of simulation, scholars propose spatial-temporal parallel algorithms aiming to combine the advantages of both types of algorithms. It often solves the
dynamic process of power system in parallel at multiple time points according to the differential nonlinear algebraic equations. The key point is to effectively solve the contradiction between the degree of parallelism and the convergence of parallel algorithms \cite{10}. A space-time parallel relaxed Newton method for transient stability calculation is proposed in \cite{11}, which can achieve a high degree of parallelism and maintain the convergence of Newton algorithms.

The above algorithms and parallelization techniques are based on the specific characteristics of power system simulation problems. It should be pointed out that such algorithm improvement based on specific domain knowledge is an essential improvement of the algorithm studied. It is an improved way to get the best performance-to-price ratio. However, this will inevitably encounter a new problem. At present, the mainstream EMT simulation parallel programs are highly dependent on professional knowledge and models in this field, and even rely on special hardware resources. It is not enough in the application of more general-purpose high-performance computing and advanced software development. To a certain extent, it restricts the expansion and upgrade of the simulation program.

In this paper, InterPSS (Internet based Power System Simulator), a power system simulation engine based on modern software engineering technology, is used to propose a general method of parallelizing EMT simulation. Firstly, with the help of InterPSS's highly open and extensible framework, the author can integrate the MATE (Multi-Area Thévenin Equivalents) algorithm that have been contributed by other key developers in the InterPSS development community. Secondly, a label propagation algorithm is proposed to achieve more effective partitioning for specific computing resources. Thirdly, the parallelization of decoupling calculation of each partition is realized by multithreading in JAVA environment, and it is coordinated effectively without a lot of deployment tasks on very general hardware platform. Finally, a large-scale EMT simulation case is used to verify the correctness and effectiveness of the proposed EMT simulation parallel framework.

2. EMT simulation engine of InterPSS

2.1. Framework of InterPSS

InterPSS is the most mature and open simulation engine in the power industry \cite{12}. Its most outstanding feature is the utilization of advanced software engineering technology, distributed development mode and modular architecture. It is one of the main research directions of IEEE open-source power system simulation software development team. The system architecture of InterPSS is shown in Fig. 1.

As can be seen from Fig. 1, the core of the whole structure of InterPSS is an object-oriented power system simulation model. It is a software system that is flexible, extensible, and easy to maintain. Its most important feature is that the source code is open, and the composition is loose. According to their own requirement, one can add additional function or replace exist function in the form of plug-in. This creates favorable conditions for this paper to expand the existing functions of InterPSS and develop parallel programs for EMT simulation.

![Fig. 1 Framework of InterPSS](image)

2.2. Object-oriented Model of EMT Simulation in InterPSS

As all other EMT simulation programs, such simulation function provided by InterPSS also deals with the power system model formed by connecting the power network model to the generator and other dynamic models. However, unlike most programs, InterPSS follows the standard specification of object-
oriented modeling and development, which is the fundamental guarantee of its openness and high scalability.

The basic network model and its extension are related to EMT simulation concerned in this paper. The basic network model of InterPSS is shown in Fig. 2, in which one can find that the model includes node sets and branch sets that must be included in all power system computing tasks. The branch sets also undertake the task of defining topological connections. In addition, the per unit value is often used in power system analysis and calculation, and it is usually necessary to specify a unified base capacity of the whole network. This is defined in InterPSS by the properties of the network.

There are three major calculations in power system: power flow calculation, short circuit calculation and stability calculation\(^{[13]}\). Logically, all these calculations need to establish the corresponding node admittance matrix when dealing with the network model, which is the task of the BaseAclfNetwork object in Fig. 3. Based on BaseAclfNetwork, the conventional power flow calculation network model (AclfNetwork) and the basic short circuit calculation network model network model (BaseAcscNetwork) can be established respectively. BaseAcscNetwork provides methods for generating sequence networks and can also be used to establish conventional short circuit calculation network model (AcscNetwork) and conventional transient simulation network model (DStabilityNetwork). In this paper, the parallelization of EMT simulation is essentially based on DStabilityNetwork.

Finally, in InterPSS, everything is object-oriented, including all kinds of simulation algorithms. Taking the EMT simulation concerned in this paper as an example, the algorithm itself is also an object. Disturbance events, numerical integration algorithms and simulation parameters are all specific and replaceable attributes or functions of EMT simulation algorithm objects. Only in this way can we parallelize the necessary functions without changing the overall architecture of the program as much as
possible, to achieve the high degree of versatility and expansibility mentioned above.

3. Realization of parallelization of EMT simulation in InterPSS

The general computing framework of EMT simulation is shown in Fig. 4. As can be seen from the figure, all dynamic models are coupled and influenced each other through the network model. To realize the parallelization of computing tasks, this paper considers that the complete computing tasks will be partitioned by an effective networking algorithm. In each integration step, the computing task is published to different computing resources for parallel calculation, and then the results of each sub-network are reasonably coordinated to obtain the result of this integration step.

To achieve the above purpose, two key problems need to be solved: (1) decoupling of the calculation and coordination of different partitions; (2) flexible and efficient partitioning algorithm. In this section, we will make an in-depth analysis to solve the two problems.

![Fig. 4 General Framework for EMT Simulation](image)

3.1 Decoupling and coordination of EMT simulation based on MATE

A method of regional decoupling calculation of large-scale power grid by MATE algorithm is introduced in [14]. Firstly, the original power grid is divided into several areas. The node set is divided into the union of several non-intersection subsets. The branches are divided into two types, one is the branch whose two ends are in the same node subset, and the other is the branch whose two ends belong to two different node subsets. The specific method of network partitioning is not discussed in the original literature, and reasonable results can be read by default. Secondly, if the current through the tie line is not considered for each subnet, the subnet can be solved independently. Thirdly, the corresponding impedance matrix, branch current vector and branch voltage drop vector are defined for the branches connecting different subnets. These quantities are used to coordinate the results of different subnets in an appropriate way to get the whole calculation results of the whole network.

Based on the MATE algorithm, a transient simulation method including simulation objects of both transmission network and distribution network is proposed in [15]. Transmission network and distribution network interact with each other via MATE. The main ideas of the partition decoupling calculation and the coordination of partition results in the parallelization of EMT simulation realized in this paper come directly from [15]. The improvement of this paper is mainly reflected in the ability to achieve a more flexible and efficient network according to the actual situation, and then truly realize the parallelization of simulation calculation, which is not realized yet in [15].

3.2 An improved label propagation algorithm appropriate for flexible partition requirements

The influence of partition results on decoupling calculation and parallel calculation is not discussed in reference [14] [15]. In fact, as a traditional problem of complex network theory, a very important starting point of Network Partitioning problem is that it can flexibly and efficiently obtain the partitioning scheme of large-scale computing task network.

The basic indicators used to evaluate the effect of parallelization of computing tasks are as follows: (1) the size of the subnet should be proportional to the computing power of a single computing resource. The purpose is to let the computing tasks allocated to all computing resources be completed as synchronously as possible, so as to avoid the mutual waiting time when the calculation results are
summarized and coordinated. (2) the interaction between subnets should be as little as possible. This kind of interaction means information exchange between different computing resources in the process of parallel computing, the time required for such interaction should be reduced as much as possible. There are several mature algorithms to achieve this task, including Kernighan-Lin algorithm\cite{10}, spectral partitioning algorithm\cite{11} and heuristic search algorithm\cite{12}.

The main goal of this paper is to provide a “universal” parallelization program framework on the premise of satisfying the best possible parallelization effect. Therefore, the networking method is also required to be universal and can adapt to a variety of hardware computing resources. For this reason, the label propagation algorithm\cite{13} is improved in this paper according to the actual needs for “universal” purpose. The general process of the classic label propagation algorithm is as follows:

1. Each node gets a label after initialization, which can be the number or name of the node. Or only some of the known feature nodes are labeled, and the rest of the nodes are empty labels.
2. Set the propagation probability for the label. If no specific propagation probability defined, the propagation probability can be set to 1 by default.
3. Update the labels for the nodes one by one. The label with the highest propagation probability in the label of the neighbor node will be selected as the new label for that node. When the label propagation probability is set to 1, it is equivalent to selecting the labels that appear most in the neighbor nodes.
4. After updating all node labels, compare them with the previous round of labels. If the labels are different, iteration should be continued until the convergence condition is reached. If the labels are the same, the iteration is completed, and the nodes with the same labels are the same community.

However, the above classical label propagation algorithm has a disadvantage. It cannot specify the number of partitions, which greatly restricts its applicability. In addition, to achieve the goal of ”general ” parallelization program framework, it is also necessary to be able to set the partition proportion manually and have sufficient partition speed. Therefore, we have improved the classical label propagation algorithm as the follow way.

1. The number of nodes labeled at initialization is equal to the number of partitions specified, and the remaining nodes have no labels. Labels are propagated to all nodes through the label propagation mechanism. So that the goal of specifying the number of partitions can be achieved.
2. Nodes are weighted according to the model complexity of the electrical components corresponding to them. The partition size is defined by the sum of the weights of the nodes in the partition instead of the number of nodes, which is closer to the goal of completing the calculation of different computing resources at the same time. Multiplication times are commonly used in program development to approximate model complexity.
3. Edges are weighted according to the amount of transmission information they bear. When updating a node, the edge weights of neighboring nodes that belong to the same label are summed up. The largest label of the sum of edge weights is the new label of the node. This makes it easier for neighbor nodes with high edge weights to pass their own labels to target nodes. So that the edge weights between partitions can be minimized and the information exchange between partitions can be reduced when partitioning the network.
4. Based on the idea of node label propagation capacity in \cite{14}, when the total computing task quantity and the scale ratio of each partition are determined, the expected computing task quantity of each partition can be determined. Label propagation capacity in this paper is defined as an increasing function of the difference between the actual scale of the node and the expected computing task. As the actual size of a partition approaches the expected size, the possibility of nodes in that partition being moved to other partitions decreases. Finally, the purpose of controlling the proportion of partition scale is achieved.

It should be noted that the partition problem is in fact discrete, so it is usually unrealistic that the partition size is exactly equal to the expected size. From the point of view of engineering application, the improved label propagation algorithm introduced here can meet the requirements of the parallelization of EMT simulation.

Take the power network model with 10000 nodes and 12217 edges\cite{15} in matPower software package
as an example. We tested a variety of partition requirements and got the following results.

Test 1: Assume that there are five partitions in which the nodes and edges are not weighted, and the expected partition size ratio is 1:1:1:1:5. The partition results are shown in Table 1.

| Tab.1 Results for Test 1 |
|--------------------------|
| Expected partition ratio (%) | 11.11:11.11:11.11:11.11:55.56 |
| Actual partition ratio (%)  | 10.87:10.84:10.65:11.63:56.01 |
| Maximum partition error     | 1.94%                        |
| Interaction edge weight ratio | 13.88%                   |
| Time (s)                   | 9.44021                     |

Test 2: Test the situation where the computing power of computing resources varies greatly. Assume that there are 18 partitions in which the nodes and edges are not weighted, and the expected partition size ratio is 1:1:1:…:1:5:10:20. The partition results are shown in Table 2.

| Tab.2 Results for Test 2 |
|--------------------------|
| Expected partition ratio (%) | 2:2:2:2:2:2:2:2:2:2:2:2:2:2:10:20:40 |
| Actual partition ratio (%)  | 1.91:1.89:1.98:2.23:2.08:2.11:1.94:1.99:1.99:2.14:2.02:2.12:1.95:1.85:10.42:19.75:39.63 |
| Maximum partition error     | 2.24%                        |
| Interaction edge weight ratio | 17.39%                   |
| Time (s)                   | 35.46993                     |

From these results, even for the ten thousand scale power grid model, the proposed network partitioning method can get acceptable results in a reasonable time. Furthermore, it has a higher adaptability to the requirements of arbitrary weight distribution and arbitrary sub-network scale. Since this kind of network partitioning method only needs to be carried out once before parallel computing, so it does not affect the effect of parallel computing procedure itself. However, the label propagation algorithm itself can also been paralleled, while is suitable for the calculation tasks with the distribution scheme needs to be adjusted in real time, which is not detailed here.

### 3.3 Realization of the Parallelization of EMT simulation in InterPSS

Since the MATE algorithm and the label propagation algorithm are both universal, most parallel computing techniques are feasible to realize the parallelization of EMT simulation in InterPSS. In this paper, it is realized based on the multithreading technology under JAVA utilizing identical processors in traditional PCs.

Ideally, if the computing tasks are intensive, for hardware platforms with N identical processors, setting the size of the thread pool to N+1 can achieve optimal utilization efficiency. In this paper, the transient simulation program based on InterPSS needs to be parallelized for each integration step, which is a computing-intensive task. Therefore, before starting parallel computing, the original power network model should be divided into N+1 subnets.

For the parallel calculation of each time step, firstly, each subnet is solved independently without considering the injection current of the interacting tie line. The calculation of each subnet is submitted to the thread pool as an independent task, which is calculated independently in different threads.

After the tasks of each thread are completed, the results of each sub-network are coordinated and modified based on the method introduced in [14]. Because this task requires multiple subnets to be operated at the same time, it cannot be performed in a separate thread, hence it can't be parallelized further. However, the sub-network method in this paper has reduced the scale of sub-network contact as much as possible. These non-parallelizable operations account for a small proportion of the whole
computing task and have a limited impact on the overall computing performance. When all the calculation tasks of this step are completed, the simulation is completed. We use the common methods in JAVA to achieve parallelization, and there are not many special operations. The parallelization method proposed in this paper is universal and does not depend on specific numerical algorithms and hardware resources. Therefore, it can be easily extended to other high-performance computing environments.

4. Case study

4.1 Generating numerical examples
For parallel computing tasks, the larger the scale of the example, the better the parallel effect. At present, the scale of power system examples that can directly carry out EMT simulation under InterPSS are limited. Therefore, similar to the method introduced in [22], the authors can generate suitable example systems with different scales based on the conventional relatively small-scale IEEE standard examples.

In this paper, the famous IEEE 3-machine and 9-bus case is used to generate large cases. The InterPSS transient simulation network object of the 3-machine 9-bus case is generated first. Then we can clone this object to get another network object, and through the object-oriented way to splice two smaller objects into larger objects. On the premise of ensuring the normal simulation of the transient simulation network, some of the nodes and edges will be deleted according to a specific mechanism to avoid the resulting network being too "uniform".

Fig. 5 Splicing procedure for bulk power system cases
Since then, all the splicing results will be used as "material" for further expansion of network objects. For example, if the 3-machine 9-node network object is regarded as an "atomic" network, another network can be obtained after cloning and simple disturbance, and a larger network can be obtained by splicing the two together. If further expansion required, one can splice any combination of the three existing networks. Finally, more and more network objects of different sizes with differences can be obtained. The specific operation process is illustrated in Fig. 5.

As a comparison, the reference [21] is based on the standard case of 10 machines and 39 nodes to generate power network models of different sizes, and the network topology is highly regular. However, the topology of the case used in this paper is relatively irregular. Obviously, the effect of grid separation and parallel calculation may not as good as that from [21], but it is not due to the methods, but due to different test cases, while the case in this paper should be more in line with the actual power grid situation.

4.2 Design of the simulation experiments to show parallelization effect

Suitable experiments are defined to verify the accuracy of the results obtained by using this method. Using the example with the scale of 1800 nodes mentioned above, the numerical integration step is set to 0.005 seconds, and the total simulation time is 20 seconds. The specific sequence of events is as follows. At 0.1 second, the three-phase short-circuit disturbance is set for node 300, and the duration is 0.25 seconds. At 0.2 seconds, the three-phase short-circuit disturbance is set for node 820, and the duration is 0.35 seconds. At 0.25 seconds, the three-phase short-circuit disturbance is set for node 1620, and the duration is 0.55 seconds.

The curve of generator power angle at node 299 before and after parallelization is shown in Fig. 6.

![Fig. 6 Comparison of output results before and after parallelization](image)

Table 3 shows the comparison of the parallelization performance of different simulation networks with the same disturbance. The simulation was performed in a Windows Server 2012 R2 64-bit virtual machine with 16.0GB memory and up to 16 CPU.

| System scale (number of nodes) | Non-parallel computing time/ms | Number of threads | Parallel computing time/ms | Acceleration ratio |
|-------------------------------|-------------------------------|-------------------|----------------------------|-------------------|
| 1800                          | 88687                         | 4                 | 66563                      | 1.332             |
| 4500                          | 148662                        | 8                 | 74361                      | 1.999             |
| 6200                          | 235580                        | 12                | 98234                      | 2.398             |
| 7200                          | 350819                        | 16                | 117901                     | 2.976             |
4.3 Result analysis
Firstly, the two curves in Fig. 6 are completely coincident because the decoupling calculation and the coordination of the results are linear operations. The simulation algorithm itself does not change and will not generate new errors from parallelization.

Secondly, it can be seen from Table 4 that the larger the scale of the simulated power grid, the greater the speedup obtained by parallelization. For further analysis, define

\[
\text{Normalized Acceleration Ratio} = \frac{\text{Acceleration ratio}}{\text{Thread number}} \times \frac{\text{Network Scale}}{1}
\]

(1)

to characterize the effect of acceleration more accurately. Fig. 7 shows the curve of relationship between the normalized acceleration ratio and the scale of power network in Table 4. As can be seen from the figure, the effect of unit speedup decreases with the increase of the size of the network. With the increase of the size of the network, the amount of information that different networks interact with each other is also increasing, which will weaken the effect of parallelization. The existing reports on parallel computing have similar conclusions. However, as can be seen from the figure, with the increase of the scale of the power grid, the downward trend is slowing down. Limited by the available hardware resources, the conclusion has not been directly verified in larger computing resources. However, it can be expected that with the increase of the size of the power grid and the number of available threads, the parallelization effect of this algorithm will be maintained to a considerable extent.

Fig. 7 Relationship between the normalized accelerative ratio and the scale of power network

5. Conclusion
In this paper, a general parallel computing framework based on the EMT simulation function of InterPSS is proposed, which can achieve flexible and efficient network partition and decoupling calculation of different subnets based on the results. It is parallelized in JAVA multithreading environment as a test to show the possibility of “general framework”, and a good effect is achieved. The large-scale simulation case shows that the parallelization effect of this method is less affected by the increase of simulation scale.

The method in this paper is not limited by specific high-performance computing resources. In particular, the label propagation algorithm proposed in this paper can in fact partition the highly heterogeneous high-performance computing platform composed of hardware with great differences in computing power. Moreover, the decoupling calculation method provided has nothing to do with the specific partition scheme, which makes this method have a strong generalization ability. The methods will be more valuable for the computing tasks that need to be carried out through cloud-edge collaborative computing and communication systems in the future.

Limited by the available hardware resources, some potential values of this paper have not been fully verified, such as the slow decline of unit speedup, parallel computing under highly heterogeneous high-performance computing platforms and so on. All these can be implemented in near future.
Acknowledgement
This work is supported by Science and Technology Project of State Grid Corporation of China (Research on the parallization of electromagnetic transient and hybrid simulation for power system with HVDC and high-proportionated new energy based on heterogeneous hardware platform, No.SGSHDK00DWJS1900400).

Reference:
[1] XUE Wei, SHU Jiwu, YAN Jianfeng, et al. Cluster-based parallel simulation for power system transient stability analysis[J]. Proceedings of The Chinese Society for Electrical Engineering, 2003, 23(8): 38-43.
[2] Chai J S, Bose A. Bottlenecks in parallel algorithms for power system stability analysis[J]. IEEE Transactions on Power Systems, 1993, 8(1):9-15.
[3] Decker I. C., Falcao D. M., Kaszkurewic Z. E., Parallel implementation of a power system dynamic simulation methodology using the conjugate gradient method[J]. IEEE Transactions on Power System, 1992, 7(1):458-466.
[4] Alvarado F L. Parallel solution of transient problems by trapezoidal integration[J]. IEEE Traps on Power Apparatus and Systems, 1979, 98(3).
[5] Scala M L, Brucoli M. A Gauss-Jacobi-Block-Newton method for parallel transient stability analysis[J]. IEEE Transactions on Power Systems, 1990, 5(4):1168-1177.
[6] Scala M L, Sbrizzai R, Torelli F. A pipefined-in-time parallel algorithm for transient stability analysis[J]. IEEE Transactions on Power Systems, 1991, 6(2):715-722.
[7] Granelli G P, Montagna M. Relaxation-Newton methods for transient stability analysis on a vector/parallel computer[J]. IEEE Transactions on Power Systems, 1994, 9(2):637-643.
[8] Scala M L, Bose A. Relaxation/Newton methods for concurrent time step solution of differential-algebraic equations in power system dynamic simulations[J]. Circuits & Systems I Fundamental Theory & Applications IEEE Transactions on, 1993, 40(5):317-330.
[9] HONG Chao. A parallel-in-time algorithm for analysis of power system transient stability[J]. Power System Technology, 2003, 27(4): 31-35.
[10] Mahseredian J, Dennetiere S, Dude L, et al. On a new approach for the simulation of transients in power systems[J]. Electric Power System Research, 2007, 77(11): 1514-1520.
[11] MU Qing, LI Yalou, ZHOU Xiaoxin, et al. A parallel multi-rate electromagnetic transient simulation algorithm based on network division through transmission line[J]. Automation of Electric Power Systems, 2014, 38(7): 47-52.
[12] Mike Zhou. https://github.com/interpss[OL]. 2021.
[13] He Yangzan, Wen Zengyin, Power System Analysis[M], Wuhan: Huazhong University of Science and Technology Press, 2016.
[14] Tomim M A, Marti J R, Wang L. Parallel solution of large power system networks using the Multi-Area Thevenin Equivalents (MATE) algorithm[J]. International Journal of Electrical Power & Energy Systems, 2009, 31(9):497-503.
[15] Huang Q, Vittal V. Integrated transmission and distribution system power flow and dynamic simulation using mixed three-sequence/three-phase modeling[J]. IEEE Transactions on Power Systems, 2017, 32(5):3704-3714.
[16] Kernighan B W, Lin S. An efficient heuristic procedure for partitioning graphs[J]. Bell System Technical Journal, 2014, 49(2):291-307.
[17] Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory[J]. Czechoslovak Mathematical Journal, 1975, 25(4):619-633.
[18] Irving M R, Sterling M. Optimal network tearing using simulated annealing[J]. Generation Transmission & Distribution IEE Proceedings C, 1990, 137(1):69-72.
[19] Zhu X, Ghahramani Z. Learning from labels and unlabeled data with label propagation[J]. Tech Report, 2002, 3175(2004):237-244.
[20] Zhao Siqi. Research and improvement of community discovery algorithm based on Spark for
large scale complicated networks[D], Changchun: Jilin University, 2018.

[21] Birchfield A B, Xu T, Gegner K M, et al. Grid structural characteristics as validation criteria for synthetic networks[J]. IEEE Transactions on Power Systems, 2017:1-1.

[22] HUANG Yuanyang. A parallel algorithm for electromechanical transient simulation by recursive block matrix inversion[J]. Power System Protection and Control, 2019,47(24):8-14.