TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/12 pp133-143
Volume 19, Number 2, April 2014

An Integrated Cyber-Physical Simulation Environment for Smart Grid Applications

Yuxin Wan, Junwei Cao*, Shuqing Zhang, Guoyu Tu, Chao Lu, Xingtao Xu, and Keqin Li

Abstract: The concept of Cyber-Physical Systems (CPSs), which combine computation, networking, and physical processes, is considered to be beneficial to smart grid applications. This study presents an integrated simulation environment to provide a unified platform for the investigation of smart grid applications involving power grid monitoring, communication, and control. In contrast to the existing approaches, this environment allows the network simulator to operate independently, importing its results to the power system simulation. This resolves conflicts between discrete event simulation and continuous simulation. In addition, several data compensation methods are proposed and investigated under different network delay conditions. A case study of wide-area monitoring and control is provided, and the efficiency of the proposed simulation framework has been evaluated based on the experimental results.

Key words: control; cyber-physical system; simulation; smart grid; wide-area monitoring

1 Introduction

Cyber-Physical Systems (CPSs) are considered among the leading technological innovations of recent years and have been referred to as the next revolution in information technology[1]. CPSs combine computation, communication, and control techniques to provide reliable and robust integration of computing and physical processes. The intrinsic characteristics that distinguish CPSs from other systems are their real-time responses and high reliability[2]. CPSs can be applied in many areas, such as medicine, aerospace, transportation, and defense[3]. Electrical power grids, which are among the largest and most complex physical systems, are typical CPS case studies. These systems are known as Cyber-Physical Energy Systems (CPES)[4] or smart grids.

Reliability and efficiency are the most important problems encountered in the modern power grid[5]. The impetus driving the development of the smart grid is the unobservability, unreliability, low resiliency, and low efficiency of the existing power grid. The smart grid offers a self-monitoring and self-healing power delivery architecture[6] with two-way communication. Its foundation is a reliable and stable two-way communication infrastructure that satisfies both real-time and off-line applications[7,8]. Smart grid applications are sensitive to network delay and packet losses; therefore, the influence of a communication network on such applications must be investigated. This study presents an integrated simulation environment developed for both the power grid and the communication network that supports the

---

- Yuxin Wan, Junwei Cao, and Guoyu Tu are with the Department of Automation, Research Institute of Information Technology, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: wanyx04@mails.tsinghua.edu.cn; jcao@tsinghua.edu.cn; tu@tsinghua.edu.cn.
- Shuqing Zhang and Chao Lu are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China. E-mail: zsq@tsinghua.edu.cn; luchao@tsinghua.edu.cn.
- Xingtao Xu is with State Grid Jiangsu Electric Power Company, Nanjing 210024, China. E-mail: xxt3322@sina.com.
- Keqin Li is with Department of Computer Science, State University of New York, New Paltz, NY 12561, USA. E-mail: lik@newpaltz.edu.

* To whom correspondence should be addressed.

Manuscript received: 2014-02-18; revised: 2014-02-24; accepted: 2014-02-25

modeling and simulation of smart grid applications.

The integrated simulation platform consists of three modules, i.e., power grid simulation, communication network simulation, and control simulation for applications. The power grid simulation module, TH-STBLT[9], is a power system simulator designed by Tsinghua University that is capable of simulating the transient states of a power system with more than 1000 nodes. The communication network simulation module is currently based on queuing systems that interact with databases and is compatible with other network simulation tools, such as NS2[10]. The control simulation module is developed based on Matlab and is able to process data in the presence of network delays or lost packets. These three modules run independently and are connected through network programs, which makes it easy to replace any of the elements. These modules also share the same virtual clock, forming a closed-loop system. To evaluate the effectiveness of the proposed simulation platform, Wide-Area Monitoring and Control (WAMC) systems[11] are considered as a study case. Several data compensation methods are proposed corresponding to different communication network conditions, and the experimental results are analyzed for their effectiveness. Additionally, this environment has revealed a cyber-impact on power systems.

2 Literature Review

The smart grid refers to the integration of electrical grid and information technologies to provide a better energy supply infrastructure[12]. The communication network is employed to link these two elements. Because much of the existing work on the smart grid focuses on the impact of communication network performance on power grid control effectiveness[13, 14], it is urgent to build an integrated environment for both power grid and communication network simulations.

A modeling and simulation framework is proposed in Ref. [15] to analyze the performance of an IP-based communication network to support Phasor Measurement Unit (PMU)-based WAMCs. The authors used OPNET tools to analyze the impact of communication network capacity and background noise traffic on PMU data delays in WAMCs. However, this study focuses on the analysis of network delays without performing closed-loop experiments and tests.

Early work on closed-loop simulation can be found in Ref. [16], which describes the difficulties of simulating both the power grid and the communication network. The power grid requires real-time simulation based on a continuous simulator, while communication network simulation uses discrete events. Consequently, a high-level synchronization program was developed to set synchronization points to solve the problem. However, this solution may cause errors to accumulate, as the method tends to convert discrete simulation to continuous simulation. The approach to simulation presented in Ref. [17] relies on the same principle.

A simulation platform based on OpenDSS and NS2 is described in Ref. [18]. The influence of average message delays on PV (photovoltaic) voltage magnitude in a wireless network environment is analyzed in the context of solar ramping problems in solar PV systems. However, due to the independent structures of OpenDSS and the NS2 simulator, the platform is not well integrated, and a separate script file is required.

The problem of synchronizing simulated clocks between two different simulators is addressed in Ref. [19]. UNIX pipes are used to communicate, and internal events in NS2 are used to trigger the power grid simulator, Modelica. NS2 controls Modelica's simulation time. However, certain events triggered exclusively in physical systems cannot be discovered, which may lead to simulation errors.

In Ref. [20], a dynamic power grid simulation is modified into discrete events. A global scheduler is imported to record the event list of the power grid and the network simulator. In addition, the two simulators share a timeline. This platform can capture all the events of the two simulators and remove any accumulated errors introduced by process synchronization. However, there is still a chance of simulation errors, for example, when one simulator encounters an error, the other simulator is not informed until the event arrives. Simulation errors may be introduced during this interval.

As noted above, the primary difficulty of integrating power grid and network simulation is the synchronization between discrete event simulation and continuous simulation. However, this issue arises only if we combine the two different types of simulators into a cyber-physical co-simulation environment. The present work proposes a new approach to this problem, which is introduced below.

3 Integrated Cyber-Physical Simulation

Our approach is based on the simple idea that a power system simulator cares only about end-to-end network delays or packet losses. Therefore, it is no longer necessary to synchronize the discrete network simulator with the continuous power grid simulator. The network simulator operates independently to record data, while the power system simulator only imports the data from the network simulation. In a typical power system, data are collected periodically from PMUs or other sensor devices with a packet size less than 2 kB[15]. Thus, the impact of power grid data on the simulation of a communication network can be considered negligible. This simulation method would still be applicable if the volume of the power data grows large enough to affect the network simulation. This is possible because the power grid simulation is fully controlled; the system can indicate the time of occurrence of a fault, so that the network packet can be added at precisely that time during the network simulation. Simulation errors can be avoided because only one simulator is operating. The integrated cyber-physical simulation framework design is shown in Fig. 1.

Network simulators such as NS2 use record files to record the event of each packet, and end-to-end network delays or lost packets can be obtained from these files. The network simulation is carried out before the power system simulation, and the results are written into a database.

The power system simulator defines the starting time of a simulation, and the simulation interval is dependent on the specific data-sampling rate of each application. For example, a WAMC system's sampling rate is approximately 10-100 ms per sample. In our test case, a simulation interval of 30 ms is assumed for the transmission of data from the PMU to a Phasor Data Center (PDC). At the end of each simulation period, the power system simulator suspends the simulation and sends the results to the queue model. There is a virtual clock in the power system simulator; each packet sent is marked with the current simulation time stamp.

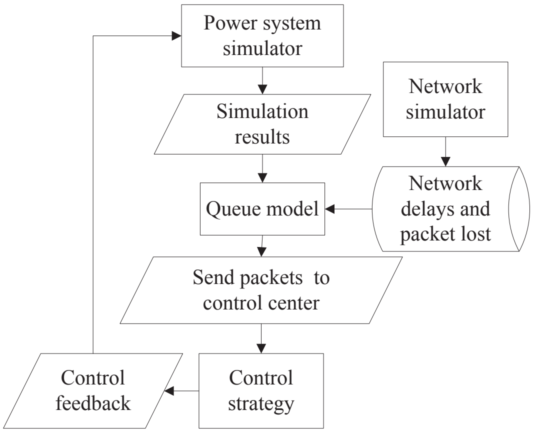


Fig. 1 The integrated cyber-physical simulation framework.

Once the queue model receives a data packet from the power system simulator, it reads the communication network simulation results from the database and calculates the current data packet delays or packet losses. The queue model then places this step packet into the queue according to its calculated delay. The time resolution of two neighboring queue positions depends on the accuracy requirements of the application. For example, if the control feedback returns in ms, the time resolution is 1 ms, which means that 1.1 ms or 1.2 ms can be treated as 1 ms. This resolution is applied to both the queue model and the controller to calculate the control feedback.

The queue model then pushes the current top-of-queue packets of one simulation period to the controller. For example, if the simulation interval is 30 ms with a resolution of 1 ms, the top 30 packets are sent to the controller.

The controller computes its outputs from the current data packets and the time resolution. For example, the 30 packets received after one simulation period are used to calculate the feedback data. Each of the 30 feedback results is processed and marked with a time stamp. The results are sent back to the power system simulator, and the controller waits for the next data packets.

When the power system simulator receives the feedback packets, it exits suspend mode and initiates another simulation period. Due to the time stamp included in the control feedback results, the simulator can specify the exact time step to incorporate the corresponding control data into the simulation.

As described above, the three simulation steps are interdependent to avoid conflicts in the simulation. The entire system state diagram is illustrated in Fig. 2.

4 System Implementation

4.1 Implementation of power system simulator

The TH-STBLT software package is a power system simulator developed by the Department of Electrical Engineering, Tsinghua University. It has recently been

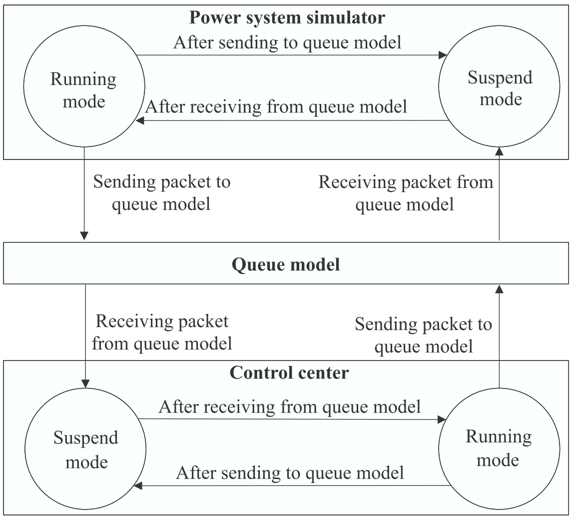


Fig. 2 System state diagram.

employed as the computational core component of a power system real-time hybrid simulation[9].

TH-STBLT is a typical and traditional simulator for large power systems. It is capable of providing dynamic simulation of power systems, including electrical distribution networks, generators and loads, motion of the interconnected generators, and coupled electromagnetic and mechanical phenomena after disturbances or faults.

In TH-STBLT, a power system is modeled using a set of ordinary differential equations and a large sparse algebraic equation[21]. The differential equations numerically describe the dynamic power components, while the algebraic equation characterizes the power network to investigate node voltages, branch currents, injected currents, etc.

TH-STBLT uses the implicit trapezoidal integration method to numerically solve the model of a power system. This method has been widely used in power system simulation due to its good convergence and stability. The LU factorization algorithm is employed to solve high-order polynomial equations with increased complexity. To improve the efficiency of processing fault or disturbance events, we propose adopting a method based on synthesis fault port admittance instead of the traditional approaches.

As Fig. 3 shows, the simulation begins by loading models of the system components and the dynamic parameters of each device. The steady-state load flow is then calculated, and the dynamic scenario is initialized. A fault/disturbance can be added to the system at a specified time, and the simulation proceeds to its core function. Differential Algebraic Equations (DAE) are applied to calculate the transient state of the system based on iterative computations. In Fig. 3, U_1 is the initial value, $U(k)$ and $U(k+1)$ refer to iteration steps, and tol is the convergence criterion. The simulation results are then collected and dispatched to the communication network. Then, the next simulation step begins, where t is the simulation time, dt refers to the simulation step size, and t_s is the simulation stop time set by the user. TH-STBLT has been widely employed and assessed in practice, and its performance and accuracy have been confirmed. We adopted the China Southern Power Grid in 2008 as an example for evaluation of TH-STBLT. This sample system contains 1393 nodes, 2371 branches, 244 generators, and 620 loads. The controllers include 244 types of exciters, 244 prime movers and governors, and 31 power system stabilizers.

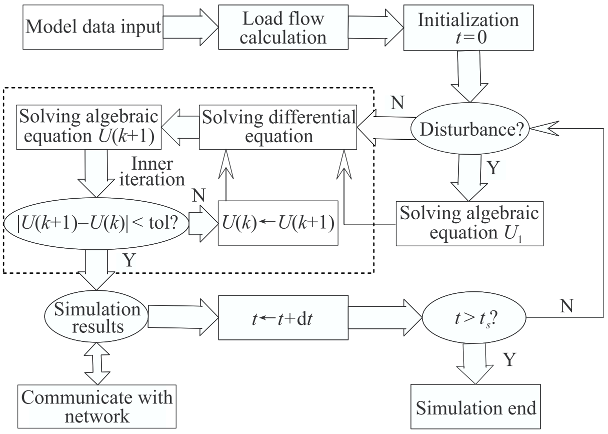
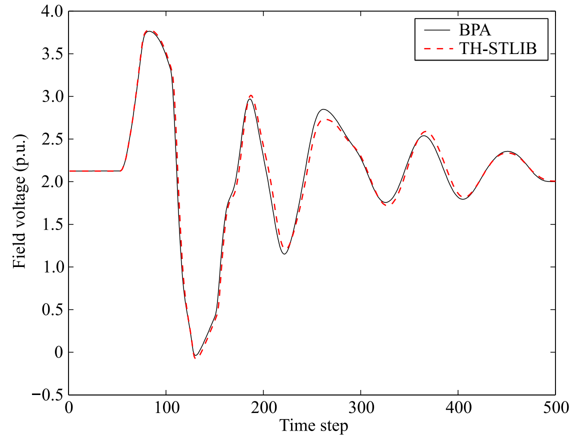


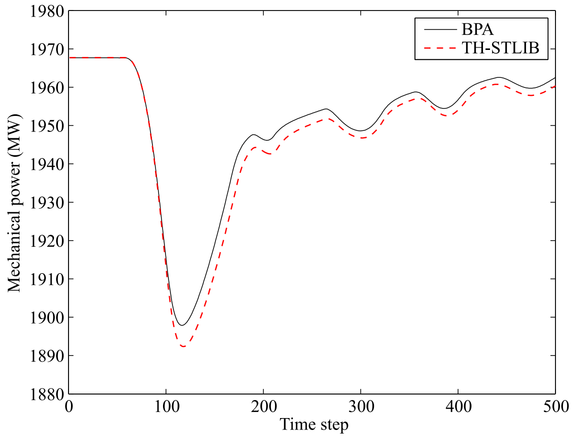
Fig. 3 The computational flow of the power system simulator.

The results of the simulation of a three-phase fault at a substation using TH-STBLT are presented in Fig. 4, compared with results obtained via the commercial software BPA, which is widely employed in power system dispatching, operating mode analysis, power system planning, etc. The simulation results demonstrate the effective and stable performance of TH-STBLT compared to BPA.

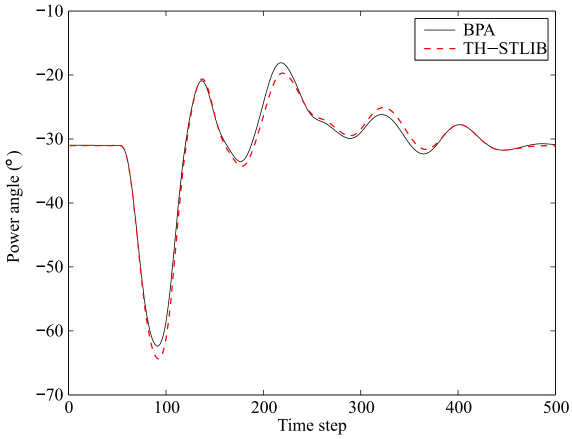
The transient characteristics of large power grids are indicated by oscillations after a fault or disturbance. Four parameters of these oscillations are used: amplitude, frequency, phase, and damping. As Fig. 4 shows, the frequency, phase, and damping produced by TH-STBLT are consistent with BPA, while the maximum amplitude difference is 0.42% of BPAs output.



(a) A large generator in Guangdong: Field voltage



(b) A large generator in Guangdong: Mechanical power



(c) A large generator in Guangdong: Power angle

Fig. 4 Performance comparisons (BPA vs. TH-STBLT).

4.2 Implementation of network simulation

Network delays due to noise traffic are generated by a circular queue. There are six steps in this circular queue, whose flowchart is illustrated in Fig. 5. In addition to reading packet delays from the database, packets based on special distributions can be generated.

Figure 6 gives an example of how the network simulation queue works.

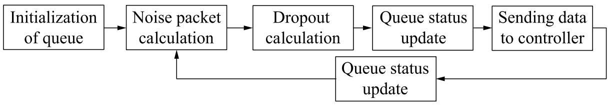


Fig. 5 Flowchart of the circular queue.

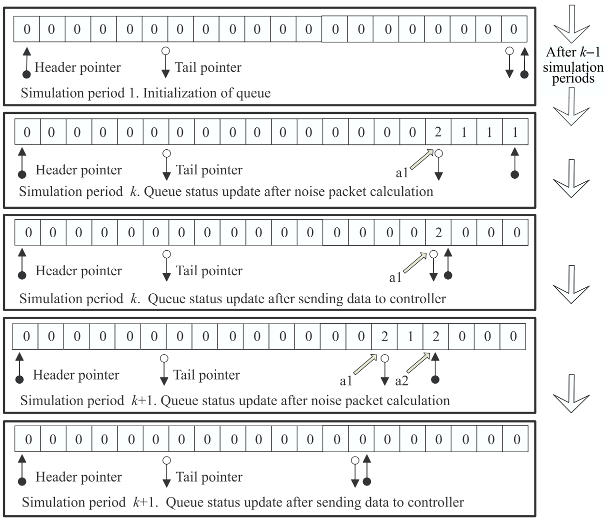


Fig. 6 Queue status update example.

The steps of the generation of packet delays are detailed below.

(1) Initializing the queue for one channel

A channel is the path from the power system simulator to the controller. The time resolution is set according to the application requirements. When starting a simulation, the Head and Tail pointers are both set to the initial position. We code the entries as type 0 to identify an empty queue position, type 1 for a noise packet, and type 2 for power grid data.

(2) Calculating the noise packet of the *k*-th period

Assume that a power grid packet a1 arrives in the k-th simulation period. The size of the noise packet is calculated according to the given noise distribution or read from the database. For example, if the first grid packet is delayed for three time units, then the number of noise packets is three, and the electrical grid packet will be placed in the fourth position in the queue. Each occupied position is marked, and the Tail pointer is moved.

(3) Updating the status of the queue after sending data in the *k*-th simulation period

Assuming that each simulation period contains a three-unit time resolution, three packets are sent to the controller from the queue during each period. At the end of the first simulation period, three packets are sent, and the Head pointer moves to the position marked as type 2. Positions whose data have been sent are re-marked as type 0.

(4) Calculating the noise packet of the *k*-th period

Assume that power grid packet a2 arrives during the second simulation period and that its calculated delay is two time units. Packet a2 is then placed in the sixth position in the queue. The Tail pointer moves back two positions.

(5) Updating the status of the queue on dropouts

In scenarios in which the calculated delay is larger than a defined threshold, the packet is regarded as lost. For example, in a normal distribution with mean μ and variance σ, the packet is lost if the calculated delay is larger than $\mu + 3\sigma$. The lost packet is not placed in the queue.

(6) Updating the status of the queue after sending data in the (*k*+1)-th simulation period

At the end of the second simulation period, three packets are sent. Both a1 and a2 are sent. The Head pointer then moves back three positions. Positions whose information was sent are marked as type 0.

4.3 Implementation of the controller

As Fig. 7 shows, the controller is implemented as three modules, i.e., data preparation, control algorithm with changeable parameter settings, and feedback return.

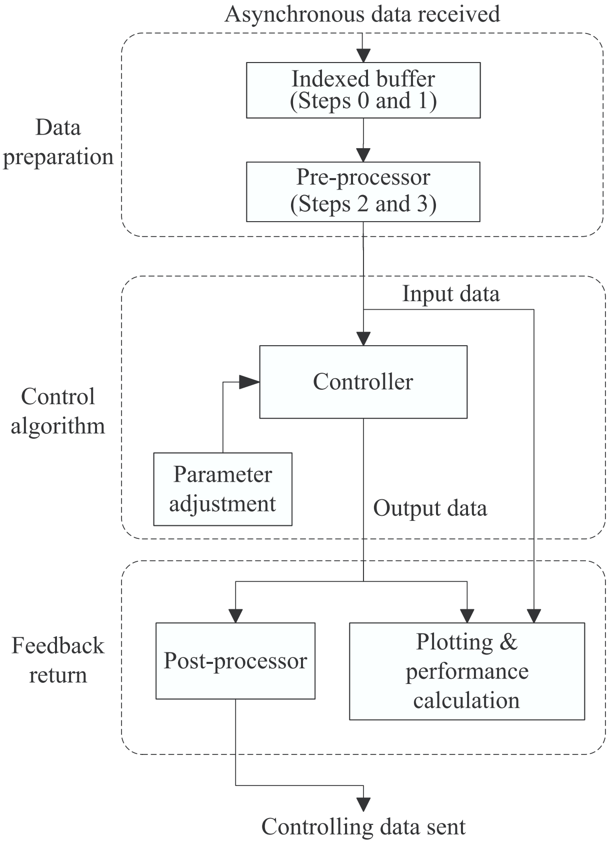


Fig. 7 Flowchart of the controller.

(1) Data preparation

In this module, asynchronous input data from all PMUs are synchronized, or compensated for if missing. One major challenge in wide-area monitoring is that the controller cannot provide outputs until input data from all the PMUs are collected, some of which may be delayed or even missing. As a trade-off between time and accurate control, data from each individual PMU are collected and sorted in an index buffer by their corresponding time stamps. The missing data must be filled in. Three steps are necessary to completing this task, which are described as follows.

Step 0 Initialize the clock label Latest-Time, indicating the latest time stamp of the data that have been utilized.

Step 1 Read data into the buffer during each period. As mentioned above, this buffer is used for data compensation. Three sub-steps comprise this process are as follows:

Step 1.1 When new data arrive, they are compared with the clock label Latest-Time. If their time stamps are less than Latest-Time, these data were collected earlier than the latest utilized data and should not be used as controller input. They are either stored for future use or simply discarded.

Step 1.2 Check the time stamps from different communication channels and place the data with the same time stamp into the correct position in the input buffer.

Step 1.3 Check to determine whether all the data with the latest time stamp have arrived. If yes, proceed to Step 3; otherwise, proceed to Step 2.

Step 2 Compensate for the missing data.

Three methods are designed to compensate for missing data and data without synchronization.

Single Channel History (SCH): When data from one channel are delayed or missing, they will be filled with the last consecutive data used in the previous period.

Full Channel Compensation (FCC): The time stamps of the latest data arrivals from different channels are recorded. The data from the last three sets in the previous periods are used to estimate the data with the present time stamp.

Single Channel Compensation (SCC): During one simulation period, the newly arrived data from each channel are used immediately. The missing data are estimated from the last three sets in previous periods.

Step 3 Pass all the signals to the controller as its inputs. Set Current Time as the time stamp of those signals.

(2) Control algorithm

In this module, with parameter adjustment, the controller's parameters can be adjusted corresponding to signal dynamics. The controller reads input data and calculates control outputs according to the control strategy or algorithm, which can be changed for different applications.

(3) Feedback return

In this module, the control signals are sent. Additionally, the controller's dynamics are plotted, and corresponding stability metrics are calculated.

5 A Case Study

In this section, WAMC in a smart grid is considered as a case study to evaluate the performance of this simulation environment.

5.1 WAMC of China Southern Power Grid

Synchronized PMU-based WAMC is an important function in the current power grid[22]. Compared with a traditional SCADA system, a PMU-based system can provide more detailed data on the power grid in real time[23]. As the sampling interval becomes smaller (in milliseconds), the requirements on system delay become more stringent. Additionally, real-time data are transmitted through the communication network to a PDC for processing[15], and the control data generated by the PDC are then sent to the end devices. Hence, the performance of a communication network plays a vital role in the WAMC.

In this case study, an actual WAMC system of the China Southern Power Grid (CSG)[11], shown in Fig. 8, is modeled in our cyber-physical simulation environment. The red dots indicate seven locations that can be divided into two regions, as shown below. Each location sends data to the control center, marked in blue lattice. The controllers are coordinated to damp the two dominant inter-area oscillation modes.

A disturbance that will cause a frequency oscillation between regions 1 and 2 is introduced into the system during the power system simulation. Obtaining control over this low-frequency oscillation is the primary focus. After data are collected from PMUs in each region, the data are sent to the control center to calculate their corresponding frequencies. A wide-area damping controller is designed to control the low-frequency oscillation. The controller output is used to control the two regions to determine the frequency control strategy. Control feedback data are then distributed to each substation from the control center.

Fig. 8 WAMC system of the China Southern Power Grid.

According to Ref. [9], a typical PMU data-sampling rate is 30 ms per sample, and the packet size is 68 bytes per sample. In this experiment, the sampling rate was set to 10 ms, and the data packet size was 60 bytes. We investigated the effects of communication delays with different means and variances on system stability. We used the quadratic performance as defined in the following equation to evaluate the system's dynamic stability:

$$\gamma = \int_0^{+\infty} s^2 \, \mathrm{d}t,$$

where s is defined as the frequency deviation between the two different regions. The quadratic performance can reflect the overall deviation after the disturbance is introduced into the power system.

5.2 Communication delay measurement and modeling

The actual measurements of the WAMC communication delay are illustrated in Fig. 9, which includes 60 000 samples over a period of 60 s.

The delay varies from 56 ms to 90 ms with a mean 68.35 ms and a variance of 11 ms. Most delays are from 60 ms to 80 ms. To test the performance of different methods with increases in delays, the normal distribution was utilized to generate delay data with means in a range of 50-130 ms. The performances of different data compensation methods were also evaluated under different mean values of delay with variances ranging from 1 ms to 15 ms.

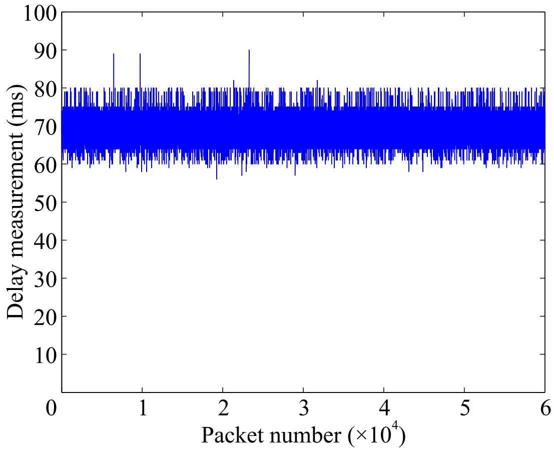


Fig. 9 Communication delay measurement from the actual WAMC system.

5.3 Experimental results

The experimental results are depicted in Figs. 10-14, where different communication delays, variances, and data compensation methods lead to different system performances. According to its definition, a higher γ indicates a worse performance.

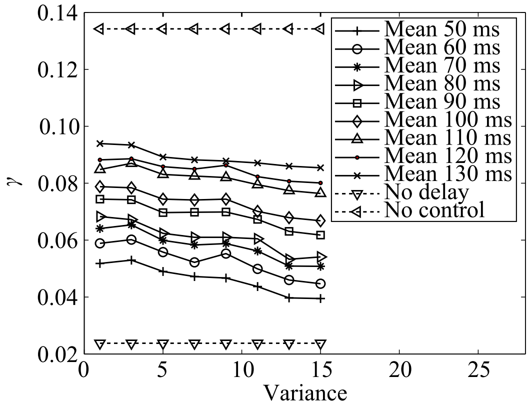


Fig. 10 Experimental results with SCH method.

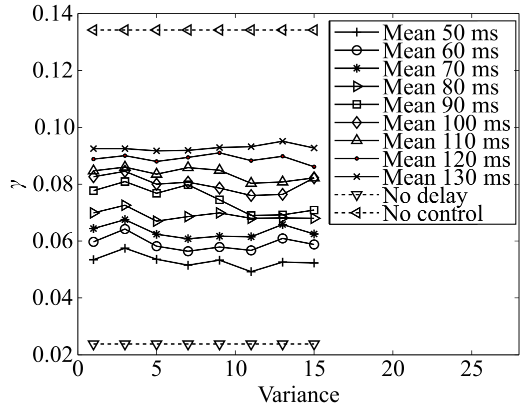


Fig. 11 Experimental results with FCC method.

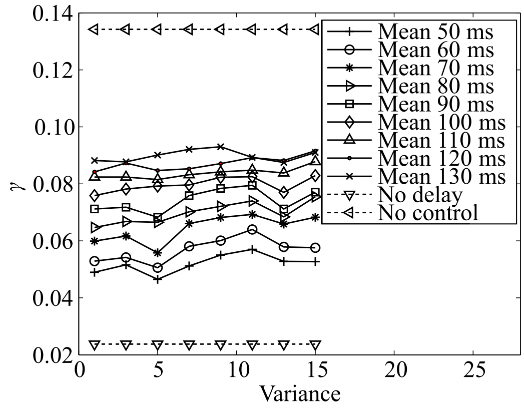


Fig. 12 Experimental results with SCC method.

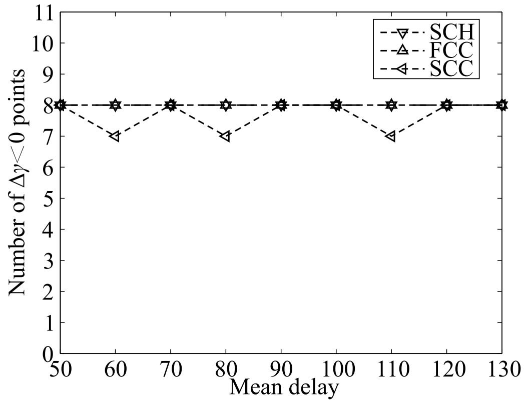


Fig. 13 Performance improvements with different methods.

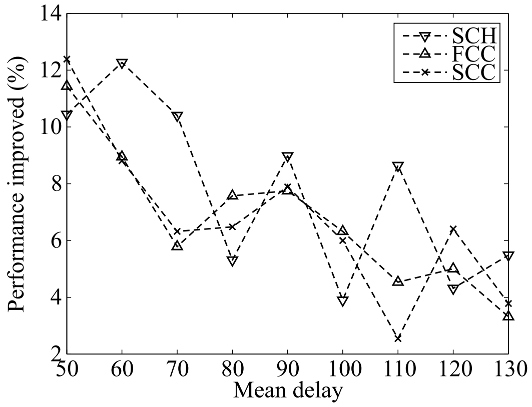


Fig. 14 Average performance improvements with mean delay increase.

As Figs. 10-12 show, system performance with WAMC presents a dramatic improvement over the performance of no control scenarios, especially when no communication delay is involved. With increased communication delays, the system performance becomes worse for all the data compensation methods.

We have also investigated the system performance with WAMC compared to the traditional synchronization framework. In the traditional method, a synchronization point is required between the network and the power grid simulators. As previously mentioned, a simulation error will occur during this synchronization. This may delay the data transmission from the power grid to the controller, and the feedback data would subsequently be delayed as well. These events will lead to a worse performance in the WAMC experiments. The system performance was assessed with mean delay values varying from 50 ms to 130 ms. For each mean value, we tested the variance from 1 ms to 15 ms with steps of two, i.e., a total of eight points.

As Fig. 13 shows, the performance of the designed system can be effectively improved in most cases. Figure 14 illustrates the average percentage of improvement in system performance compared to the traditional framework. With increases in mean delays, the difference between methods becomes less significant. It is reasonable to assume that the system performance of WAMC will drop in response to the increased delay and, therefore, that the correction of simulation errors will exert less influence on the system.

This experiment resulted in some additional findings. It can be observed that the system performances of different methods vary with increased variances in the delay. Most methods lead to better system performance when the delay variance increases within a certain range. The overall variation in system performance within a single method is consistent when the mean delay changes. This means that performance under different delay variances is more dependent on the method itself than on the mean delay. If the variance of delay has little impact on system performance, the same characteristics may be revealed with various values of mean delay.

When the delay variance is small, methods requiring no data synchronization, such as SCC, exhibit a better performance. However, with an increased delay variance, methods such as FCC with synchronization perform better. This is because when the delay variance is small, the input data are likely to be already synchronized, and interpolation estimations in the synchronization step may lead to errors. However, when the delay variance increases, the input data are more random, and data synchronization becomes necessary. This result indicates that data compensation methods should depend on different network delay scenarios.

We conclude from these experimental results that the proposed simulation method provides better performance than the traditional synchronization framework. Data compensation methods play an important role in WAMC systems, which are important parts of smart grids. Data compensation methods should be chosen according to network delay variations. To identify the best method in an unstable network environment with a large variance in delays, the performance should be tested under a particular network delay mean.

6 Conclusions

Cyber-physical systems focus on close interactions between cyberspace and the physical world. The impact of cyber-systems (e.g., communication networks) on physical systems (e.g., power grid systems) can be investigated via simulations. This study is a preliminary attempt to develop an integrated cyber-physical simulation environment, especially for smart grid applications. The implementation uses communication network simulation, power grid simulation, and WAMC control simulation. An existing well-developed and improved power system simulator, TH-STBLT, was adopted in this work.

The difference between our proposed simulation framework and others is that our framework solves the synchronization problem between discrete event simulation and continuous simulation by running the two simulators asynchronously. However, because the two simulation tools run asynchronously in the proposed framework, it will be difficult to apply it in real-time simulations for online application. This simulation framework can be used in the assessment of any power grid system designs. In our experiment, this simulation platform was tested using the WAMC system in a smart grid as a case study. Several data compensation methods were developed and investigated under different network delay conditions with or without data synchronization. The experimental results are useful toward WAMC system design and implementation.

The network model in this experiment is simple, as only a normal distribution is used to generate the delay means. In future studies, more network models will be investigated and integrated into this environment. Furthermore, the existing WAMC for CSG, with hundreds of nodes, will be added and investigated in detail in this simulation environment before its actual operation.

Acknowledgements

This work was supported in part by the National Key Basic Research and Development (973) Program of China (Nos. 2013CB228206 and 2011CB302505), the National Natural Science Foundation of China (No. 61233016), and 2012 State Grid S&T project, Advanced Study of Power Quality—Key Technologies and Applications.

References

[1] E. A. Lee, Cyber physical systems: Design challenges, in *Proc. 11th IEEE Int. Symp. on Object Oriented Real-Time Distributed Computing (ISORC)*, Orlando, USA, 2008, pp. 363-369.

[2] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, Cyber-physical systems: A new frontier, in *Proc. IEEE Int. Conf. on Sensor Networks, Ubiquitous and Trustworthy Computing*, Taichung, China, 2008, pp. 1-9.

[3] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, Cyber-physical systems: The next computing revolution, in *Proc. 47th ACM/IEEE Design Automation Conf. (DAC)*, Anaheim, USA, 2010, pp. 731-736.

[4] M. D. Ilic, X. Le, U. A. Khan, and J. M. F. Moura, Modeling future cyber-physical energy systems, in *Proc. IEEE Power & Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, USA, 2008, pp. 1-9.

[5] K. Moslehi and R. Kumar, A reliability perspective of the smart grid, *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 57-64, 2010.

[6] H. Farhangi, The path of the smart grid, *IEEE Power & Energy Magazine*, vol. 8, no. 1, pp. 18-28, 2010.

[7] H. Li, L. Lai, and W. Zhang, Communication requirement for reliable and secure state estimation and control in smart grid, *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 476-486, 2011.

[8] G. N. Ericsson, Cyber security and power system communication—Essential parts of a smart grid infrastructure, *IEEE Trans. Power Delivery,* vol. 25, no. 3, pp. 1501-1507, 2010.

[9] S. Zhang, L. Tong, X. Liang, and C. Hong, A novel real-time hybrid simulator base on two personal computers and analog interfaces for data exchange, in *Proc. IEEE Power & Energy Society General Meeting*, Calgary, Canada, 2009, pp. 1-7.

[10] The network simulator—ns-2, http://www.isi.edu/nsnam/ns/, 2014.

[11] C. Lu, X. Wu, J. Wu, P. Li, Y. Han, and L. Li, Implementations and experiences of wide-area HVDC damping control in China southern power grid, in *Proc. IEEE Power & Energy Society General Meeting*, San Diego, USA, 2012, pp. 1-7.

[12] J. Cao, Y. Wan, G. Tu, S. Zhang, A. Xia, X. Liu, Z. Chen, and C. Lu, Information system architecture for smart grids, (in Chinese), *Chinese J. of Computers*, vol. 36, no. 1, pp. 143-167, 2013.

[13] M. Chenine, I. Al Khatib, J. Ivanovski, V. Maden, and L. Nordstrom, PMU traffic shaping in IP-based wide area communication, in *Proc. 5th Int. Conf. on Critical Infrastructure*, Beijing, China, 2010, pp. 1-6.

[14] A. G. Phadke and J. S. Thorp, Communication needs for wide area measurement applications, in *Proc. 5th Int. Conf. on Critical Infrastructure*, Beijing, China, 2010, pp. 1-7.

[15] M. Chenine, E. Karam, and L. Nordstrm, Modeling and simulation of wide area monitoring and control systems in IP-based networks, in *Proc. IEEE Power & Energy Society General Meeting*, Calgary, Canada, 2009, pp. 1-8.

[16] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components, *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 548-558, 2006.

[17] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, Integrated hybrid-simulation of electric power and communications systems, in *Proc. IEEE Power Engineering Society General Meeting*, Tampa, USA, 2007, pp. 1-8.

[18] T. Godfrey, S. Mullen, R. C. Dugan, C. Rodine, D. W. Griffith, and N. Golmie, Modeling smart grid applications with co-simulation, in *Proc. 1st IEEE Int. Conf. on Smart Grid Communications*, Gaithersburg, USA, 2010, pp. 291-296.

[19] V. Liberatore and A. Al-Hammouri, Smart grid communication and co-simulation, in *Proc. IEEE Energytech*, Cleveland, USA, 2011, pp. 1-5.

[20] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, Power system and communication network co-simulation for smart grid applications, in *Proc. IEEE Power & Energy Society General Meeting—Innovative Smart Grid Technologies (ISGT)*, Anaheim, USA, 2011, pp. 1-6.

[21] H. Li, L. Tong, and X. Xie, A method of fast stability simulation for online transient pre-decision, in *Proc. IEEE/PES Asia and Pacific Transmission and Distribution Conf. & Exhi.*, Dalian, China, 2005, pp. 1-4.

[22] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, Synchronized phasor measurement applications in power systems, *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 20-27, 2010.

[23] A. Mao, J. Yu, and Z. Guo, PMU placement and data processing in WAMS that complements SCADA, in *Proc. IEEE Power & Energy Society General Meeting*, San Francisco, USA, 2005, pp. 780-783.

Yuxin Wan is currently a PhD candidate of Department of Automation, Tsinghua University, China. He received the BEng degree from Tsinghua University in 2008. His research interests include cyber-physical systems and smart grid applications.

Junwei Cao is currently a professor and deputy director of Research Institute of Information Technology, Tsinghua University, China. He is also the director of Open Platform and Technology Division, Tsinghua National Laboratory for Information Science and Technology. His research interests include advanced computing technology and applications. Before joining Tsinghua in 2006, Junwei Cao was a research scientist of Massachusetts Institute of Technology, USA. Before that he worked as a research staff member of NEC Europe Ltd., Germany. Junwei Cao got his PhD degree in computer science from University of Warwick, UK in 2001. He got his MEng and BEng degrees from Tsinghua University in 1998 and 1996, respectively. Junwei Cao has published over 130 academic papers and books, cited by international researchers for over 3000 times. Junwei Cao is a senior member of the IEEE Computer Society and the member of the ACM and CCF.

Shuqing Zhang is currently a research assistant with the Department of Electrical Engineering, Tsinghua University, China. He received his BEng and PhD degrees from Tsinghua University in 2005 and 2010, respectively. His research interests include sub-synchronous resonance in power system, electro-magnetic/electrical-mechanical hybrid real-time simulation, IoT applied in powersystem, and power load modelling as well as identification of dominant parameters.

Guoyu Tu is with Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, China. She received the BEng and PhD degrees from Tsinghua University in 2002 and 2010, respectively. Her research interests include data mining, stochastic scheduling, and optimization for large scale systems including smart grid, manufacturing, and service network.

Chao Lu is currently an assistant professor with the Department of Electrical Engineering, Tsinghua University, China. He received the BEng and PhD degrees from Tsinghua University in 1999 and 2005, respectively. His research interests include power system analysis and control, and intelligent control applications.

Xingtao Xu is the director of Electric Power Quality Research Center, JiangSu Electric Power Company, senior engineer, expert enjoying special allowance of the State Council. His research interests include theoretical research and engineering application of electrical energy quality monitoring and governance, reactive voltage optimization and control field. He presided over the State Grid, provincial, municipal science and technology projects of China, and has won 8 national invention patents, 26 science and technology awards, including “the supply network reactive voltage optimization centralized control system” which won the second prize of national scientific and technological progress award.

Keqin Li is a SUNY distinguished professor of computer science at State University of New York. He is also an intellectual ventures endowed visiting chair professor at Tsinghua University, China. Keqin Li received his BEng degree in computer science from Tsinghua University, China in 1985, and PhD degree in computer science from the University of Houston, USA in 1990. He was an assistant professor (1990-1996), an associate professor (1996-1999), a full professor (1999-2009), and has been a SUNY distinguished professor of computer science since 2009 in State University of New York at New Paltz. He was the acting chair of Department of Computer Science during Spring 2004. His research interests are mainly in design and analysis of algorithms, parallel and distributed computing, and computer networking. He has over 285 refereed research publications. He is currently or has served on the editorial board of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *Journal of Parallel and Distributed Computing*, *International Journal of Parallel, Emergent and Distributed Systems*, *International Journal of High Performance Computing and Networking*, *International Journal of Big Data Intelligence*, and *Optimization Letters*.