Development of Distributed Simulation Platform for Power Systems and Wind Farms

Hu, Rui; Hu, Weihao; Chen, Zhe

Published in:
Proceedings of the 41th Annual Conference of the IEEE Industrial Electronics Society, IECON 2015

DOI (link to publication from Publisher):
10.1109/IECON.2015.7392633

Publication date:
2015

Document Version
Accepted author manuscript, peer reviewed version

Development of Distributed Simulation Platform for Power Systems and Wind Farms

Rui Hu, Weihao Hu, Zhe Chen
Department of Energy Technology
Aalborg University
Aalborg, Denmark
rhu@et.aau.dk, whu@et.aau.dk, zch@et.aau.dk

Abstract—**The study of wind power system strongly relies on simulations in all kinds of methods. In industry, the feasibility and efficiency of wind power projects also will be verified by simulations at first. However, taking time cost and economy into consideration, simulations in large scales often sacrifice model details or computing precision in order to gain acceptable results in higher simulating speed and lower hardware costs. To balance the contradiction of costs and performance, in this paper, a novel distributed simulation platform based on PC network and Matlab is proposed. Compared with other simulation approaches, this platform can improve the speed of simulations in large scales without sacrificing details or precision largely. By means of connected computers and paralleled models, it becomes easier to study further about harmonics, control strategies in current experiment conditions. Through the data interfaces, the platform can import practical data to simulate environment situations, faults and devices, which makes the simulation much more close to reality and forms a test-bed for wind farms and power systems as well. Thus, The platform can connect to certain Supervisory Control and Data Acquisition (SCADA) systems and Energy Management System (EMS), etc. to realize non-real-time semi-physical simulation for wind farm and power system control researches.**

Keywords—distributed simulation; testbed; power system; wind farm

I. INTRODUCTION

Nowadays, more and more power electronic devices have penetrated into power system. To study and simulate details of this system especially with tens or hundreds wind turbines needs smaller time step than traditional power system simulations. It requires a large amount of investments to build a pure physical experimental platform for wind turbines [1]. Besides the inflexibility of the physical dynamic models, there are also difficulties and constraints to conduct related experiments on it to test the interrelationship between wind farms and power system as well, because immature or thoughtless control strategies may cause damage to the physical platform and power system [2]. Due to the complexity of wind farms, it's more difficult and not economic to use physical models to conduct experiments in large scales. Currently, when simulating wind farms or a cluster of wind turbines, off-line simulation tools like Matlab/Simulink, PSCAD are frequently-used. Although software simulation is simpler to construct and operate, it's difficult for them to communicate with SCADA or EMS to form a HIL system for practical projects. Firstly, simulation software normally is less supportive to simulate the influence caused by communications among devices. This kind of simulations now are usually investigated by separate software such as OpenNET, NS, etc. Secondly, it is inconvenient and low-efficient to change parameters instantly to simulate random events. At last, the time cost may be unacceptable when the model has a larger scale, higher order or higher precision. These years, to make models closer to reality, notable efforts has been made by semi-physical simulation tools such as Hardware in Loop (HIL) and Rapid Controller Prototype (RCP) [3]. HIL systems usually have friendly interfaces with simulation software such as Matlab [4]. Technically, they are computers which have high performance hardware with digital IO boards and special real-time operation system with model codes [5]. Its real-time feature based on hardware and software not only makes the target model much closer to reality, but also makes the platform more flexible than totally physical one. Due to these characteristics, HIL can shorten the development period and test controllers in labs only use digital models [6]-[8]. In reference [5], offline models deployed in distributed real-time simulation platforms are discussed and tested. However, building a HIL platform like RTDS, RT-Lab and Speed-goat for large scale simulation requires much more investments and operating skills than software simulation. Immature models or operation faults may cause fatal damage to the platform as well. Whilst, it will lose the real-time feature when the model scale or precision exceeds the hardware capability. Normally, the verification and simulation mainly focus on the control strategies and algorithms, the communication performance is omitted or just substituted by approximate control parameters like delays, unbalanced data loads. From practical views, the communication system is one of the most important factors which decide whether the control system can be practiced, but it is ignored in most cases. Hence, a test-bed which owns the ability to integrate controllers and large scale of simulation models in a low cost is significant.

Power systems and wind farms are usually comprised of large scale of generators and all kinds of components. Many of them will be omitted when taking computation cost into account. For instance, many assumptions are made to simplify the model adequately. In the transient stability research work for wind farms, equivalent model with same MVA rating is usually adopted. The main assumptions are as following [9]-[11].

1) all the wind turbines are operating at the same constant wind speed;
2) all the wind turbines are equipped with a rapid current controller;
3) the initial states are the same.

Although the results are acceptable when simulations focus on macroscopical problems, further problems and details may be concealed. The balancing of computation and cost could be an obstacle for further researches on interaction among wind farms, power systems and other distribution systems.

Motivated by the aforementioned situations of power system and wind power simulation, a PC network based simulation-testbed platform is proposed in this paper. Both advantages of software simulation and semi-physical simulation are considered. The platform makes it possible for controller testing in a large scale of system in non-real-time state and conducting simulation with high flexibility in an economical way.

II. Principles And Design of Distributed Simulation Platform based on PC Network

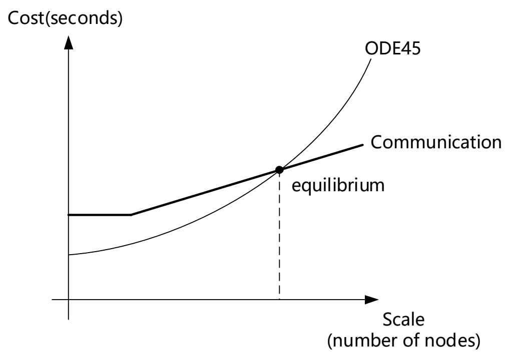

Figure 1. Cost relationship between simulation algorithm and communication

The principles and feasibility of distributed simulation is quite obvious. The complexity of numerical algorithms, such as commonly used Runge-Kutta methods, is non-linear and the time consumption usually increases largely with the augmenting of the model scale, improved precision requirements, higher model order, or reduction of the simulation step. Using distributed computation and simulation can effectively shorten the time cost by dividing the computation jobs and assigning them to clusters of paralleled running computers. Distributed simulation system also needs time for data exchange and synchronization especially in cyber based paralleled computation systems. Nevertheless, when the simulation time cost in a single computer exceeds the maximum communication in one simulation step or control cycle, the model is suitable for distributed simulation.

The time cost of the communication of a system based on Ethernet IEEE 802.3 standard according to figure 3 can be estimated by the following equations:

$$\text{Cost} = \frac{\text{Eth}(\sum_{i=1}^{n} x_i) + \text{Eth}(\sum_{i=1}^{n} y_i)}{K_{baud}} + C_{switch} + C_{protocol} \quad (1)$$

where

$$Eth(\text{m}) = \begin{cases} 64 & m \le 46 \\ 18 \times ceiling(\text{m}/1500) + \text{m} & m > 46 \end{cases} \quad (2)$$

K_{baud} is the communication speed of the Ethernet. 1000 Mbps and 100 Mbps are normally applications. x_i is the i*th* node data length which sent by the master node. y_i is the ith node data length which received by the master node. C_{switch} is the data exchange time cost switches. $C_{protocol}$ is the time cost of protocol transformation. m is the data length which transported in the network.

In the platform architecture illustrated in Fig. 3, Model Exporting Tool is responsible for export models and its signals, parameters, and variants from off-line simulation software like Matlab or PSCAD with the help of secondary development tools they have provided. Consequently, the models can be converted to editable codes in C++ or other languages.

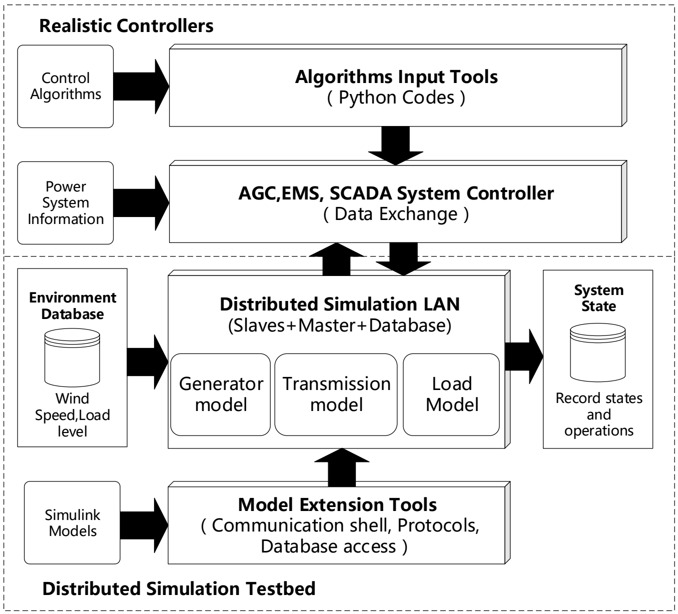

Figure 2. Architecture of distributed simulation-testbed platform

Model Modify Tool adds extra codes into the models. The extra codes consist of Communication Module and Virtual Device Module. Communication Module is the core component for data exchange among simulation nodes. It keeps the exchanged data from mismatch according to the synchronous methods depicted in figure 8 and figure 9. Virtual Device Module is the emulator of the communication interface of a practical device, like wind turbine or energy storage unit. Protocols like Modbus, Profi-Bus, etc. are integrated in this module to wrap the required data. It makes the real controllers like SCADA systems able to communicate and be compatible with the model.

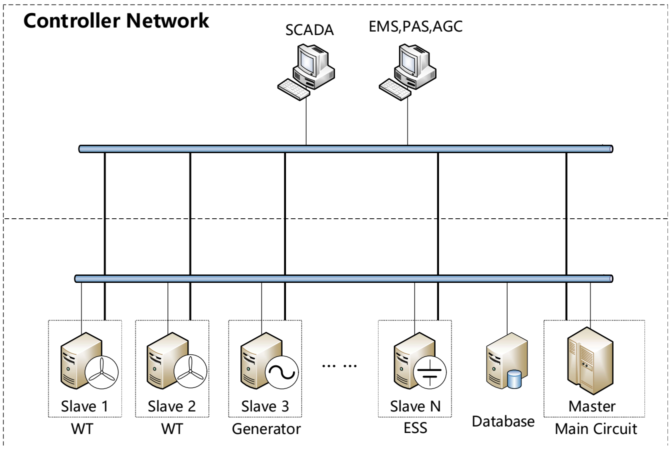

Figure 3. Network framework of distributed simulation-testbed platform

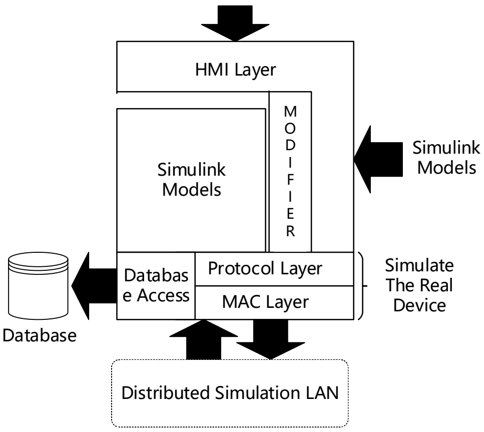

Figure 4. Software structure of a single simulation node

After applying these modules in a single PC node, the framework can be delineated in figure 4. In the platform architecture, all PC nodes import simulation model codes which generated by Simulink or PSCAD through User Interactive Layer. The Modifier Module implements the aforementioned function by adding extra codes. After this step, the raw model is able to communicate with other computer nodes, receives synchronization commands and control instructions. The extra codes can be classified into Protocols Parsing Module source codes, Data Storage Module codes and Communication Layer codes respectively.

The local area network consists of one master PC node and many other slave nodes logically. The master node simulates the impedance characteristics of the main circuit and controls the communication logic in order to synchronize the model states among all computers. A slave node usually corresponds to a certain device controller model which carries the control strategies and algorithms for a single device. Slave nodes and the master node maintain data coherence of microgrid models through communication.

III. Division and Configuration of Models

Because of the memory capacity limitation of a single computer, when simulating wind farm of large scale or with high precision, the memory requirement and simulation time cost can be unacceptable. The proposed platform simulates wind farms and power systems in distributed way. It separates control block and main circuit to simplify models on each node to solve this problem [8].

The main circuit includes impedance, buses, loads, controlled sources and power electronic devices to simulate electrical and physical characteristics. Slave nodes are responsible for simulating each device's characteristic curves, control methods and algorithms. In each simulation step, slave nodes require electrical parameters, physical values from master node to generate control signals according to its own algorithms. Then the signals are transported to master node to control the power electrical switches and controlled sources. Figure 5 shows how the permanent-magnet synchronous generator (PMSG) model is divided in the proposed platform according to the principle.

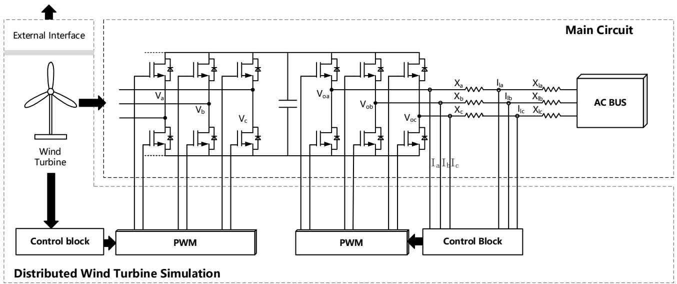

Figure 5. Division of a PMSG model

In a similar way, taking energy storage unit (ESU) slave simulation node as an example, the principles of model division method are shown in figure 6. The battery's characteristics and control logic will be put into ESS slave node, the three-phase bridge will be placed in the master node. Before every simulation step, master node sends output voltage and output current to ESU slave node. ESU slave node sends control information for three-phase bridge to master node as well.

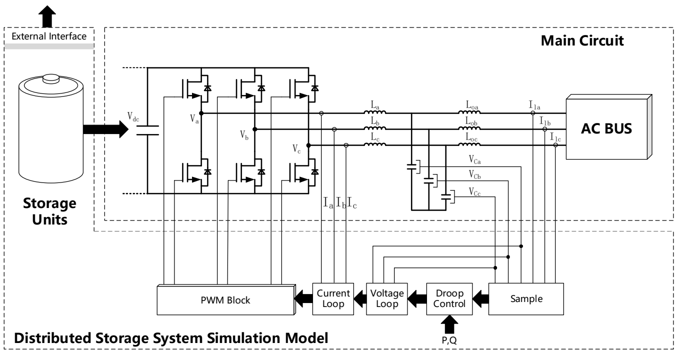

Figure 6 Division of an ESU model

The external interface added to the model can be used to input real environmental data to the model, like wind speed, illumination conditions, etc.

IV. EXPORTING, MODIFYING, GENERATING AND SYNCHRONIZING OF SIMULINK MODELS

After modeling inverters, loads, generators and wind turbines respectively, it's very convenient to convert Simulink models to gcc C/C++ codes or Microsoft Visual Studio project by using Code Generation tools provided by Matlab [12], [13]. The raw codes are not distributed and cannot communicate with other models. Hence, the mentioned shell codes should be added to perform a second development. The shell codes also integrate protocols like TCP/IP Modbus, Profi-bus, etc. to make models controllable according to the aforementioned description. Via these modifications, the computer nodes are more like real devices which can interact with SCADA, AGC, AVR, etc. Researchers can change environment data like wind speed, operate the devices in the power systems randomly, or simulate device communication faults. All the data and operation will be recorded in the database which can be implemented with MySQL, SQL Server and so on.

To make this platform become a semi-physical simulation system, the extra codes must have clock control interface as well as the real controllers which are connected to it. Different from current HIL systems, the platform makes the real clock inside the controllers comply and synchronize with virtual simulation clock which could be much slower than the real one inside the platform. With this interface, the platform will output time synchronization signal to real controllers in order to make their internal clock coordinated with model's simulation clock. Because the platform usually runs in non-real-time state, the internal clock of real controllers should be slower when they connected to this platform than to real power system devices. This is the only requirement for real controllers, fortunately, it is easy to achieve by the PC and digital technology based control systems. This principle is shown in figure 7.

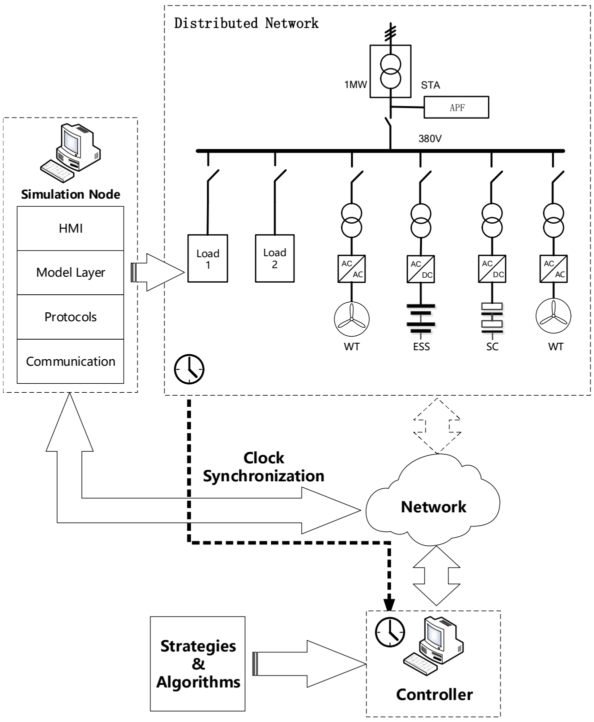

Figure 7. Principle of clock synchronization

Model state synchronization is the vital factor to guarantee the validate results. All computers should complete data synchronization before the function rt_OneStep() is called. This function is generated automatically by the tools of MatLab mentioned above. At first of each step, the master node sends corresponding main circuit states to all slave nodes and receives control information from them. Then it sends simulation signal to all nodes to start calculation of current step. Hence, the time cost of a single simulation step in distributed simulation systems is determined by the largest time consumption among the computers. The control logic and procedures of master node and slave nodes are shown in figure. 8 and 9. The simulation time will be calculated after every simulation step is finished, then the clock synchronization signal will be sent by master node to the controllers using standards similar to IEEE 1588. Besides, the import of real environmental data using the external interface added to the model also should be synchronized to the simulation clock too.

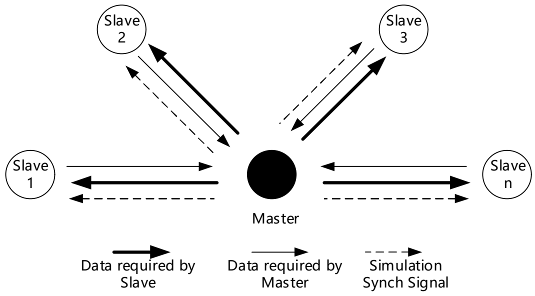

Figure 8. Principle of data synchronization

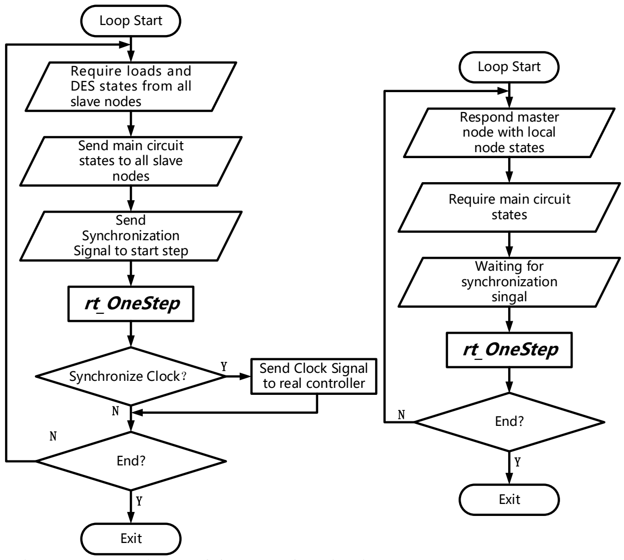

Figure 9. Procedures of data synchronization

V. EXPERIMENTS AND ANALYSIS

a) Feasibility Verification

Simulation for a 2 WM distributed system comprised of 4 storage units rated at 500 kW with a simple secondary frequency and voltage control strategy based on PI regulator is conducted to verify the feasibility of the platform. At the 4th second, the restoring procedure started. The frequency returned to 50Hz at 5.3s and the voltage is restored at 4.8s. From the results in figure 10, it can be concluded that proposed distributed simulation didn't affect the basic functions of origin simulation model, the platform with proposed concepts and principles is feasible.

b) Performace Verification

In addition, Simulation of a customized IEEE 30 buses system with relatively heave computation burden has been conducted on proposed platform. The simulation is just for verification of the accelerating effects and speed comparing, there is no specific research goal attached. Figure 11 is the IEEE 30 Buses system used for testing in heavy computation burden. The system configuration and data are employed from [14], [15]. To compare the time consumption with traditional offline simulation, the verification simulation work is conducted with different configurations from 2 computers to 4 computers to gain the influences caused by different number of computation nodes. Besides, the simulation is conducted alone in one computer using traditional off-line simulation software. According to the outcomes, it has been proved that the platform and model division method do have the accelerating effects. The simulation of heavy computation burden can be accelerated in some extend without affecting the precision.

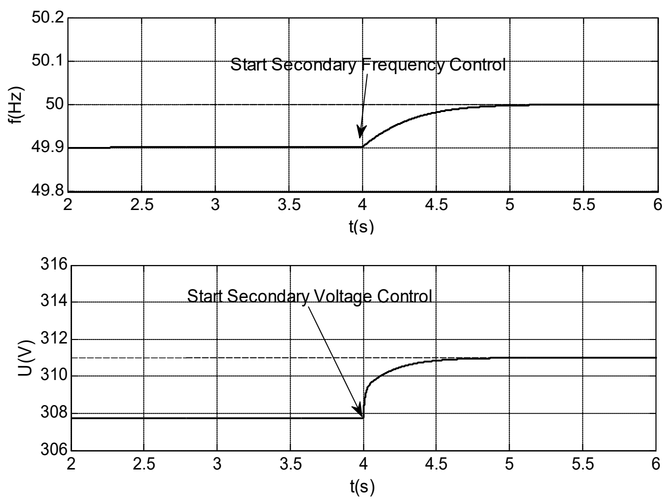

Figure 10. Results of secondary control conducted on distributed simulation system

TABLE I Comparison Of Time Consumption Using Different Distributed Scales

Number of Computers	Real time consumption (second)	Simulation time (second)	Configuration Parameters	
			Slover	Computer
1(offline)	383.6	1	•**Solver:** Discrete ODE 23 •**Step:** 50e-4 s	•**CPU:** Intel Core i5 3230M 2.6Ghz •**RAM:** 4G DDR3 1333 Mhz
2	334.2	1		
3	269.1	1		
4	221.9	1		

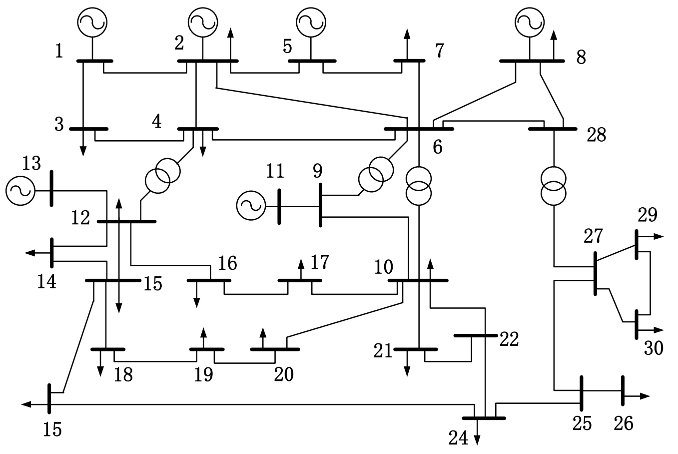

Figure 11 IEEE 30 Buses system

VI. Conclusions and Future Work

Aiming at utilizing advantages of software and semi-physical simulation, shortening time and reducing investment of projects, a distributed simulation platform is proposed. The principle of the platform is simple. Utilizing a cluster of computers to share the calculation loading. Experiments and simulations conducted on it have proved that the platform has the potential to save time and investment for wind farm and power system experiments. Wind farm models can include more details with less concern about computation burden. It is also probably a more economic approach to test and verify system controllers than real-time semi-physical simulation platform. It also may benefit the education.

In the future, more optimization work will be needed to improve the platform's performance. Further efforts should be made to balance the computation burden among each computers.

References

[1] Guo, Honghao, et al. "Real-time simulation of BLDC-based wind turbine emulator using RT-LAB." Electrical Machines and Systems, 2009. ICEMS 2009. International Conference on. IEEE, 2009.

[2] GUO Li, WANG Chengshan. Dynamical Simulation on Microgrid with Different Types of Distributed Generations [J]. Automation of Electric Power Systems, 2009 ,33(2): 82-84.

[3] LIU Jing-bo, XIAO Wen-jing, ZHOU Hong-lin. RTDS-based Hardware-in-the-loop Test of Grid-side Converter in the Wind Power Generation System [J]. Power Systems, 2013 ,47(10): 38-39.

[4] ZHOU Lin, JIA Fang-chen, GUO Ke, FENG Yu. Testing analysis on a Kind of PV Generation Simulation System Using RT-LAB [J]. High Voltage Engineering, 2010，36(11)：2814-2822.

[5] Liu, Xian. "Quality of optical channels in wireless SCADA for offshore wind farms." Smart Grid, IEEE Transactions on 3.1 (2012): 225-232.

[6] Hu, Zurong, et al. "Research on speed control system for fixed-pitch wind turbine based on disturbance observer." World Non-Grid-Connected Wind Power and Energy Conference, 2009. WNWEC 2009. IEEE, 2009.

[7] LI Peng, DING Chengdi, WANG Chengshan, al et. A parallel algorithm of transient simulation for distributed generation system based on multi-core CPU [J] Proceedings of the CSEE, 2013, 33(16): 171-180.

[8] Ullah, Nayeem Rahmat, Torböjrn Thiringer, and Daniel Karlsson. "Voltage and transient stability support by wind farms complying with the E. ON Netz grid code." Power Systems, IEEE Transactions on 22.4 (2007): 1647-1656.

[9] Gautam, Durga, Vijay Vittal, and Terry Harbour. "Impact of increased penetration of DFIG-based wind turbine generators on transient and small signal stability of power systems." Power Systems, IEEE Transactions on 24.3 (2009): 1426-1434.

[10] Conroy, J., and R. Watson. "Aggregate modelling of wind farms containing full-converter wind turbine generators with permanent magnet synchronous machines: transient stability studies." IET Renewable Power Generation 3.1 (2009): 39-52.

[11] YANG Daliang, LU Ziguang, HANG Naishan. Design of integrated real-time simulation platform for power electronic system [J]. Electric Power Automation Equipment, 2011, 31(10): 139-143.

[12] LI Yang, GENG Chang-yu, ZHANG Li-fen. Data synchronization over different platforms under module of socket communication [J]. Journal of Beijing Institute of Technology, 2002, 31(10): 81-87.

[13] Min, Liang. Decomposition algorithms for multi-area power system analysis. Diss. Texas A&M University, 2007.

[14] Totonchi, Ibrahem, et al. "Sensitivity analysis for the IEEE 30 bus system using load-flow studies." Electric Power and Energy Conversion Systems (EPECS), 2013 3rd International Conference on. IEEE, 2013.

[15] Paquin, Jean-Nicolas, et al. "Real-Time and Off-Line Simulation of a Detailed Wind Farm Model Connected to a Multi-Bus Network." Electrical Power Conference, 2007. EPC 2007. IEEE Canada. IEEE, 2007.