A Real-time Transient Stability Simulation Tool for Large-scale Power Systems

Vahid Jalili-Marandi, Member, IEEE, Eric Robert, Vincent Lapointe, Member, IEEE, Jean Bélanger, Member, IEEE

Abstract—Development of the Phasor tool for real-time transient stability simulation in large-scale power systems is presented in this paper. This tool can be used for performing contingency studies, testing control devices, as well as training purposes in academia and industry. The Phasor tool includes a model library that is extensible based on the user requirements. The accuracy of the tool has been evaluated in comparison to other commercial but non-real-time transient stability simulation packages, and its real-time performance has been tested on an eMEGAsim real-time simulator for large-scale power systems in the order of 10000 buses, 2500 generators, and over 4500 control devices.

Index Terms—Large-scale systems, power system simulation, power system transient stability, real-time simulation.

I. INTRODUCTION

The need for real-time assessment of transient stability analysis was highlighted by the recent black out events in the USA (August 14, 2003) and Europe (November 4, 2006). Based on an analysis of the contributing factors to these events, deficiencies in real-time security assessment and system monitoring were identified as the most important issues [1]. Transient stability simulation is the key component in control and dynamic security assessment tools, and the ultimate goal in this field has always been to perform these simulations as fast as real-time for realistic power systems. However, mathematical modeling and numerical solution of transient stability phenomena is computationally onerous for large-scale systems [2], and on the other hand the continuous growth in electricity demand as well as the consequent expansion of power grids creates new and complex problems. Thus, power engineers are exploring novel methods for quick and efficient simulations of large-scale systems [3], [4].

Transient instability in power systems has been extensively studied since the 1920s and a lot of knowledge and experience is available in the literature [2]. There are also many commercial software packages such as PSS/E, EUROSTAG, ETAP, DlgSILENT that are widely used to analyze systems for their transient stability behaviour [5]. Despite of the all-in-one philosophy that exists in these tools, they are not designed for real-time and hardware-in-the-loop (HIL) simulation. From the other side, available commercial digital real-time simulators such as eMEGAsim [6], RTDS [7], and Hypersim [8] address real-time and HIL simulation, but they are originally designed and built for electromagnetic transient studies, not for transient stability simulations. Of course, that is possible to perform transient stability simulation with these tools by using simpler models, multiple processors, and larger time-steps. However, the question that arises is whether or not this practice is the most efficient, accurate, and cost-effective simulation approach [9].

eMEGAsim is a real-time simulator based on commercial-off-the-shelf computer components [10]. It is used by manufacturers and research institutions to simulate complex power grid and power electronic devices in real-time for the HIL applications. The simulator exploits parallel and distributed computational architectures on multi-core processors and PC-clusters. The time-step of the model running on eMEGAsim can be in the range of hundreds of nanoseconds to typically 50µs, which is suitable for electromagnetic transient studies. However, this small time-step is inefficient for transient stability simulations of realistic-size power systems.

The objective of this paper is to report the recent advancements in the development of the Phasor tool as a new solution added to eMEGAsim simulator. The Phasor tool is designed specifically for real-time transient stability simulation of large-scale power systems. The core of the tool is based on the phasor-domain solution of power systems. This simulation package can be used to perform dynamic security assessments and test the functionality of hardware such as global control devices in large-scale power systems. Furthermore, it can be utilized for training purposes in the academic laboratories or for industry level operators. Although, eMEGAsim is a parallel multi-core based simulator, the reported implementation of the Phasor tool is designed to use only one CPU core. The tool is linked with MATLAB/SIMULINK library, and it is compatible with both Linux and Windows operating systems. In addition to its own user data interface, load flow and dynamic data files from PTI’s PSS/E are also importable. Table I briefly introduces general features of the Phasor tool.

The paper is organized as follows: in Section II, there is a brief overview of the transient stability study in power systems and the numerical methods used in the Phasor tool. The concept of the real-time simulation, the simulator architecture, and challenges for implementation of the Phasor tool for real-time simulation are described in Section III. The user interface and library of the Phasor tool are explained in Section IV. Experimental real-time simulation results on several test cases, accuracy validation, performance evaluation, and a discussion of the results are in Section V. Section VI presents the conclusion.
sparse matrix methods have been exploited efficiently to Modified Euler integration method have been used. Moreover, In the Phasor tool the partitioned approach along with the integration methods.

Equations, and then this set is lumped into the (2) resulting in an implicit integration method converts (1) into a set of algebraic equations, which is solved for the algebraic variables, and these solutions are partitioned and simultaneous approaches [11]. In the system, while equation (2) describes the network and constraints on (1).

Solution of these equations requires employing numerical integration methods that discretize equation (1). The overall solution approach can be classified into two groups, namely, partitioned and simultaneous approaches [11]. In the partitioned solution approach the differential equation set (1) is solved separately for the state variables and the equation set (2) is solved for the algebraic variables, and these solutions are then alternated. In the simultaneous approaches, however, an implicit integration method converts (1) into a set of algebraic equations, and then this set is lumped into the (2) resulting in a larger set of algebraic equations including all the variables. In each of these approaches one can use various types of integration methods.

In the Phasor tool the partitioned approach along with the Modified Euler integration method have been used. Moreover, sparse matrix methods have been exploited efficiently to factorize and solve network nodal equations.

### III. REAL-TIME TRANSIENT STABILITY SIMULATION

#### A. Concept of Real-time Simulation

The term “real-time” has been traditionally used by the computer industry to refer to interactive systems where the computer response is sufficiently fast enough to satisfy human users. In the case of power systems, real-time simulation implies that the computer must solve the model equations and output the results for interface purposes within the model integration time-step [12]. In general, digital real-time simulation may be defined as a faithful reproduction of output waveforms, by combining systems of hardware and software, which would be identical to the waveforms or effects produced by the real power system being modeled.

Depending on the time taken by the computer to complete the computation for one time-step two situations can be assumed. As shown in Fig. 1(a), if the execution time, \( T_e \) (which includes both the computation and the data exchange), for the simulation of any time-step is smaller or equal to the time-step used, the simulation is called real-time. On the other hand, as shown in Fig. 1(b), if \( T_e \) for any time-step is greater than its time-step execution time overruns take place. In this case the simulation is non-real-time or offline. If such a situation is observed, to fit the simulation in one time-step either the step-size must be increased or the numerical complexity of the model must be reduced.

Hardware-In-the-Loop (HIL) simulation refers to a simulation, where parts of the fully digital real-time simulation, such as control or protection systems, are made with actual physical components. In this case, the simulation proceeds with the device-under-test connected through input and output interfaces such as converters and signal conditioners. The simulation can also be modified with the user defined control inputs, for instance closing or opening of switches to connect or disconnect the components in the simulated power system. HIL simulation minimizes the cost of investment, and it improves the design and test of control and protection devices through the use of prototypes once the system model is established with the help of fully digital real-time simulation. This type of simulation is also advantageous where it is practically impossible to analyze the interactions between a device and the system under several operating conditions such as severe faults. For example, power system engineers can use HIL simulation to verify the safe operation of newly designed control device before installing it in the power grid, or to find and resolve the erroneous operation of an existing device.

#### B. Real-time Simulator Architecture

The OPAL-RT’s eMEGAsim is a PC-based simulator comprised of two groups of computers known as target nodes, and hosts. Target nodes are the computational cores that carry out the simulation, and each of them is powered by the modern high-performance distributed supercomputer technology found in off-the-shelf INTEL or AMD multi-core processors. However, the host is a computer that the user can utilize to develop, design, and evaluate a model in offline mode. The host computer also provides the interface between the user and the target nodes. High-speed communication links connect multiple targets to each other, as well as hosts and targets. External hardware can also be connected to the simulator via the FPGA-based (Field-Programmable Gate Array) analog/digital inputs/outputs. The processors, i.e. CPUs, in one target communicate with each other through shared memory.

The targets are also capable of eXtreme High Performance (XHP) mode execution, in which one core is dedicated entirely to run real-time operating system tasks and schedulers, while other cores perform the computations.
Running a model in the XHP mode ensures that the simulation is in real-time; otherwise the simulator reports the number of overruns where the computation could not fit in the assigned fixed time-step.

C. Practical Challenges

Taking a code that performs well in offline simulation, and making it work in real-time simulation creates several types of challenges. Although the majority of the difficulties that appeared in developing the Phasor tool were related to computational issues, they are of strong interest in the power systems community. The primary challenges are:

1) Coalesced data pattern: Implementing the transient stability code in real-time requires very efficient cache usage, memory management, and data storage. Basically, for real-time simulation there should be no system calls during the simulation. Thus, the amount and pattern of memory access must be kept as low and consistent as possible. This is not a serious issue where the system size is relatively small and all the required data is automatically stored in cache. However, the limitation shows up in the case of large-scale systems where the amount of input data is too large to be stored in cache. In Phasor tool for each component of the system there is a data screen to save only the variables and parameters that are strictly needed during run-time in a coalesced pattern. Thus, with a memory layout that maximizes coalescence we minimize the amount of memory accesses.

2) Contiguous memory pattern: In transient stability simulation the order of updates of component that have state variables is important. For example, if a synchronous generator (SG) is equipped with a power system stabilizer (PSS), at each time-step, the states of the PSS must be updated based on the values of the last time-step to find out the reference voltage for the excitation system for the current time-step, and then the SG model can be updated. This sequential order implies the use of a contiguous memory allocation pattern for sequential access for the Phasor tool instead of a random memory arrangement. The contiguous pattern also helps to maximize the use of the memory bandwidth as it activates streaming via the automatic hardware pre-fetching process of the CPU.

3) Pre-calculation: Another challenge for real-time operation of the Phasor tool appears where the topology of the power system changes (e.g. when a transmission line is tripped to be out of service). These cases change the network admittance matrix, and consequently the LU factors must be recalculated or modified for the new topology. Depending on the system size, these calculations can be a bottleneck for real-time simulation. Therefore, to avoid overruns in large-scale systems a pre-calculation feature is added to the Phasor tool. With this feature the user can define a scenario of events that might happen in the system, by declaring the location, type, and time of the event. This scenario is evaluated and the LU factors for all of the events that cause topology changes are pre-calculated and stored before starting the real-time simulation. The type of event can be:

- Bus fault
- Load shedding and load restoration
- Actions for circuit breaker trip and reclose

IV. PHASOR TOOL USER INTERFACE

A. Data Entry

The Phasor tool is built as a MATLAB/SIMULINK S-function, and a graphical mask is designed to facilitate interactions between user and the tool. However, the core of the interface is a data file that lists all the components existing in the system, the parameters, and initial values. A Microsoft Excel workbook is designed that makes it possible for the user to enter the data of the network. As an advanced feature, the PTI’s PSS/E description of the system, the power flow raw data file and dynamic files (i.e. *.raw and *.dyr), can also be used as the input data files for both offline and real-time simulation.

B. Models Library

The library of the Phasor tool includes most fundamental components required for power system modeling. The library contains the following models: synchronous generator, load, excitation system, power system stabilizer, turbine and governor, two-winding transformer, and transmission line. The details of each item are described in the Appendix. Currently, except for the synchronous generator, the library provides one type of model for each item. For the synchronous generator, however, the library includes two types of models. One is the so called classical model that only describes the mechanical behavior of machine, and the second one is the 6th order detailed model. The detail model, which allows modeling of the mechanical power and voltage controllers, includes the transient and sub-transient reactance as well as amortisseur windings.

The Phasor tool is based on a C++ template that makes its library extensible. Its architecture is flexible either to register a new type of model to each existing item (e.g. a new type of excitation system or a new model of synchronous generator) or to build and add a new item to the present tool (e.g. a three-winding transformer or a wind turbine). Therefore, if the user wishes to model a component that does not exist in the library, the new component can be integrated to the solver by defining specific interfacing functions and modules.
V. EXPERIMENTAL RESULTS

In this section we demonstrate results of offline and real time simulations using the Phasor tool. The accuracy of the simulation has been verified using the PSS/E software from Siemens Energy Inc.

A. Test Systems

As shown in Fig. 2, several test systems of increasing size were constructed to explore the efficiency of the Phasor tool. Table II lists the specifications of each test system in terms of the number of its components. The Scale 1 system is the IEEE’s New England test system with 10 generators and 39 buses. This system was duplicated several times and interconnected via an appropriate number of transmission lines to create larger systems that are highly meshed. Thus test systems with Scale of 2, 4, 8, 16, 32, 64, 128, 180, and 256 were obtained. The Scale 180 system was constructed based on the maximum real-time compute capacity of the current target computer hardware. In Table II the column “Controller” lists the total number of control devices (including excitation system, power system stabilizer, and turbine and governor), and the column indicated by “Other” gives the total number of loads, transmission lines, and transformers in the system.

B. Accuracy Validation

The accuracy of the Phasor tool was validated using PTI’s PSS/E software program. The case study used in this section is the Scale 1 test system whose one-line diagram is shown in Fig. 2. Several fault locations have been tested and the results were compared with those of PSS/E. In all cases results from the Phasor tool match the PSS/E results very well. To demonstrate a sample of simulation, a three-phase fault happens at Bus 5, at time 3s and it is cleared after 100ms. The voltage magnitude of some buses obtained from the Phasor tool is presented in Fig. 3. For comparison PSS/E results are superimposed in this figure. As can be seen the Phasor tool is completely stable during the steady-state of the system, i.e., t < 3s. During the transient state and also after the fault is cleared, the program results closely follow the results from PSS/E. The maximum discrepancy between voltage magnitudes from Phasor tool and the PSS/E simulation was found to be 0.11%, using the error-measure defined in (4):

$$
\varepsilon = \max \left( \frac{V_{\text{PSS/E}} - V_{\text{Phasor}}}{V_{\text{PSS/E}}} \right)
$$

(4)

Fig. 2. Construction of test systems for transient stability simulations. The Scale 1 system shows the one-line diagram of the IEEE’s New England Test System.
### C. Performance Evaluation

To demonstrate the performance of the Phasor tool for real-time simulation, the test systems described in Table II were used to perform three types of simulation:

- Offline with Windows OS
- Offline with Linux OS
- Real-time with Linux OS (XHP mode)

In this experiment Windows OS runs on a 2.5GHz INTEL CPU, whereas Linux OS runs on a 3.3GHz INTEL CPU. Although CPUs in both cases are multi-core, the Phasor tool, as mentioned before, is programmed to utilize only one CPU core. The scenario of the simulation is as described in the accuracy validation test. The last three columns of Table II summarize the results of these simulations. The tests were run for 10sec and the step-size was set to 10ms for all systems. The table shows that all systems up to Scale 180 run in real-time, but there are overruns for Scale 256. When the step-size is increased to 15ms, the Scale 256 also runs successfully in real-time.

An important observation that one can make from these simulation results is the scalability behaviour of the Phasor tool. Basically, scalability is defined for parallel processing based algorithms, and it reveals how efficient the parallel multi-processor is in comparison with the single processor simulation. As mentioned before, the Phasor tool is designed to utilize only one core of a multi-core CPU, so the parallel processing does not have any meaning for this application. However, the 3rd column of Table II shows that as the system size increases the simulation time increases linearly. This linear behaviour is promising for a parallel multi-core based implementation of the Phasor tool.

### VI. CONCLUSION

This paper presented the development of the Phasor tool for the eMEGAsim simulator to perform real-time transient stability simulation. This tool can be used for dynamic security assessments, contingency studies, functionality tests of hardware such as controllers in large-scale power systems and micro-grids, as well as for training purposes in academic laboratories or for industrial operators.

The Phasor tool is designed to be efficient for simulation of large-scale power systems. Although, the simulation results demonstrated that the Phasor tool is effective for real-time simulation of systems in the order of 10000 buses, ongoing research at OPAL-RT Technologies is aimed at real-time simulation for systems in the size of 20000 buses. Moreover, the library of the tool is extending to include suitable models for other components found in power grids such as static var compensators and HVDC lines.

### VII. APPENDIX

The library of Phasor tool is easily extensible, as the user can define a new model and register it to the library. Currently, the library contains models for: synchronous generator, load, excitation system, power system stabilizer, turbine-governor, transformer, and transmission lines. The details of these models are described in this section.

#### A. Synchronous Generator

The Phasor tool includes two models for synchronous generators. Equations (5) to (10) describe Park’s equations with an individual dq reference frame fixed on the generator’s field winding [14]:

\[
\delta(t) = \omega_R \cdot \Delta \phi(t) \\
\Delta \phi(t) = 0.5 \mu |T_e|^2 + T_m(t) - D \Delta \omega(t) \\
\psi_{fd}'(t) = \omega_R [e_{fd}'(t) - R_{fd}' f_{fd}'(t)] \\
\psi_{1d}'(t) = -\omega_R R_{1d} i_{1d}'(t) \\
\psi_{1q}'(t) = -\omega_R R_{1q} i_{1q}'(t) \\
\psi_{2q}'(t) = -\omega_R R_{2q} i_{2q}'(t)
\]

where \( \omega_R, \mu, H, D, R_{fd}, R_{1d}, R_{1q}, R_{2q} \) are constant system parameters whose definition can be found in [14]. The classical model of the synchronous generator includes only equations (5) and (6) where mechanical power is constant at its steady state value in (6).
B. Load
All loads are modeled as constant impedances in the network equations.

C. Excitation system
A simplified model based on IEEE Type AC4 excitation system is implemented.

D. Power System Stabilizer
A speed sensitive stabilizer is modeled based on the following block diagram.

E. Turbine and Governor
The steam turbine-governor is modeled based on the following block diagram.

F. Transmission line
The PI model is used for transmission lines.

G. Transformer
Transformer is modeled with its positive sequence impedance.

VIII. ACKNOWLEDGMENT
The authors gratefully acknowledge Dr Jean Mahrseredjian from École Polytechnique de Montréal for making it possible for us to perform accuracy validation experiments.

IX. REFERENCES

X. BIOGRAPHIES

Vahid Jalili-Marandi (S’06-M’11) received his Ph.D. degree in energy systems from the University of Alberta, Edmonton, AB, Canada, in 2010. Currently, he is with OPAL-RT Technologies, Montréal, as a simulation specialist in power system applications. His research interests include transient stability simulation methods, large-scale power systems simulation, real-time simulation, parallel processing, and general purpose computing on GPUs.

Eric Robert is the software architect at OPAL-RT Technologies. He received his B.Eng. degree from École Polytechnique de Montréal, QC, Canada, in 1999. He worked several years at Ubisoft Montréal notably as a technical architect and as a lead programmer. Since 2010, he joined the R&D team to work on the development of the real-time simulator RT-LAB.

Vincent Lapointe (M’11) is a simulation software specialist at OPAL-RT Technologies. He received his M.A.Sc. degree from Laval University, Quebec City, QC, Canada, in 2000. Since 1997, he has worked on the development of the real-time simulator RT-LAB. He is also the team leader of the Electromechanical Systems R&D team.

Jean Belanger (M’87) received the B.S. degree in electrical engineering from Laval University, Quebec City, QC, Canada, in 1971, and the M.S. degree from the École Polytechnique de Montréal, Montreal, QC, Canada. He is with OPAL-RT Technologies, Montréal, where he is the co-founder, CEO, and CTO. Founded in 1997, OPAL-RT Technologies develops and commercializes digital real-time simulators for the design of systems and the testing of electronic controllers. Under his direction and technological leadership, OPAL-RT became a well-known developer of state-of-the-art real-time simulators capable of simulating all sorts of mechanical and electrical systems, including the fastest power electronic converters used in a wide range of industries—from hybrid vehicles to entirely electrical-driven aircraft, and from microgrids to very large ac/dc power systems.