IEEE Workshop on Modeling and Simulation of Cyber-Physical Energy Systems 2013

May 20 2013, Berkeley CA

Co-located with the Fourth ACM International Conference on Future Energy Systems
Technically co-sponsored by the IEEE Industrial Electronics Society

# MSCPES 2013 PROCEEDINGS

Photo © Rich Niewiroski Jr.

IEEE MSCPES 2013

## Message from the chairs

Cyber-physical energy systems (CPES) use computers and networks to orchestrate energy production, distribution, and consumption. Such systems challenge existing engineering methods for control systems design, software engineering, and networking systems because of their heterogeneity, very large scale, safety criticality, and vulnerability to security risks. The behavior of such systems is also strongly affected by markets and regulation, factors that don't always manifest easily in engineering design.

This workshop, the first of its kind to our knowledge, germinated through discussions between the Austrian Institute of Technology and UC Berkeley centered on identifying the best modeling and simulation tools for CPES. It quickly became clear that no comprehensive off-the-shelf solution exists, but also that a surprisingly diverse group of people and organizations are working on the problem and developing pieces of the puzzle. It became clear that it was time to galvanize a community through a workshop, and the Fourth ACM International Conference on Future Energy Systems seemed like the obvious home for such a workshop. We pulled together an outstanding group of professionals to serve on the program committee, secured IEEE technical co-sponsorship from the IEEE Industrial Electronics Society and its technical committee on Smart Grids, issued a call for papers, and received an unexpectedly strong and diverse group of submissions. We expect that the resulting workshop will launch a vibrant community with common interests that will make the workshop annual event while the problem area remains open. And that an outcome of the workshop will be the preparation and submission of best-of-class papers for an open special section on this topic in the IEEE Transactions on Industrial Informatics.

We thank the program committee for their excellent service and, most importantly, we thank the authors and presenters for their thoughtful contributions to the advancement of this important field.

*Peter Palensky (Austrian Institute of Technology)*
*Edward A. Lee (University of California Berkeley)*

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

**IEEE MSCPES 2013**

Berkeley UNIVERSITY OF CALIFORNIA

IEEE MSCPES 2013

# Conference Organizers

**General Chair:** Edward A. Lee, University of California Berkeley, USA

**Program Chair:** Peter Palensky, Austrian Institute of Technology, Austria

**Program Committee**

Christoph Grimm, University Kaiserslautern, Germany

Michael Wetter, LBNL, USA

Sven Christian Mueller, TU Dortmund, Germany

Hiroaki Nishi, Keio University, Japan

Seung Ho Hong, Hanyang University, Korea

Wolfgang Kastner, TU Vienna, Austria

Wilfried Elmenreich, University Klagenfurt, Austria

Pierluigi Siano, University Salerno, Italy

Jeannie Falcon, National Instruments, USA

Kishor S. Trivedi, Duke University, USA

Jenny Yan Liu, Concordia University Montreal, Canada

**Venue**

The Ubiquitous Swarm Lab at UC Berkeley  
490 Cory Hall (Hearst Avenue at Gayley Road)  
Room 490H (4th floor)  
Berkeley, CA 94720

# Program

| | |
|---|---|
| 09:00-9:45 | **Opening and Keynote**<br>Kyle Anderson, Jimmy Du, Amit Narayan and Abbas El Gamal: *GridSpice: A Distributed Simulation Platform for the Smart Grid* |
| 09:45-10:00 | **Coffee** |
| 10:00-12:00 | **Session 1**<br>Ayan Banerjee, Joydeep Banerjee, Georgios Varsamopoulos, Zahra Abbasi and Sandeep Gupta: *Hybrid Simulation of CyberPhysical Energy Systems*<br>Yi Deng, Hua Lin, Sandeep Shukla, James Thorp and Lamine Mili: *Co-Simulating Power Systems and Communication Network for Accurate Modeling and Simulation of PMU based Wide Area Measurement Systems using a Global Event Scheduling Technique*<br>Jason Fuller, Selim Ciraci, Jeff Daily, Andrew Fisher and Matthew Hauer: *Communication Simulations for Power System Applications*<br>Atiyah Elsheikh, Muhammed Usman Awais, Edmund Widl and Peter Palensky: *Modelica-enabled rapid prototyping of cyber-physical energy systems via the functional mockup interface*<br>Edmund Widl, Wolfgang Mueller, Atiyah Elsheikh, Matthias Hoertenhuber and Peter Palensky: *The FMI++ Library: A High-level Utility Package for FMI for Model Exchange*<br>Matthias Stifter, Roman Schwalbe, Filip Andrén and Thomas Strasser: *Steady-State Co-Simulation with PowerFactory* |
| 12:00-12:45 | **Lunch** |

| | |
|---|---|
| 12:45-14:45 | **Session 2**<br>Andrea Monacchi, Dominik Egarter and Wilfried Elmenreich: *Integrating Households into the Smart Grid*<br>Marco Grassi, Michele Nucci and Francesco Piazza: *Ontologies for Smart Homes and Energy Management: an Implementation-driven Survey*<br>Jibonananda Sanyal and Joshua New: *Simulation and Big Data Challenges in Tuning Building Energy Models*<br>Soumyo Chakraborty, Sandeep Shukla and James Thorp: *A Hierarchical Networked Micro-Simulator to Study Grid-Integration of Renewables and Electric Vehicles*<br>Alberto Lamadrid, Tim Mount and Ray Zimmerman: *On the Capacity value of Renewable Energy Sources in the presence of Energy Storage and Ramping Constraints*<br>Mengmeng Yu, Seung Ho Hong, Min Wei and Aidong Xu: *A Homogeneous Group Bargaining Algorithm in a Smart Grid* |
| 14:45-15:00 | **Coffee** |
| 15:00-16:40 | **Session 3**<br>Ilge Akkaya, Edward A. Lee and Patricia Derler: *Model-Based Evaluation of GPS Spoofing Attacks on Power Grid Sensors*<br>Javier Moreno Molina, Xiao Pan, Christoph Grimm and Markus Damm: *A Framework for Model-Based Design of Embedded Systems for Energy Management*<br>Usman Khan and Alex Stankovic: *Security in cyber-physical energy systems*<br>Alexander Schirrer, Oliver Koenig, Sara Ghaemi, Friederich Kupzog and Martin Kozek: *Hierarchical application of model-predictive control for efficient integration of active buildings into low voltage grids*<br>Sven Christian Müller, Ulf Häger and Christian Rehtanz: *Integrated Coordination of AC Power Flow Controllers and HVDC Transmission by a Multi-Agent System* |
| 16:40-17:30 | **Closing Keynote, Discussion**<br>C. Zhao, U. Topcu, N. Li and S. H. Low : *Power system dynamics as primal-dual algorithm for optimal load control*. (presentation only) |

# GridSpice: A Distributed Simulation Platform for the Smart Grid

Kyle Anderson, *Member, IEEE,* Jimmy Du, *Member, IEEE,* Amit Narayan, *Member, IEEE,*
and Abbas El Gamal, *Fellow, IEEE*

***Abstract*—GridSpice is a simulation framework for the smart grid that integrates existing electric power point tools. The framework provides computational scale and modeling capability to represent diverse scenarios in large interconnected grid systems. Currently, GridSpice integrates a transmission and economic dispatch package based on MatPOWER and the distribution system simulator Gridlab-D. GridSpice provides computational scale by parallelizing large simulation jobs across many virtual machines using Amazon Web Services. GridSpice provides three primary mechanisms for setting up simulations: a Python library, a browser-based user interface, and a Representational State Transfer (REST) API, which allows users to interface with existing data management systems or post-processing tools. GridSpice is available in open-source under the BSD license.**

***Index Terms*—Power System Simulation, Electric Vehicles, Multiagent Systems**

## I. Introduction

SMART grid technologies blur the traditional boundaries between generation, transmission, distribution, and end-use loads. Historically, utilities and system operators had the task of matching generation to a relatively predicable and uncontrollable load profile. With the adoption of technologies such as rooftop solar, demand response, and controllable electric vehicle chargers, operators have more options to match supply and demand at the cost of increased modeling complexity. Existing electric power simulators provide well-proven point tools for transmission networks, e.g., Siemens PSS/E [17], distribution networks, e.g., Gridlab-D [1], OpenDSS [21], and CYMDIST [16], or general optimal power flow, e.g. MATPOWER [3]. No mainstream tool supports co-simulation of multiple domains. In addition, many existing simulators run on standalone workstations, limiting their ability to handle the increasing computational demands of modern smart grid scenarios. Finally, existing tools often use proprietary input formats and exhibit dependencies on particular operating systems or 3rd party libraries, making it difficult to integrate them into existing planning and operations processes.

In this paper, we describe GridSpice, a cloud-based simulation platform that addresses the aforementioned limitations of existing simulation systems. GridSpice provides a flexible framework that runs industry standard simulation tools as separate but synchronized processes on a cluster. Each subsystem within the network model runs in a simulator designed for that purpose while GridSpice synchronizes the boundary state of these loosely coupled processes. Since GridSpice can run each of these processes on a separate node of a cluster, it is possible to simulate a transmission network with hundreds of connected generators and distribution networks on a sufficiently large cluster. Furthermore, GridSpice allows users to specify different levels of simulation granularity for different nodes in the transmission network. For example, an abstracted aggregate load forecast may be used for some load serving entities (LSEs), while running a full distribution simulation for other LSEs on the same transmission network. In addition to partitioning large interdependent networks to run on a cluster, GridSpice also makes it easy to run embarrassingly parallel tasks such as iterative grid analysis. Users can quickly evaluate the effects of many potential changes to the grid and compare the results. An example use case would be determining the ideal locations to add storage elements on the grid (i.e., the user evaluates each potential location independently in parallel).

The GridSpice framework allows users to edit models and control simulations through a Secure Representational State Transfer (REST) API [12]. Since the REST interface is based on HTTP requests, users may control the system through the language of their choice and automatically synchronize their models with energy management systems (EMS), distribution management systems (DMS), and supervisory control and data acquisition (SCADA) systems. For user convenience, the client side of this REST API has been implemented in Python as a scripting tool to perform tasks such as iterative grid architecture optimization.

GridSpice eases adoption into existing work flows by providing an easy-to-use browser-based graphical user interface (GUI) described in Section IV. New users can become familiar with the features of the system through the GUI before using the scripting interface, and advanced users can use the GUI to complement the scripting interface when they wish to perform visual checks on their models. This makes GridSpice ideal for both academic courses and professional use.

The rest of the paper is organized as follows. In the next section, we provide an overview of how we split a simulation into a set of loosely coupled processes running on a cluster with synchronized boundary state. In Section III, we provide some simple pseudocode examples of how to perform a simulation using GridSpice. In Section IV, we describe the software system implementation.

## II. Simulation Clusters

GridSpice simulations run on a dynamically sized cluster consisting of a master node and worker nodes as depicted in

Figure 1. The master node accepts simulation requests from the front end server as described in Section IV, and starts a supervisor process for each new simulation. The supervisor process is responsible for breaking up the simulation into smaller tasks which run on the worker nodes, and keeping the shared state synchronized. The master node load balances jobs across the worker nodes using Oracle Grid Engine [14] as a task queue. Each worker node has a configurable number of slots, each of which can run one job. Figure 1 depicts a cluster in which there are 3 new simulations requiring 18, 22, and 12 worker slots. A supervisor process is running on the master node for each of these simulations sending requests to GridEngine to reserve the required slots. In this figure, each worker node has 8 slots available to run tasks from from a large simulation. Since some tasks may finish sooner than others, there is generally a different number of available slots on each worker node.

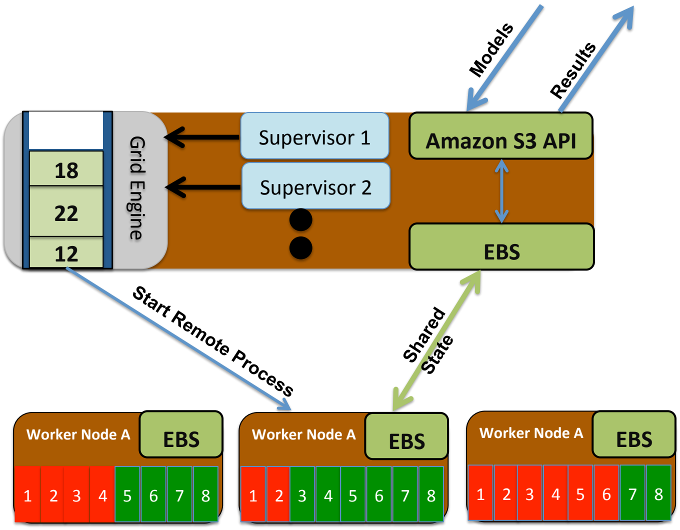


Fig. 1. Simulation cluster.

The supervisor process determines the number of required slots by simply counting the number of networks in the simulation. Each GridSpice simulation consists of exactly one transmission network, zero or more distribution networks, and zero or more generators. However, users can run distribution-only simulations by creating a single-bus transmission network with the distribution network(s) attached to it. Similarly, users can run transmission-only simulations by fixing the loads and generator parameters for each bus instead of attaching a distribution network. Each of the networks in the simulation runs as a worker task.

Each distribution network and generator can run either as a separate agent or read from a static time series data file. For example, a user may choose to simulate a 5000-bus transmission network with 200 distribution networks and 20 generators attached. The load buses on the transmission network that do not have attached distribution simulations may read their $P$ and $Q$ values from a fixed time series file or a defined probability distribution. Similarly, the generation buses that do not have generator simulations attached may read their costs and constraints from a fixed time series file.

Once the worker tasks have started, the supervisor process uses the RePast Simphony [2] to maintain a global simulation clock and synchronize the boundary states between these subsimulations as shown in Figure 2. The supervisor maintains a proxy for each worker task, and keeps track of which worker tasks need update messages when a dependent worker task changes its state. We will now refer to each worker task as an agent in the simulation.

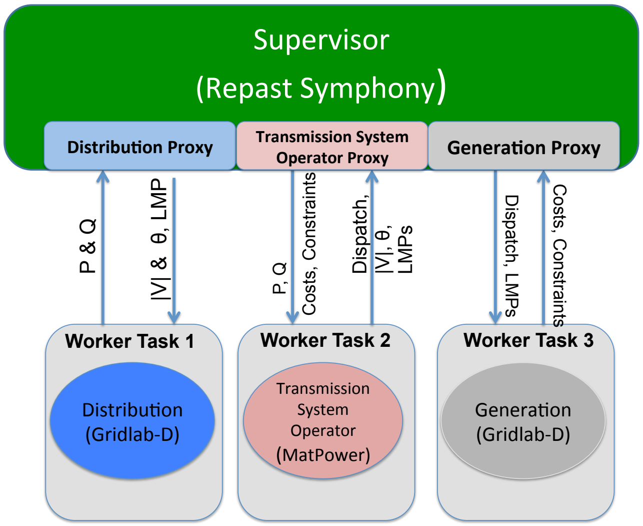


Fig. 2. Three agent simulation consisting of synchronized distribution, transmission, and generation agents running as worker taskes on separate virtual machines.

Each agent runs a simulation tool designed for the network under simulation along with a script to synchronize any shared state with the supervisor process. This script is designed to be extensible so that new simulation tools or can be easily added into the GridSpice framework.

The agents for distribution and generation simulation use Gridlab-D while the agent acting as the system operator runs a lightweight transmission and economic dispatch package based on MATPOWER. In its original form, MATPOWER provides one-shot solutions of the optimal power flow problem. Unlike Gridlab-D, it does not have its own internal simulation clock. Our simple tool re-runs MATPOWER at each time step, resetting the ramping constraints for the next time step based on the operating points from the previous time step. This does not guarantee a globally optimal dispatch schedule over time, however, and we plan to improve this scheme in a future release of GridSpice.

At each time step, the system operator agent publishes the locational marginal price, voltage magnitude, and voltage angle for each load bus on the transmission network as well as the power injection, voltage magnitude, and voltage angle for each generation bus. The system operator agent listens to updates of $P$ and $Q$ from each load bus as well as updates of the costs and constraints from each generator. Analogously, the distribution agents publish their $P$ and $Q$ values at each time step, and listen to updates of their voltage magnitude, voltage angle, and LMP. The generation agents publish their costs and constraints as well as receive notifications about dispatch schedules, voltage magnitude, and voltage angle on the transmission grid.

## III. Example

The following example provides a simple demonstration of the cosimulation capability of GridSpice. It shows how GridSpice may be used to place 200 solar panels over several distribution networks connected to the IEEE 14-bus transmission model [11]. The generators and distribution networks attached to the transmission buses are summarized in Table I.

We use a simple greedy algorithm to select customer locations that may be attached to different transmission buses. Therefore, the locational marginal prices of the transmission network can affect the assignment of the panels. The psuedocode is shown in Algorithm 1.

**Algorithm 1** Greedily place solar panels across multiple distribution networks considering varying locational marginal prices

```
$Model \leftarrow loadModel()$
**for** $k$ = 1 to 200 **do**
    $candidates \leftarrow \{\}$
    **for** $j$ = 1 to 50 **do**
        $model' \leftarrow$Model.copy()
        $networkCount \leftarrow$model'.getDistributions().size();
        $x \leftarrow$Random( 0, networkCount )
        $customerCount \leftarrow$model'.getDistNetwork(x).size();
        $x \leftarrow$Random( 0, customerCount )
        $customer \leftarrow$model'.getNetwork(0).getCustomer(x);
        customer.attach( new SolarPanel() )
        $candidates \leftarrow (model' \vee \{$candidates$\})$
    **end for**
    $Model \leftarrow \underset{C \in candidates}{\operatorname{argmax}} \; Score(parallelSim(C))$
**end for**
```

TABLE I
Transmission network for Example 2.

| Bus | Type | Connections |
|---|---|---|
| 0 | Swing | Generator 1 |
| 1 | PV | Generator 2 |
| 3 | PV | Generator 3 |
| 4 | PQ | Distribution A |
| 5 | PQ | Distribution B , Distribution C |
| 6 | PV | Generator 4 |
| 7 | PQ | Distribution D , Distribution E |
| 8 | PV | Generator 5 |
| 9 | PQ | Distribution F, Distribution G, Distribution H |
| 10 | PQ | Distribution I |
| 11 | PQ | Distribution J |
| 12 | PQ | Distribution K |
| 13 | PQ | Distribution L |
| 14 | PQ | Distribution M |

## IV. System Architecture and Implementation

This section provides an overview of the architecture of the the GridSpice system and implementation details of its user interface, model storage, scripting, and interfacing with external tools and data inputs. Figure 3 gives a top-level view of the entire system. The front-end server coordinates all actions between the user, the data, and the simulators. There are three different mechanisms through which the user can interact with the front-end server—a browser based graphical user interface, a Python library, and any 3rd party library using the REST interface. The front-end server begins a simulation by sending a request to the master node of a simulation cluster, beginning the process described in Section II. We briefly describe each component of this system below.

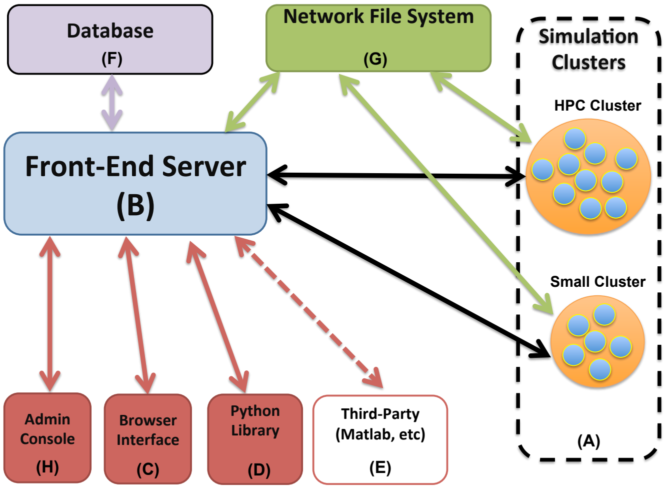


Fig. 3. GridSpice Top-Level System Architecture.

### A. Simulation Clusters

The GridSpice system can have many simulation clusters of varying sizes, and permissions for each cluster can be set on a per-user basis. The operation of the simulation clusters depicted in Figure 3-(A) are explained in detail in Section II

### B. Front End Server

The front-end server coordinates all actions between the user, the data, and the simulators as shown in Figure 3-(B). This components of the front end server are expanded in Figure 4 and discussed below.

The front end server performs the following key functions:

1) *Importing models.* The front-end server can import models from a specified XML format. If the models do not already contain GIS information, we will run an layout algorithm to generate a visualization.
2) *User Authentication.* Each request is authenticated using a cookie or API key.
3) *Access Control Lists.* Access control lists are maintained for each network model in the system specifying which users can access which models.
4) *Simulation Initialization and Monitoring.* The front-end server sets up and monitors the simulations running on the cluster described in Section II

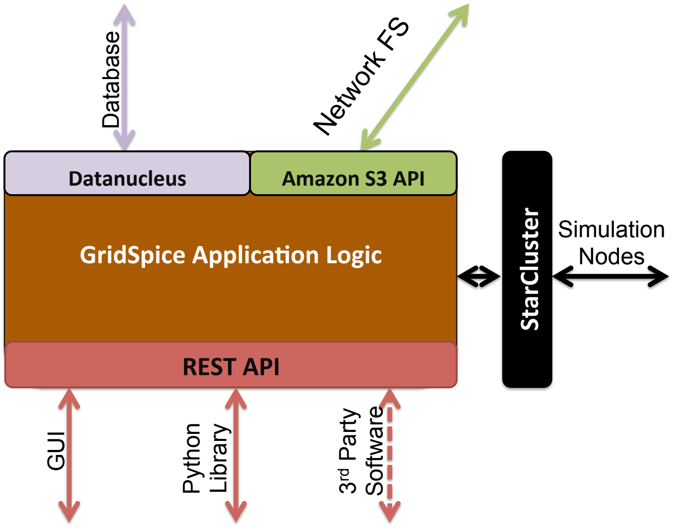


Fig. 4. Front end server.

5) *Cluster Management.* Users with appropriate privileges can start, stop, create, and delete clusters of varying sizes.

### C. Browser Interface

GridSpice provides a graphical user interface (GUI) that allows users to perform the basic tasks of the GridSpice System. The GUI provides a subset of the features available through the REST API and Python library. It is intended to be a tool for beginners to familiarize themselves with the system and for experienced users to sanity check their models. The interface runs in JavaScript in the client's browser and was built using Google Web Toolkit [19] and the EXT-GWT library [20].

The GUI provides several different useful views:

1) *GIS Editor.* If the network model contains GIS information, the GIS Editor allows users to visualize the network on a Google map as shown in Figure 5. If the network model does not contain GIS information, the GIS Editor can visualize the map on a blank background after running the GridSpice layout tool described in Section IV-B.
2) *Explorer Editor.* The graphical user interface offers a separate explorer view that allows the user to list elements in a hierarchical spreadsheet format shown in Figure 6. In this view, the user can apply batch updates to properties of objects that match a given regular expression. The user can apply a number of built-in macros such as adding a roof-top solar to 50% of the buildings in the network.
3) *Object Editor.* The object editor provides a graphical menu to edit the properties of the object. For example, a transformer object has a number of editable properties such as phase, max power, and nominal voltage.
4) *Import Wizard.* The graphical user interface provides an import wizard through which users can upload network models and load data files into the open project.

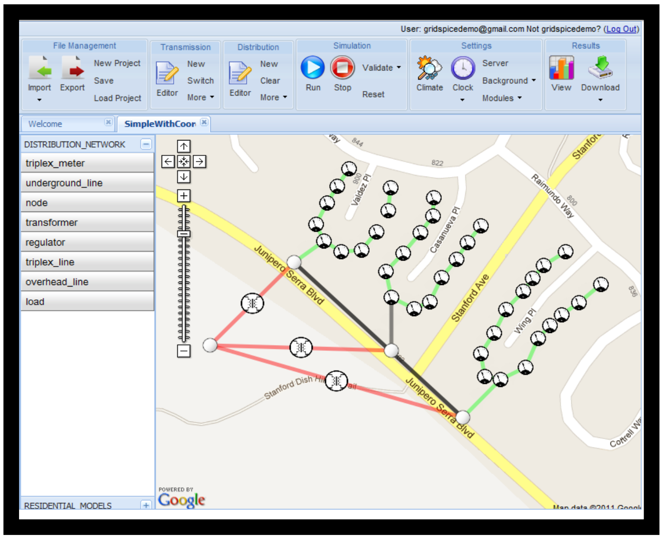
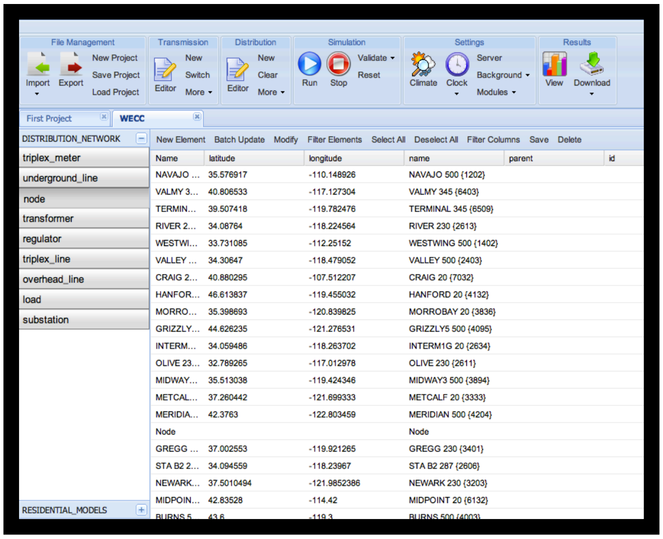


Fig. 5. GIS Editor in GridSpice browser UI.



| Name | latitude | longitude | name | parent | id |
|---|---|---|---|---|---|
| NAVAJO ... | 35.576917 | -110.148926 | NAVAJO 500 {1202} | | |
| VALMY 3... | 40.806533 | -117.127304 | VALMY 345 {6403} | | |
| TERMIN... | 39.507418 | -119.782476 | TERMINAL 345 {6509} | | |
| RIVER 2... | 34.08764 | -118.224564 | RIVER 230 {2613} | | |
| WESTWI... | 33.731085 | -112.25152 | WESTWING 500 {1402} | | |
| VALLEY ... | 34.30647 | -118.479052 | VALLEY 500 {2403} | | |
| CRAIG 2... | 40.880295 | -107.512207 | CRAIG 20 {7032} | | |
| HANFOR... | 46.613837 | -119.455032 | HANFORD 20 {4132} | | |
| MORRO... | 35.398693 | -120.839825 | MORROBAY 20 {3836} | | |
| GRIZZLY... | 44.626235 | -121.276531 | GRIZZLY5 500 {4095} | | |
| INTERM... | 34.059486 | -118.263702 | INTERM1G 20 {2634} | | |
| OLIVE 23... | 32.789265 | -117.012978 | OLIVE 230 {2611} | | |
| MIDWAY... | 35.513038 | -119.424346 | MIDWAY3 500 {3894} | | |
| METCAL... | 37.260442 | -121.699333 | METCALF 20 {3333} | | |
| MERIDIA... | 42.3763 | -122.803459 | MERIDIAN 500 {4204} | | |
| Node | | | Node | | |
| GREGG ... | 37.002553 | -119.921265 | GREGG 230 {3401} | | |
| STA B2 2... | 34.094559 | -118.23967 | STA B2 287 {2606} | | |
| NEWARK... | 37.5010494 | -121.9852386 | NEWARK 230 {3203} | | |
| MIDPOIN... | 42.83528 | -114.42 | MIDPOINT 20 {6132} | | |
| BURNS 5... | 43.6 | -119.3 | BURNS 500 {4003} | | |

Fig. 6. Hierarchical Explorer Editor in GridSpice browser UI .

### D. Python Library

The Python library provides a convenient way for users to create and edit network models, run simulations, and collect results. The library provides a class for each of the available object types with their defined properties and units. This allows the user to easily understand how to change the models.

The Python library allows users to run simulations asynchronously so they can simulate multiple instances of a network in parallel, and hone in on a desired solution. The Python library also provides a module for setting up a new GridSpice system from scratch using one's own Amazon EC2 credentials and our Amazon Machine Image (AMI).

### E. Third-Party Library

The GridSpice front end server implements a REST interface with standard CRUD (create-read-update-delete) interface to each of the main entities in the system. This standardization allows the interface to be implemented in any language or external library. The provided Python library and Browser-based GUI are examples of client-side applications that use this interface, and can referenced as a template for making calls to the GridSpice server from the language or library of choice. Since the REST interface uses the HTTP protocol, built-in Matlab libraries can be used to authenticate with the application server and download the simulation outputs.

### *F. Database*

The front-end server represents metadata associated with all accounts, projects, models, simulations, and data files as plain old java objects (POJOs).

### *G. Network File System*

The GridSpice network file system is implemented using Amazon S3, which provides a redundant data storage infrastructure that allows data to be securely read and written from anywhere on the web.

### *H. Administration Console*

GridSpice provides a web-based administration console hosted on the front-end server. This console allows an authorized administrator to create user accounts, grant users access to projects and models, create new simulation clusters, and start/stop/delete existing simulation clusters.

## V. Conclusion

GridSpice provides a scalable and extensible platform for modeling, design, and planning of the smart grid. It allows utilities, energy services providers, regulators, researchers, educators, and students to solve problems in the smart grid that cannot otherwise be accurately modeled by existing simulators either because of scale or modeling capability. GridSpice's ability to run on public cloud systems with a pay-as-you-go model makes it possible for budget-constrained entities to simulate complex smart grid scenarios. Its generic simulator wrappers also allow users to plugin new tools to seamlessly interoperate with current tools. We have shown that the framework can run aggregated simulations using separate distribution and transmission system simulators while adding minimal overhead. The GridSpice framework has been used in several class projects [6], [7], [8], and [9].

## Acknowledgment

The authors would like to thank Bob Entriken with the Electric Power Research Institute (EPRI) and Professor Ram Rajagopal of Stanford for their insight and guidance. We thank the TomKat Center for Sustainable Energy, Cisco Systems through the Energy and Environment Affiliate Program, Stanford Graduate Fellowship (SGF), and the Stanford Electrical Engineering Research Experience for Undergraduates (REU) Program for providing the financial support to make this project possible. We would also like to thank Pacific Northwest National Labs for their support of the Gridlab-D package through this process.

## References

[1] Chassin, DP; Schneider, K; GridLAB-D: An Open-source Power Systems Modeling and Simulation Environment, Pacific Northwest National Labs

[2] North, M.J., T.R. Howe, N.T. Collier, and J.R. Vos, "A Declarative Model Assembly Infrastructure for Verification and Validation," in S. Takahashi, D.L. Sallach and J. Rouchier, eds., Advancing Social Simulation: The First World Congress, Springer, Heidelberg, FRG (2007).

[3] R. D. Zimmerman, C. E. Murillo-Snchez, and R. J. Thomas, "MATPOWER Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," Power Systems, IEEE Transactions on, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[4] Starcluster. Computer software. Massachusetts Institute of Technology, n.d. Web.

[5] Java Universal Network/Graph Framework. Computer software. JUNG, n.d. Web.

[6] Kallevig-Childers, K. and Thomas, E. and Vogel, L. Comparison of Community-Scale and Distributed Residential- Scale Photovoltaics. CEE272R: Modern Power Systems Engineering. Spring 2012

[7] Nolen, C. and Pearson, A. and Provost, G., Electrical characteristics of distributed Photovoltaics. CEE272R: Modern Power Systems Engineering. Spring 2012

[8] Arnold, E. and Burdick, A. and Mei, S., Photovoltaic integration on distribution networks. CEE272R: Modern Power Systems Engineering. Spring 2012

[9] Anderson, K., Stochastic Control of Electric Vehicle Charging. CS229: Machine Learning. Autumn 2012

[10] Common Information Model. (2012). CIM User Group. Available: http://cimug.ucaiug.org/

[11] Power Systems Test Case Archive. (2012). University of Washington Electrical Engineering. Available: http://www.ee.washington.edu/research/pstca/

[12] Fielding, Roy T.; Taylor, Richard N. (2002-05), "Principled Design of the Modern Web Architecture" (PDF), ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery) 2 (2): 115150, doi:10.1145/514183.514185, ISSN 1533-5399

[13] Amazon Web Services. Computer software. Amazon, n.d. Web.

[14] Oracle Grid Engine. Computer software. Vers. 6.2. Oracle, n.d. Web.

[15] DataNucleus. Computer software. DataNucleus, n.d. Web.

[16] CYMDIST - Distribution System Analysis. Computer software. Cooper Power Systems, n.d. Web.

[17] PSS/E. Computer software. Siemens, n.d. Web.

[18] CIMTool. Computer software. Arnold deVos, n.d. http://wiki.cimtool.org/index.html.

[19] Google Web Toolkit. Computer software. Vers. 2.5. Google, n.d. Web.

[20] EXT-GWT Framework. Computer software. Sencha, n.d. Web.

[21] OpenDSS. Computer software. Electric Power Research Institute, n.d. Web.

# Hybrid Simulator for Cyber-Physical Energy Systems

Ayan Banerjee, Joydeep Banerjee, Georgios Varsamopoulos, Zahra Abbasi, and Sandeep K. S. Gupta
IMPACT Lab, Arizona State University, Tempe, Arizona, USA
Email: {abanerj3,jbanerje,georgios.varsamopoulos,zabbasi1,sandeep.gupta}@asu.edu

***Abstract*—Simulating cyber-physical energy systems such as data centers requires the characterization of the energy interactions between the computing units and the physical environment. Such interactions involve discrete events such as changes in operating modes of cooling units, and also transient processes such as heat flow. An event-based simulator fails to capture continuous transient effects while a time-stepped simulation can ignore events occurring within the decision interval. This paper proposes an error-bound hybrid simulator that integrates discrete event-driven (ED) simulation and finite-horizon time based simulation (FHT) and simulates energy interactions in a cyber-physical energy system. We apply this simulator to a data center case and validate the simulation results by comparing them with simulations performed in previous literature. We also evaluate the accuracy of the simulator by comparing the test case results with realistic data obtained from a real data center deployment. The error bound of the simulator is a user input and influences the time interval of the ED and FHT modules.**

## I. Introduction

Energy-aware and green computing initiatives necessitate management of the equipment according to the energy needs of the computation. Hence, design of computing infrastructure such as data centers should include an integrated analysis of computational processes, energy transfer, and thermal dynamics to reduce cost of operation (e.g., electricity cost), carbon footprint (e.g., by use of alternative energy source), and increase sustainability and lifetime [1], [2]. Such an integrated approach mandates the consideration of data centers as *cyber-physical energy systems* (CPESes), which exhibit both computational behavior (cyber processes) and significant energy exchange (physical processes) within themselves and with the environment [3]. This paper presents the design and deployment of a holistic simulator for CPESes and shows its application to aide the sustainable design of data centers.

Traditionally, simulation of data centers has relied on steady-state assumptions on the cyber performance, e.g., arrival rate, service rate, or throughput [4], [5], and steady-state evaluation of physical dynamics, e.g., thermal interaction [6]–[8]. However, several design parameters of CPESes exhibit a high degree of variability, for example in data centers input workload, thermal exchange between servers and cooling units, and energy supply from renewable sources. Recent research has shown considerable savings in energy and cost in CPESes when such variabilities are factored into the design and management phase [9].

Typically, steady-state assumptions on CPES processes enable event driven simulation (ED). For example, if an average utilization of a data center is assumed over a long period of time, then the server resource allocation following load balancing policy remains steady and does not change. Thus, only events such as infrequent changes in utilization levels have to be simulated. Transient assumptions however, need simulation at a very fine time granularity, i.e. finite horizon time-stepped simulation (FHT), to enable accurate evaluation. In a CPES, both steady-state properties and transient variabilities coexist. Employing event-based simulation on transient properties will lead to non-scalable computation time due to large number of events. On the other hand, FHT simulation may ignore events that occur within a discrete time interval. Thus, a CPES simulator should employ a hybrid simulation paradigm with integration of event-based execution for steady-state processes and time-stepped execution for transient processes into a cohesive simulation engine. Such hybrid operation of CPESes is emphasized by several researchers in different domains [10], [11]. Moreover, both ED and FHT involve discretization of time that will introduce error in evaluation of the continuous dynamics. Such errors can lead to wrong estimation of energy consumption or cyber-performance. Further, the simulator also has to provide a guaranteed error bound on its estimation of the steady-state and transient processes. This paper has the following contributions:

**1.** a formal description of CPES simulation and the hybrid simulation tool that captures the relevant and necessary cyber and physical events and processes,

**2.** evaluation of thermal aware server provisioning algorithms in data centers under transient assumptions using the simulator and validation of results using experiments in BlueCenter[1],

**3.** a theoretical analysis to bound the error of simulations.

## II. Cyber-physical energy system model

The system model of a CPES is shown in Fig. 1, which is detailed for the specific example of data centers in Fig. 2. The CPES has computing units (servers in data centers), which process requests to provide a service, and also has physical behavior such as electricity consumption and thermal interactions with cooling units through air flow. The interactions between the computing units and physical systems in a CPES are limited to energy transfer for operation of the computing units and heat transfer leading to thermal effects. The CPES takes workload as input and provides service with performance guarantees.

In a CPES, there are both discrete events and continuous processes. We consider $\theta$ as the time for which we observe the CPES. Events occur at discrete times $t_j \leqslant \theta$, $j \in J$, where $J$ is the exhaustive set of events occurring during $\theta$, and $t_j$

The work is partly supported by NSF grants CNS #1218505 & #0855277.

[1] BlueCenter is an experimental data center infrastructure developed by the IMPACT Lab (http://impact.asu.edu) at Arizona State University (ASU).

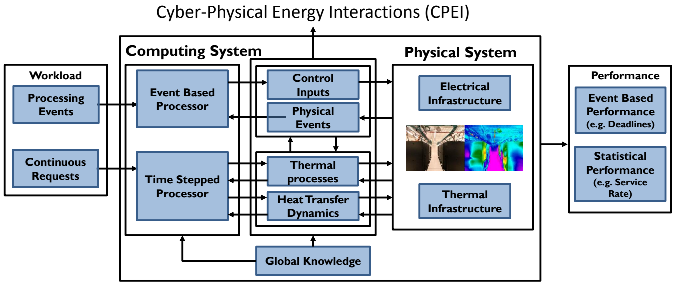


Fig. 1. Cyber-physical energy system model.

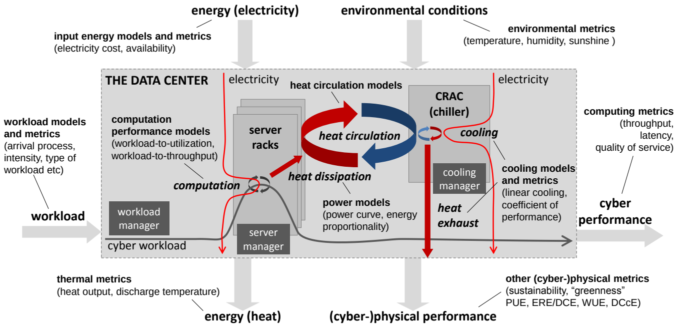


Fig. 2. Data center as a cyber-physical energy system

is the time when the $j^{\text{th}}$ event occurs. In the simulator, we assume discrete functions to determine the effects of events. Continuous time processes and their effects are represented by continuous functions. As a convention, we will denote discrete functions with capital letters and continuous functions with small letters. In a CPES, we consider $n$ computing units. We assume that each computing unit processes some computational workload. Such workload can be of two types: a) event-based ($w_e$), which typically consists of long invariant processing requests that arrive in small numbers at a given time (e.g, high performance computing or HPC jobs), and b) time-stepped ($w_f$), which typically consist of short, highly varying processing requests that arrive in large numbers at a given time (transactional or Internet-like jobs). In a data center CPES, the continuous workload can be represented by an arrival rate, or number of requests per unit time, $w_f = f(t)$ for any $0 \leqslant t \leqslant \theta$, while the event-based workload can be represented by server requests at given times, $w_e = F(t_j)$ for any $j \in J$.

The computing units have two workload managers: an event-based, $M_e$, and a time-stepped, $M_f$. The simulator maintains the current state $CS(t)$, which is a vector consisting of state parameters of the computing units, physical systems and cyber-physical energy interactions. The workload manager takes the workload and the current state as input and outputs the utilization of each computing unit over time. The utilization for event-based workload is a discrete function whereas the utilization for time-stepped workload is a continuous function:

$$\begin{aligned} M_e(w_e(t_j), CS(t_j)) &= U(t_j), \text{ and} \\ M_f(w_f(t), CS(t)) &= u(t), \end{aligned} \tag{1}$$

where $U(t_j)$ and $u(t)$ are $n$-dimensional vectors consisting of the discrete utilization $U_i$ and continuous utilization $u_i$ of a computing unit $i$ at the event time $t_j$ and at the time instant $t$, respectively. The overall utilization of the computing unit is

$$\mu(t) = u(t) + U(t_j), \text{ for } t_j < t \leqslant t_{j+1}. \tag{2}$$

The utilization of the computing units controls the power consumption of the servers:

$$p(t) = f_p(\mu(t)), \tag{3}$$

where $f_p$ is a transformation function obtained by profiling the computing units. In data centers this power consumption is the key link that gives rise to energy interactions.

The heat interactions within a CPES is fueled by the power consumption of the computing units. For example, in a data center, the air flow causes heat from one server to circulate within the room and affects the temperature rise at other servers. The heat map of a data center follows a set of linear differential equation (discussed in Section III-A2). The temperature rise map in a data center also is linearly dependent on the amount of heat flowing into a particular region. Based on the temperature rise, the cooling unit in a data center reacts by serving cold air into the data center. The rate of change of cold air temperature is also linearly related to the inlet temperature of the data center. Such transient processes thus affect the different physical parameters of the CPES. These physical parameters as well as the computational parameters form a CPES **state**. In case of data center CPES, the state consists of: a) server utilization, b) server power consumption, c) re-circulated heat matrix, where each entry is the fraction of heat from a computing unit to some other, d) inlet temperatures of the server, and e) outlet temperature of computer room air conditioning units (CRACs). A sample of this state at any time $t$ is $CS(t)$, or the current state of the data center CPES.

Moreover, although the interaction are governed by linear continuous dynamics, the evolution of the physical parameters of the CPES can cause discrete events. For data centers, if the server inlet temperature rises above a pre-specified redline, then the processing speed of the server reduces i.e. it throttles. If the inlet temperature of the CRAC increases beyond a certain point, then it starts extracting more heat. These events can have effects on both the management algorithms and the physical dynamics. For example, a server throttling will change the job execution times, which will result in a change in the utilization vector $\mu(t)$. Again a change in CRAC power extraction will change the heat flow dynamics in the data center. Note that there exist a close coupling between decisions of the management algorithms and the effects in the physical domain through the current CPES state variables. Further, we notice that there are both discrete domain and continuous domain variables in a CPES. Thus, a CPES simulator should allow hybrid simulations in two major senses: a) it should allow simulations of management algorithms that are aware of physical dynamics in the data center, and b) it should allow both event-based and time-stepped simulations.

## III. Overview of proposed simulator (SIM)

In an event-based simulation, the time is discretized into unequal intervals, where each interval starts at the start of an event. While in a time-stepped simulation the time is discretized into small equal intervals. The event-based repre-

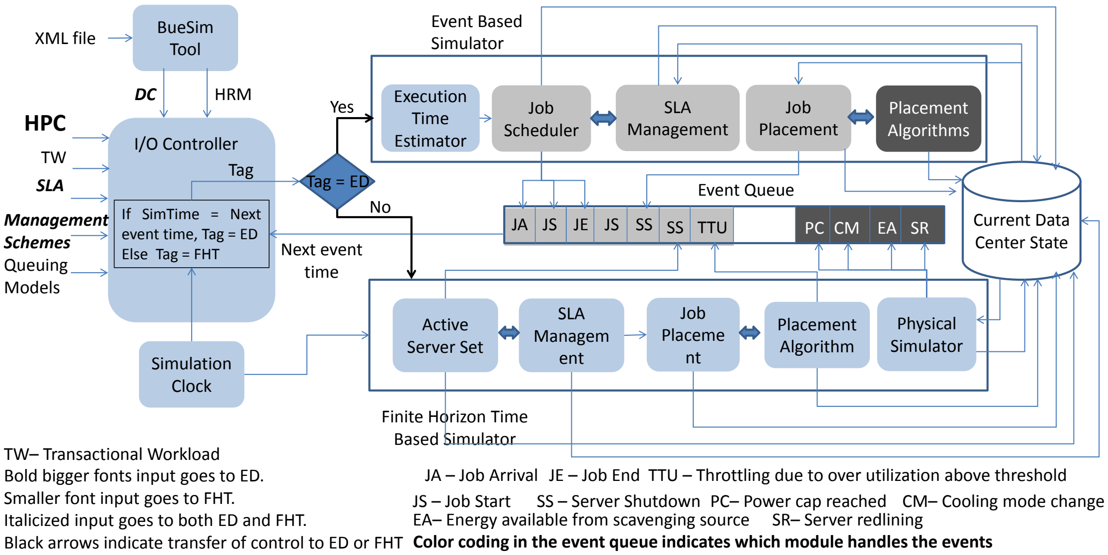

XML file
BueSim Tool
DC
HRM
HPC
TW
SLA
Management Schemes
Queuing Models
I/O Controller
Tag
If SimTime = Next event time, Tag = ED Else Tag = FHT
Simulation Clock
Yes
Tag = ED
No
Next event time
Event Based Simulator
Execution Time Estimator
Job Scheduler
SLA Management
Job Placement
Placement Algorithms
Event Queue
JA JS JE JS SS SS TTU
PC CM EA SR
Current Data Center State
Active Server Set
SLA Managem ent
Job Placeme nt
Placement Algorithm
Physical Simulator
Finite Horizon Time Based Simulator
TW– Transactional Workload
Bold bigger fonts input goes to ED.
Smaller font input goes to FHT.
Italicized input goes to both ED and FHT.
Black arrows indicate transfer of control to ED or FHT
JA – Job Arrival JE – Job End TTU – Throttling due to over utilization above threshold
JS – Job Start SS – Server Shutdown PC– Power cap reached CM– Cooling mode change
EA– Energy available from scavenging source SR– Server redlining
Color coding in the event queue indicates which module handles the events

Fig. 3. The hybrid simulation engine for data centers.

sentation of time expedites the simulation in case the events occur sparsely. The time-stepped representation of time is important for continuous evaluation of physical dynamics as well as highly varying computational workloads.

If a time-stepped representation is applied to both ED and FHT simulation then events that occur in between two time intervals may be ignored. However, if we apply event-based simulation to both ED and FHT simulation then we will have a large number of events to process at a given time which will slow down the simulation. Further, since we are discretizing the time into intervals the evaluation of the physical dynamics incurs errors. In the following sections we discuss our hybrid simulator and also show how we can bound the error in estimation of the physical dynamics.

We propose a two-tier hybrid simulator which has both event driven (ED) and finite horizon time-based (FHT) simulation. The FHT simulator uses short time cycles (slots) to manage: (i) transient behavior, (ii) interaction between the event-driven simulation and time-based simulation of cyber events, and (iii) events associated with the transactional workload. The ED simulator simulates: a) resource management modules that usually has long decision time interval such as server on/off management, b) low variability workloads such as HPC workloads, and c) events generated by the FHT.

Fig. 3 shows a simple example of how our simulator works for data centers. The example assumes that there is one-tier FHT and ED simulator engine, and that the simulator simulates job scheduling, server and cooling management. The input to the simulator is a data center configuration specified in XML. The specification is processed by BlueSim[2] to obtain transient heat models of data centers (discussed in Section III-A2). The XML specification and the transient heat model parameters along with the HPC or transactional workload, management schemes, and service level agreements are provided as input to the controller. As shown in Fig. 3, the event driven simulator acts as a controller to the time based simulator. Jobs are first fed to the job dispatcher in the ED. According to workload tags in the jobs similar to the one suggested in SIM [4] the transactional jobs are routed to the FHT, while the HPC jobs are routed to the ED. ED simulator processes events such as job arrival, job start and job end, and communicates with the workload manager, and placement modules to assign jobs to servers. FHT simulator processes the statistical information of the workload instead of every individual job. Therefore, the data center processes of these workload is simulated per-slot instead of per-request. Both ED and FHT can generate events and update $CS(t)$ of the data center.

Initially, the control is with the ED. The ED processes an event and gives the control back to the FHT. The FHT runs a fine grain time simulation until the simulation clock suggests that it is time for processing the next event in the event queue. The control is then transferred to the ED. In the following sections we discuss the models of computation and physical processes used by the simulator.

### A. *Transient models used by the simulator*

The proposed simulator needs to consider both cyber and physical transients. Cyber transients include dynamic queuing models with mathematical expressions for utilization, service rate, and latency, workload generation models, and server throttling models. The physical transients include cooling models, heat recirculation models, and server power models.

*1) Cyber aspects:* An efficient simulator should rely on fast quantitative models to predict the transient and steady-state metrics of the computation's performance. Models of transient variation of performance is determined from experiments conducted in the BlueCenter. The second graph in Fig. 4 shows the variation of CPU utilization of IBM X Series 336 servers with different arrival rates which clearly shows a threshold utilization (150 req/s) after which the utilization becomes constant. This effect is due to rejection of jobs by the

[2] BlueSim is a CFD simulator developed by IMPACT Lab [4]. It carries out a set of CFD simulations to obtain heat recirculation matrix that can be used to do fast simulations of transients.

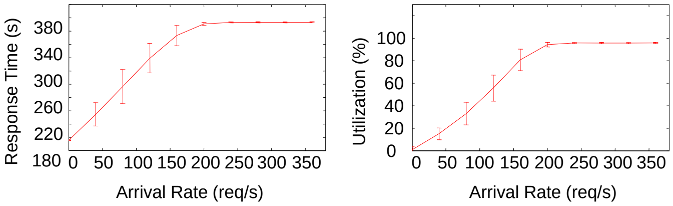


Fig. 4. Round trip time and utilization v.s. arrival rate, respectively.

servers since the arrival rate is greater than the service rate. At this stage the servers are utilized at a constant rate and the round trip time also levels off. The saturation of utilization shown by the first graph in Fig. 4 is modeled by fitting a curve to this graph using the following function:

$$\text{performance} = f(\lambda, w, s, T) = \begin{cases} f_1(\lambda, w, s) \text{ if } T < T_{\text{threshold}}, \\ f_2(\lambda, w, s) \text{ if } T \geqslant T_{\text{threshold}}, \end{cases}$$

where $\lambda$ denotes the workload intensity, $w$ denotes the workload type , and $s$ denotes the server type.

Particularly, the simulator should provide $f_1$ and $f_2$, to map the input workload of a server to its utilization level $u$ and the delay $d$ of requests, e.g., $(u, d) = f_1(\lambda, w, s)$. This mapping is obtained by assuming a dynamic queuing model similar to the one used in large scale data networks [12].

Based on the decision of data center management, it manages the workloads by assigning it to a server queue. It simulates the transient workload processing of the server by considering a G/G/c queue dynamics where rate of change of queue is denoted by $\dot{Q}_t$. The service rate for the server is currently considered uniform, however it varies based upon different states of the servers, such as throttling. Managing the arrival rate $a_t$, departure and service rate $s_t$ into queues it computes the server utilization $u_t$. If there is excess workload than the queue capacity ($q_{max}$) of the active servers, it is rejected. We use the queue model from [13]:

$$\begin{array}{ll} \dot{Q}_t = \frac{(a_t - s_t) - (a_{t-1} - s_{t-1})}{t-(t-1)}, & \text{if } a_t - s_t < q_{max}, \\ \dot{Q}_t = 0, & \text{if } a_t - s_t > q_{max} \text{ or } a_{t-1} - s_{t-1} > q_{max}, \\ a_t - s_t = q_{max}, & \text{if } a_t - s_t > q_{max}, \\ a_t - s_t = 0, & \text{if } a_t - s_t < 0. \end{array} \qquad (4)$$

**Throttling Model:** Throttling of servers occur when the temperature rises above a redline value. We define a new parameter Red Line Temperature (RLT) which is the maximum outlet temperature for any of the server after which it throttles. The estimation of the redline temperature is modeled using physical models described in Section III-A2.

*a) Workload (generation) models.:* The simulator uses traces and synthetic workload models particularly to model Internet-type workload. The synthetic models should exhibit long-term and short term variations of the workload. The long term variations can be modeled using seasonal models such time series techniques e.g., SARIMA [14] while the short term variations can be done using previously proposed stochastic based models for different workload types, such as transactional web workload [15], and Web 2.0 workload [16].

We generate the scalable workload using Surge [17]. As workloads get more diverse, many researchers focus on considering transient workload and its effect of data center dynamics [18]. The generated workload in our study has a time-slot granularity of 10 ms.

*2) Physical aspects:* The physical aspects include:

*a) Heat circulation model:* The simulator uses the heat recirculation model proposed in [9], [19] which predicts three aspects of thermal behavior: *division (spatial distribution)*: how the heat produced by each computing server is split and circulated into each other server or to each chiller; *temporal distribution*: how the heat portion of one server to another is distributed over time, which also captures the *hysteresis*: how long the heat takes to travel from one unit to another. A key assumption of this model is that the airflows in the data center are controlled and dominated by the fans such that the portion of air arriving everywhere over time is invariant to temperature. This ignores changes in the airflow due to pressure changes caused by the temperature such as hotter air rising faster (a likely cause for slight inaccuracy in our model). The heart of the model is a collection of $n \times n$ ***temporal contribution curves***, $c_{ij}(t)$, each one denoting how heat arrives to the air inlet of a *sink* (*receiving*) server $j$ from the air outlet of a *source* server $i$. (For notational convenience, the chiller also is included in the $ij$ enumeration). A $c_{ij}$ curve conforms to Eq. 5.

$$c_{ij}(t) : (-\infty, 0] \to [0, 1], \text{ with } \int_{-\infty}^{0} c_{ij}(t)dt = 1. \qquad (5)$$

$$\bar{T}_{ij}(t) = \int_{-\infty}^{0} c_{ij}(\tau) T_{i-}(t+\tau) d\tau. \qquad (6)$$

$$T_{j+}(t) = \sum_{i=1}^{n} w_{ij} \bar{T}_{ij}(t) = \sum_{i=1}^{n} w_{ij} \int_{-\infty}^{0} c_{ij}(\tau) T_{i-}(t+\tau) d\tau. \qquad (7)$$

A key concept of the model is the ***contributing temperature*** $\bar{T}_{ij}(t)$ of a source server $i$ to a sink server $j$, which is the temperature rise as affected by the history of the source server $i$'s outlet temperature $T_{i-}(t)$ through the spreading of $c_{ij}$ (see Eq. 6). In addition to the $c_{ij}$ curves, the model assumes $n \times n$ ***weighting factors*** $w_{ij}$, which define how each contributing temperature $\bar{T}_{ij}(t)$ factors into the actual temperature $T_{j+}(t)$ at the air inlet of server $j$. The weighting factors $w_{ij}$ can be organized into a matrix $W$. The core hypothesis of the model is that the temperature at the air inlet of a server $j$ is the convex weighted sum of the temperature contributions of all servers (and of the chiller) as they accumulate over time from the past (Eq. 7), where $\sum_{i=1}^{n} w_{ij} = 1, \forall j$, because of the thermal isolation assumption.

*b) Cooling models:* The cooling process has two aspects. One aspect is to describe the "supply" temperature of a chiller given the input temperature to it. Transient cooling models have been developed in our prior work [9], [20], [21]. These transient cooling models are used in the simulator. The RLT of the servers can be estimated using the heat flow dynamics within the data center. The heat flow in data center is represented using the models described in our prior work [9].

### *B. Error analysis of the simulation*

Transient models are continuous time models. In order to simulate in a computer, the time has to be discretized. Such discretization can lead to errors in estimating the values of the variables in the physical system models. In this section we determine a temporal sampling such that the error of simulation does not exceed a pre-specified bound $\epsilon$. To this effect, we observe that most of the models considered in this work are linear time invariant transients. The only non-linearities existing in the data center transient is the conversion of arrival rate to utilization and then from utilization to power which is ignored in this error analysis. For example, if we consider the transient heating model, the error analysis can be done by considering the linear time invariant system,

$$\dot{x} = Ax + B, \tag{8}$$

where $x$ is the $n+2n^2$ dimensional vector obtained from the transient cooling model. The vector consists of outlet temperatures of all $n$ chassis, the temperature contributions of a chassis $j$ to the inlet of a chassis $i$, which is $n^2$ in number, and $n^2$ other intermediate variables used to express the transient heating effects as an LTI system. We want to ensure that the error due to temporal discretization is less than $\epsilon$, i.e.,

$$\|x(t+\tau) - x(t)\|_\infty < \epsilon, \forall \tau \in [0, h_t], \tag{9}$$

where $\|x(t+\tau) - x(t)\|_\infty$ denotes the infinity norm which means the maximum of the elements in the vector $x(t+\tau) - x(t)$. Here $h_t$ is the time interval of the FHT.

If we consider the maximum time variance of the vector $x(t)$ and express it in terms of $h_t$, then we can get an expression for $h_t$ as shown below in terms of $\epsilon$.

$$\begin{aligned} \|\dot{x}(t)\|_\infty &= \|Ax(t) + B\|_\infty, \qquad (10)\\ \frac{\|x(t+h_t) - x(t)\|_\infty}{h_t} &\leqslant \|A\|_\infty\|x(t)\|_\infty + \|B\|_\infty,\\ \|x(t+h_t) - x(t)\|_\infty &\leqslant h_t(\|A\|_\infty\|x(t)\|_\infty + \|B\|_\infty). \end{aligned}$$

Thus, to satisfy Equation 9, the right hand side of the Equation 10 can be less than $\epsilon$. Thus, we have

$$h_t(\|A\|_\infty\|x(t)\|_\infty + \|B\|_\infty) \leqslant \epsilon \Rightarrow h_t \leqslant \frac{\epsilon}{\|A\|_\infty\|x(t)\|_\infty + \|B\|_\infty}. \tag{11}$$

The value of $h_t$ depends on the maximum value of the elements in $x(t)$. This maximum value can be arbitrary unless we define the type of errors we cannot tolerate. Equation 11 can be used in two ways: a) given an error of $\epsilon$ °C on the outlet temperature of the servers an appropriate $h_t$ can be determined such that the simulation error does not go beyond $\epsilon$, b) given a $h_t$, FHT time slot, the error in simulation can be determined.

## IV. Validation and Evaluation

The hybrid simulator takes a mixture of SURGE workload trace for a total of 30 percent of total data center utilized and HPC workload trace from ASU HPC data center. The simulation was run for a 24 chassis data center with 5 servers in each chassis, service rate of 200 request per second, queue capacity of 30 requests and RLT of 35 °C. The simulation was run for 1 hour with each iteration having time intervals of 100 ms. Two epochs of 30 minutes each were considered for workload management with chassis 14 through 19, 22 and 24 on in the first epoch and 16 and 17 in the second epoch. The CRAC switches cooling mode when the inlet temperature exceeds 28 °C. The hybrid simulator is compared with GDCSim developed by the authors, was run for same input data. The two simulations were compared with respect to the total power consumed by the data center (Figs. 5 and 6),

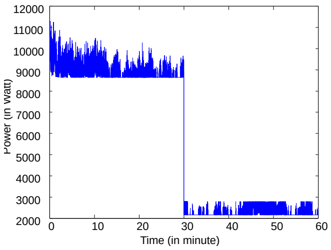


Fig. 5. Total power consumed in the data center using transient simulator for 1 hour period.

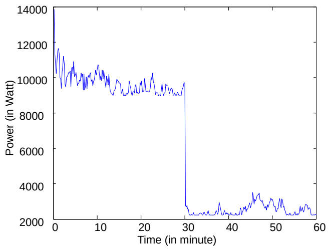


Fig. 6. Total power consumed in the data center using GDCSim simulator for 1 hour period.

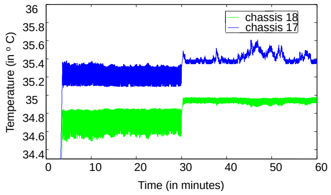


Fig. 7. Inlet temperature variation using transient simulator for chassis 17 and 18 for 1 hour period.

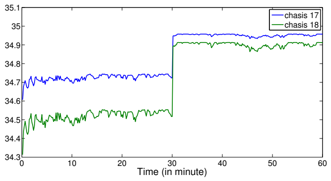


Fig. 8. Inlet temperature variation using GDCSim simulator for chassis 17 and 18 for 1 hour period.

and inlet temperatures (Figs. 7 and 8) for two chassis (chassis 17 and 18). We see that although the average trend is same in both power and temperature key events are missed in GDCSim simulator. Several short CRAC switching events are missed while computing total power hence GDCSim underestimates the power by 7.3%. From the temperature plots we see that due to high variability of temperature there are several places where chassis 17 crosses RLT and is not detected by GDCSim.

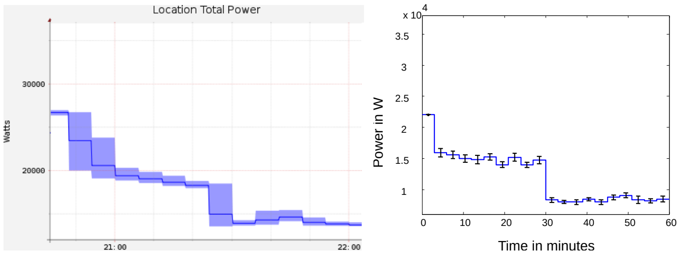
The same utilization schedule was run in BlueCenter and we recorded the power consumption. Figs. 9(a) and (b) compare the power consumption from BlueCenter and the simulator averaged over 3 minutes. There is a near match between the two curves from experiment and simulation plots.

### A. Error evaluation

Physical events such as server redlining, power consumption reaching a cap, or CRAC mode change are based on thresholds on $x(t)$. For example, if we want to capture all server redlining events then we should have $\|x(t)\|_\infty = RLT = 35$ °C, $\|A\|_\infty = 1$ or maximum value of heat recirculation, and $\|B\|_\infty = \frac{p_{max}}{C_p}$ where $p_{max}$ is the maximum power of any server and $C_p$ is the specific heat, then for our chosen simulation time granularity $h_t = 100$ ms we get an $\epsilon$ of 3.7 °C. This means that the outlet temperature predictions have an error of $3.7/(T_{max} - T_{min}) = 9.25\%$, where $T_{max} = 45$ °C the maximum temperature attained by any server and $T_{min} = 5$ °C the minimum temperature. The error can be further reduced by reducing $h_t$, but consequently increasing the simulation time.

## V. Related Work

Authors in [22] develop a simulation tool to evaluate the data center power efficiency through simulating the cyber aspects of data centers. The tool uses steady-state queuing models to simulate a data center servicing multi-tier web applications. BigHouse further improves the MDCSim models



(a) (b)

Fig. 9. a) Power consumption of servers in BlueCenter (image captured from BlueCenter's Sentry Power Manager[3] monitoring software), and b) Power consumption from simulator with same utilization.

though introducing stochastic queuing system which provides a stochastic discrete time simulation of generalized queuing models to simulate the performance of the data center applications in more detail than that of MDCSim [15]. CloudSim, is capable of simulating the per-server power and performance metrics, where the main abstraction unit is a VM [23]. The above tools ignore simulating the physical aspect of data centers and the potential interaction between cyber and physical aspects. They rely on the steady-state models to estimate the power consumption and performance of servers, lacking simulating the transient processes that can have long-time effects on data center (e.g., redlining) and an error analysis to specify the confidence intervals of the results. GDCSim, used to compare with the proposed simulator, is a cyber physical simulator, which accounts for both batch and transactional jobs, however it relies on steady-state models to estimate the metrics related to cyber and physical aspect of data centers. There are also some simulation tools that provide estimation of data center physical aspects such as Mercury [6], but ignore interaction between physical and cyber aspects.

## VI. Conclusions

This paper proposes an error bounded hybrid simulator that can process both transient and steady-state behavior in a cyber-physical energy system. The simulator has an event driven module which considers computational and physical events and a finite horizon time driven module which considers transient computation and physical processes. We have used it for data center systems and validated its operation through experiments in a real deployments, BlueCenter. We also propose an analysis on the error bounds of the simulator. The error bound can be varied by adjusting the step size of the FHT. With a 100 ms time step we obtained an upper error bound of around 9% of the average temperature rise.

## References

[1] S. K. S. Gupta, T. Mukherjee, G. Varsamopoulos, and A. Banerjee, "Research directions in energy-sustainable cyber-physical systems," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 57 – 74, 2011.

[2] S. K. Gupta and G. Varsamopoulos, "Towards formal framework for designing energy-cps," *National Workshop on Research Directions for Future Cyber-Physical Energy Systems*, 2009.

[3] A. Banerjee, K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security and sustainability of mission-critical cyber physical systems," *IEEE Proceeding on cyber-physical systems*, 2011.

[4] S. K. S. Gupta, R. R. Gilbert, A. Banerjee, Z. Abbasi, T. Mukherjee, and G. Varsamopoulos, "GDCSim - an integrated tool chain for analyzing green data center physical design and resource management techniques," in *International Green Computing Conference (IGCC)*, 2011.

[5] J. S. Ahn and P. B. Danzig, "Packet network simulation: speedup and accuracy versus timing granularity," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 5, pp. 743–757, 1996.

[6] T. Heath, A. P. Centeno, P. George, L. Ramos, and Y. Jaluria, "Mercury and Freon: temperature emulation and management for server systems," in *ASPLOS-XII*. New York, NY, USA: ACM Press, 2006, pp. 106–116.

[7] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "Tacoma: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 2, pp. 11:1–11:37, jun 2012.

[8] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888 – 2904, dec 2009.

[9] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, and S. K. S. Gupta, "A transient model for data center thermal prediction," in *International Green Computing Conference (IGCC)*, jun 2012.

[10] A. Banerjee and S. K. S. Gupta, "Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study," *Intl' Conf' on Cyber-Physical Systems*, April 2013.

[11] A. Banerjee and S. K. Gupta, "Tackling new frontiers in modeling and analysis of cyber-physical systems," *1 st workshop on Cyber-Physical Systems Education*, 2013.

[12] Y. Guo, W. Gong, and D. Towsley, "Time-stepped hybrid simulation (tshs) for large scale networks," in *INFOCOM 2000.*, vol. 2. IEEE, 2000, pp. 441–450.

[13] S. Bohacek, J. P. Hespanha, J. Lee, and K. Obraczka, "A hybrid systems modeling framework for fast and accurate simulation of data communication networks," in *ACM SIGMETRICS*, New York, NY, USA, 2003, pp. 58–69.

[14] V. D. Van Giang Tran and S. Bacha, "Hourly server workload forecasting up to 168 hours ahead using seasonal arima model," in *Industrial Technology (ICIT), 2012 IEEE International Conference on*, march 2012, pp. 1127 –1131.

[15] D. Meisner, J. Wu, and T. F. Wenisch, "Bighouse: A simulation infrastructure for data center systems," in *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on*, apr 2012, pp. 35 –45.

[16] A. Abhari and M. Soraya, "Workload generation for youtube," *Multimedia Tools and Applications*, vol. 46, pp. 91–118, 2010.

[17] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 1, pp. 151–160, 1998.

[18] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. of the 3nd ACM Symposium on Cloud Computing, SOCC*, vol. 12, 2012.

[19] G. Varsamopoulos, M. Jonas, J. Ferguson, J. Banerjee, and S. K. S. Gupta, "Using transient thermal models to predict cyberphysical phenomena in data centers," *Sustainable Computing: Informatics and Systems*, 2013, in print.

[20] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta, "Integrating cooling awareness with thermal aware workload placement for hpc data centers," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 2, pp. 134 – 150, 2011.

[21] G. Varsamopoulos, A. Banerjee, and S. K. S. Gupta, "Energy efficiency of thermal-aware job scheduling algorithms under various cooling models," in *International Conference on Contemporary Computing IC$^3$*, Noida, India, aug 2009, pp. 568–580.

[22] S.-H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "Mdcsim: A multi-tier data center simulation platform," in *CLUSTER.*, 31 2009-sept. 4 2009, pp. 1 –9.

[23] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *CoRR*, vol. abs/0903.2525, 2009.

[3] http://www.servertech.com/products/sentry-power-manager/

# Co-Simulating Power Systems and Communication Network for Accurate Modeling and Simulation of PMU based Wide Area Measurement Systems using a Global Event Scheduling Technique

Yi Deng, Hua Lin, Sandeep Shukla, James Thorp, Lamine Mili
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061 USA
{yideng56, birchlin, shukla, jsthorp, lmili}@vt.edu

*Abstract*-- **In this paper, we describe our global event-driven co-simulation framework GECO that we developed to co-simulate power systems dynamics with data network activities. The GECO framework utilizes global event scheduling across two different simulators with distinct simulation disciplines -- to eliminate common synchronization errors often found in federated simulation platforms. We also illustrate the use of the GECO framework on two PMU-based WAMS applications: cyber-attacks impacts analysis on all-PMU state estimation, and PMU-based out-of-step protection. The experimental results described in this paper not only show the efficacy of the GECO framework but also illustrate the utility of GECO in WAMS modeling and simulation.**

***Keywords—Phasor Measurement Units (PMUs), Wide Area Measurement System (WAMS), Co-Simulation, Cyber-Attack, Out-of-Step***

## I. Introduction

The modern power systems are becoming more complex and hard to predict with the deployment of advanced information technologies and communication technologies. As a result, system engineers must gradually pay more attention to the impacts of underlying communication networks [1]. They must pay close attention to both the power system itself and the characteristics of the communication infrastructure and their influence on relevant applications running on it. The existing cyber-attacks such as the Stuxnet worm have indicated that any malicious cyber-attack will not just cause harm to the cyber system, but will also severely affect the normal operation of physical systems under the control of the cyber system being attacked. Field-testing could be too late to observe any major defect in the system and retrofit modifications. Therefore, research efforts related to constructing a co-simulation platform for power systems and the corresponding communication systems have become a new research direction.

To accurately model and simulate the interactive behavior between a power system and a communication network, we have to construct an appropriate co-simulation platform. Owing to the uniqueness and complexity of each individual system, there are only a few co-simulators developed specifically for power system applications [2]. EPOCHS [3] pioneers the efforts to integrate a power system modeling tool with underlying communication network simulator as an integral platform. A similar work in [4] improves the synchronization mechanism based on DEVS formalism and integrated with NS2. Authors in [5] make an effort by integrating MATLAB Simulink and OPNET to study the reliability of wide area measurement system (WAMS) under information and communication architecture. An integration of virtual test bed software and OPNET in [6] is proposed to simulate distributed power electronic devices in a small-scale application. Co-simulation platform is also used as a test bed for assessing the vulnerabilities and cyber security issues in SCADA systems. Authors in [7] integrate PowerWorld and RINSE to construct a SCADA security test bed. All the proposed co-simulation platforms are designed for the traditional power system applications so that it is reasonable to ignore the possible synchronization errors. However, the phasor measurement unit (PMU) based WAMS applications are more sensitive to the co-simulation errors that may be introduced by the two-simulator interaction. This is because of the time resolution in WAMS data collection, and state estimation are in the order of 30 ms. Synchronizing two simulators at that granularity would solve the synchronization error problems but would slow down the co-simulation too much.

In this paper, we describe a Global Event-driven CO-simulation (GECO) framework [2] and analyze how to eliminate the synchronization errors using a global event scheduling technique. Then we study two WAMS applications based on this framework. All PMU state estimation and PMU-based out-of-step (OOS) protection are selected to demonstrate our methodology. Since the communication requirements of these two WAMS applications are different, we apply different co-simulation scenarios and study the cyber impacts on these two power systems. More power system applications and experiments using GECO have been described in our previous work [8][9][10].

The rest of the paper is organized as follows: in Section II, the principle of dynamic power system simulation, event-driven network simulation, and global event-driven co-simulation is described. The architecture of GECO framework and its implementation are reviewed in Section III. In the following two sections, we study the power system contingency of the two WAMS applications respectively. The paper is concluded in section VI.

This work was funded by NSF project EFRI 0835879.

## II. Global Event-Driven Co-Simulation Framework

In order to combine a power system simulator and a network simulator into an integral co-simulation framework, the co-simulation platform has to synchronize the simulation time from two distinct simulation models. This section describes the simulation mechanisms for both the power system and the communication network respectively and reviews our optimized global event scheduling technique.

### *A. Dynamic Simulation for Power System*

According to the power flow characteristics, a power system dynamic simulation that exhibits the dynamic behaviors of power systems can be modeled as a continuous time system simulation. System state variables such as voltages, currents, and phasors change in a continuous manner. Typically, the system dynamic variables can be expressed by a set of differential equations. For numerical analysis and simulation, the differential equations are discretized and the time base is evenly divided into small steps. When simulating, the next system state will be derived from current system state. In addition, small variations of the state variables are integrated to approximate the system trajectory. The discretized time step is around 5ms, which is often very small, so that system variables do not have an abrupt transition within the time step.

In implementation, the dynamic simulation procedure of the power system is numerically discretized in a sequence of discrete iteration rounds as shown in Fig. 1. First, the simulation calculates the power network flows to initialize system states at time $t_0$. Then the simulation accesses an iteration loop, which is advanced by a fixed time step $\Delta t$. Within the iteration, the simulation has to firstly calculate the power network boundary variables and then calculate the next variables of the dynamic models. Next, the system state variable derivatives are calculated. The last stage of the iteration is to calculate the integrations for the next iteration [2]. The simulation will continue to execute until it reaches a predefined stop time.

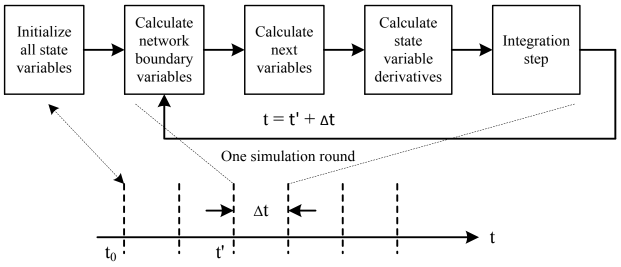

Initialize all state variables
Calculate network boundary variables
Calculate next variables
Calculate state variable derivatives
Integration step
t = t' + Δt
One simulation round
Δt
t0
t'
t

Fig. 1. The dynamic simulation procedure of a power system - from [2]

### *B. Event-Driven Simulation for Communication Network*

The most common method for network simulation is the discrete event-driven simulation. Discrete event-driven simulation is used for systems whose operating state is only subject to changes due to discrete events. In reality, the discrete events are usually unevenly distributed with respect to time. The method of time discretization into small time intervals cannot be applied to discrete event systems since it is difficult to choose the time step. Selecting either too small or too long of a time step will decrease the simulation efficiency. Therefore, the discrete event-driven simulation method with a dedicated event scheduler and an event list queue is applied. An event list is a queue that stores system events with timestamps in a chronological order. Fig. 2 shows a communication network simulation that runs in an event-driven manner. The objective of the communication is to send a packet from node 1 to node 4. The top event in the queue is “node 1 sends packets to node 2” with timestamp. Then the lower event “node 2 receives packets from node 1” will be generated and stored in the event queue. This simulation will continue in this way until the ending identifier is met.

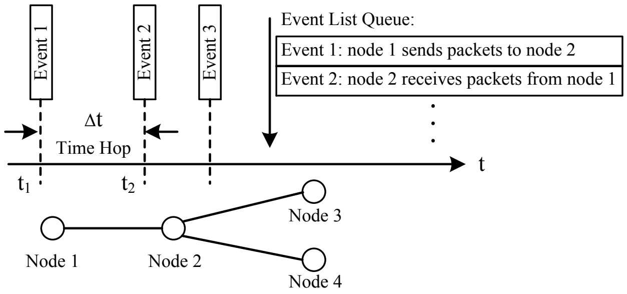

Event 1
Event 2
Event 3
Event List Queue:
Event 1: node 1 sends packets to node 2
Event 2: node 2 receives packets from node 1
Δt
Time Hop
t1
t2
t
Node 1
Node 2
Node 3
Node 4

Fig. 2. The communication network simulation procedure - from [2]

### *C. Optimized Co-simulation Framework*

The most challenging hurdle for designing a co-simulation platform is to schedule a synchronization mechanism for two distinct simulators. Authors in [3] provide an intuitive way that predefines explicit time-stepped for hybrid simulator synchronization. In this method, several synchronization points have to be predefined. When the co-simulator starts to execute, the individual simulators run independently until both of them reach the predefined synchronization point. At this point, the two simulators pause and exchange information reciprocally. After that, two simulators restart and repeat the synchronization procedure. From Fig. 3 we can find that this synchronization method will bring simulation errors shown as “Error 1” and “Error 2”. If an exchange request happens within two synchronization points, the system has to hold it until the next synchronization point. These errors generate unexpected time errors that may accumulate over time.

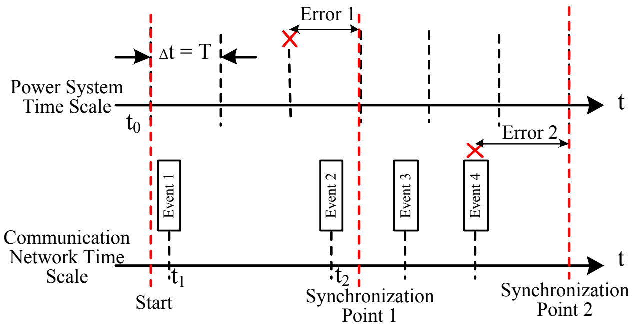


Fig. 3. Two types of synchronization errors - from [2]

To prevent the synchronization errors and increase the simulation accuracy, a sophisticated co-simulation scheme is illustrated in Fig. 4. Our co-simulation runs globally in a discrete event-driven manner. Since the power system dynamic simulation is in fact implemented in a discrete form as shown in Fig. 1, we consider each of the simulation round as a special discrete event in this framework. A global event queue is designated as a global time reference and coordinator. The global event queue is designed through sequencing the power system iteration events and communication network events according to the event timestamps. This scheme is illustrated in Fig. 4. By checking the content of the global event queue, the co-simulator knows whether the next event is a power system event or a communication network event. In addition, the entire simulation procedure is capable of suspending each event and yield a control back to the simulator. This method can effectively prevent the time delay errors introduced by synchronizations.

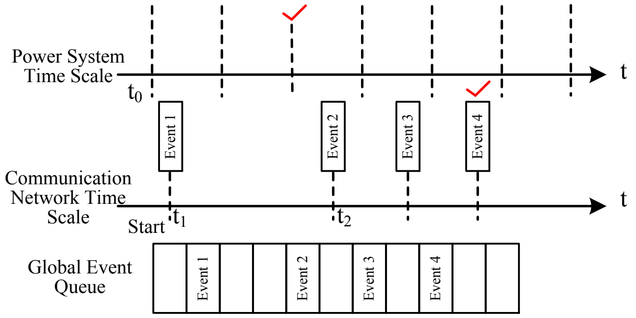


Fig. 4. Event-driven synchronization without errors - from [2]

## III. Implementation and System Construction

### A. GECO Framework Implementation

We implement our GECO framework by integrating two individual simulators: GE's Positive Sequence Load Flow (PSLF) and Network Simulator 2 (NS2). Fig. 5 shows the main structure of our co-simulation framework. The global scheduler and global event queue are derived directly from the counterpart in NS2. A bi-directional interface middleware is implemented between NS2 and PSLF to exchange information.

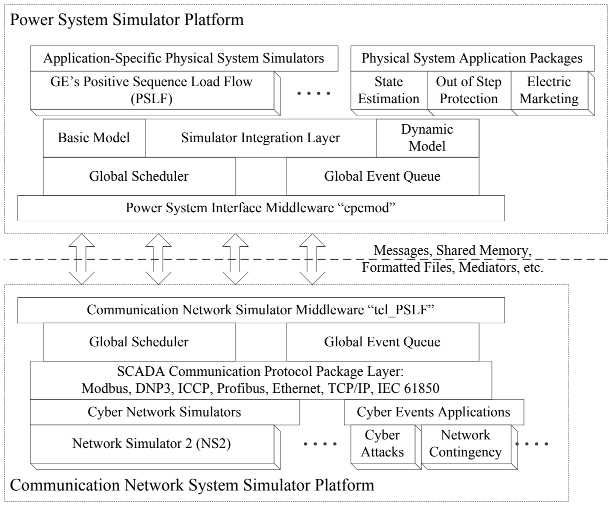

Power System Simulator Platform
Application-Specific Physical System Simulators
GE's Positive Sequence Load Flow (PSLF)
....
Physical System Application Packages
State Estimation
Out of Step Protection
Electric Marketing
Basic Model
Simulator Integration Layer
Dynamic Model
Global Scheduler
Global Event Queue
Power System Interface Middleware "epcmod"
Messages, Shared Memory, Formatted Files, Mediators, etc.
Communication Network Simulator Middleware "tcl_PSLF"
Global Scheduler
Global Event Queue
SCADA Communication Protocol Package Layer: Modbus, DNP3, ICCP, Profibus, Ethernet, TCP/IP, IEC 61850
Cyber Network Simulators
Cyber Events Applications
Network Simulator 2 (NS2)
....
Cyber Attacks
Network Contingency
....
Communication Network System Simulator Platform

Fig. 5. Co-simulation framework layered architecture

Within the power system simulator infrastructure, a power system interface middleware named "epcmod" is designed as a service function for NS2. During each iteration round, this middleware updates all the power measurements for NS2 and receives feedbacks from NS2 to change the settings of the power system accordingly. It has the ability to halt the PSLF simulation process, and coordinate with the global scheduler and wait for the command to run in the next round.

Within the network system simulator platform, we also designed an interactive middleware, which is implemented by C++ class "tcl_PSLF", to drive the simulation of PSLF and coordinate the actions in between. This class pre-allocated a sequence of the power system iteration and put them in the global event queue. When an iteration round needs processing, it sends the command to PSLF to restart the suspended simulation. The communication protocol packages, which are located in a higher layer of NS2, provide the network protocols used in power systems such as Modbus, DNP3, TCP/IP, etc.

### B. System for Co-Simulation Experiments

To evaluate the co-simulation framework, we choose the New England 39-bus system as the test bed for evaluating the following WAMS applications [11]. As shown in Fig. 6, there are 39 buses, 34 transmission lines, and 10 generators in the system. In our following experiments, we assume that PMUs are deployed in each bus and each generator to measure its local voltages, currents, phasors and rotor angles etc. More realistically, the entire system is divided into four local regions as denoted by the dotted lines in Fig. 6. Each local region is equipped with one Phasor Data Concentrator (PDC) to gather local PMU measurements. There are four PDCs, which are placed at bus 2, bus 6, bus 21, and bus 27 respectively. A Super PDC is assigned to be installed at bus 16 as the control center for all WAMS applications. PMUs, PDCs and the Super PDC are connected by an interconnected communication network. The topology of the communication network is assumed to be the same as the transmission lines. Meanwhile, network routers are placed at each bus to send, receive, and route measurements between devices.

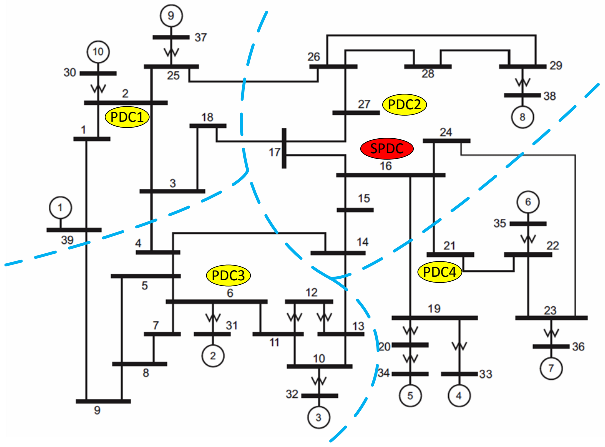


Fig. 6. WAMS on New England 39-bus system [11]

The 39-bus WAMS network is implemented in GECO, and other basic power system models and communication devices are implemented in PSLF and NS2 respectively. Some co-simulation settings are kept constant among all the different WAMS applications as summarized in Table I [8]. In addition, to ensure enough co-simulation accuracy, we set the smallest simulation step of PSLF to be 10ms in our experiments.

TABLE I. CASE I: CO-SIMULATION SETTINGS

| | | |
|---|---|---|
| Communication Link Bandwidth | $BW$ | 1 Gbps |
| Communication Link Delay | $D$ | 5 ms |
| PMU Measurement Rate | $\lambda$ | 30 times/sec |
| PDC Timer | $T_p$ | 50 ms |
| Super PDC Timer | $T_s$ | 50 ms |
| Phasor Packet Size | $S$ | 500 Bytes |
| PSLF Iteration Step | $Delta$ | 10 ms |
| PMU Measurement Error | $e$ | 1% |

## IV. CASE I: ALL-PMU STATE ESTIMATION

### A. *Case Description*

State estimation is the cornerstone of other WAMS applications such as contingency analysis, system integrity protection, electricity price prediction etc. In a traditional power system, a SCADA system is responsible for collecting unsynchronized power flows and transmission line currents for state estimator. Limited by the data acquisition period around 3-4 seconds, the state estimator of the power system is difficult to get accurate dynamic states [12][13]. Due to the non-linear properties of the state estimator, the state estimation process is an iterative flow and may have divergence issues.

With the help of PMUs, an all-PMU based state estimator can improve the deficiencies presented in the traditional ones. The distributed PMUs can receive GPS signals as a timing reference to synchronize acquisition data and can directly calculate the positive sequence voltage and current phasors at selected buses [12]. Since the minimum time resolution of a commercial GPS timing pulse should be less than 250 ns, in a 60 Hz power system environment, the phase angle error should be no more than 0.02 degrees. The updating rate of the PMUs is defined at 30 times per second. Meanwhile, the communication networks for all-PMU WAMS use advanced packet-switching mechanism, so that a complicated network topology may be deployed. Also, since the PMUs can directly measure the node states, the system state estimation process is largely simplified and faster [13].

### B. *Cyber-Attack Co-Simulation Results*

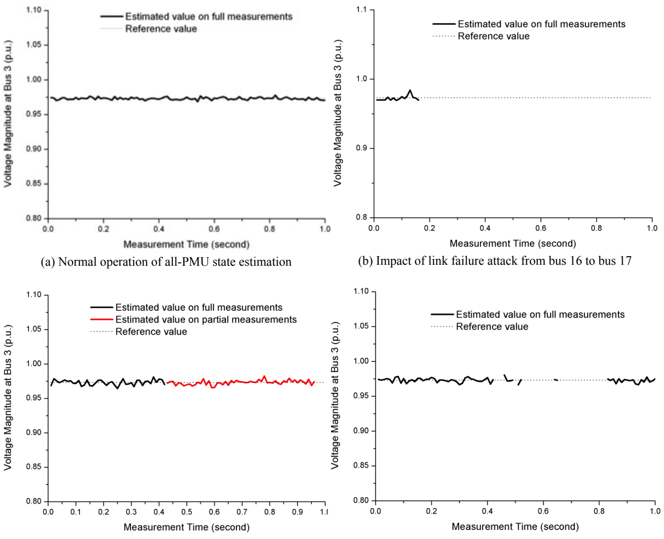
In this case study, one normal operation scenario and three cyber-attack scenarios are executed and verified on GECO to show the impacts of the communication network on the all-PMU state estimator. In simulation, we select the estimated voltage magnitude at bus 3 as the impact indicator by comparing the estimated results with the actual reference values (around 0.97 p.u.). The simulation durations of all scenarios are 1 second.

#### 1) *Normal Operation*

In Fig. 7 (a), the all-PMU state estimation results at bus 3 show the normal operation of the system. During co-simulation, we introduce small variations, which are random PMU measurement errors to make the simulation more realistic. There are no line fault attacks or network failures attacks in the system. We can see in the Fig. 7 (a) that the estimated voltage magnitude is very close to the real value calculated through system parameters in Table I.

#### 2) *Link Failure Attack*

In the link failure attack scenario, a communication link from bus 16 to bus 17 is blocked when the co-simulation procedure reaches at 0.2 seconds. The state estimation results shown in Fig. 7 (b) indicate the entire system states become unobservable after 0.2 seconds. The reason for no simulation output after 0.2 seconds is that the network connection from bus 16 to bus 17 is a critical path for establishing the measurement system. When this link is cut off by attackers, the routing scheme has to start a backup mechanism to find out an alternative but longer routing path for the measurements. After the new route is established, the communication delays for critical measurements increase such that they cannot arrive at the Super PDC before its timer expires. Therefore, the system becomes unobservable.



(a) Normal operation of all-PMU state estimation

(b) Impact of link failure attack from bus 16 to bus 17

(c) Impact of link saturation attack from bus 16 to bus 17

(d) Impact of DoS attack on the router at bus 16

Fig. 7. All-PMU state estimation cyber-attack co-simulation results

#### 3) *Link Saturation Attack*

Another common attack scenario is that the communication network is suffering from link saturation attack. In this experiment, a malicious traffic is injected into the link from bus 16 to bus 17 at 0.2 seconds of the co-simulation. From the estimation results shown in Fig. 7 (c), we can find that the malicious traffic does not affect the state estimator immediately. Instead, it gradually saturates the link and the impact starts to appear around 0.42 seconds. After that, the entire communication link is saturated and the effective measurements have to be stored and competed against the malicious traffic. Occasionally, some measurements will be discarded due to timer expiration. However, as we can see from the Fig. 7 (c), the system state can still be recovered after 0.42 seconds, and estimated from other redundant measurements. From the results in Fig. 7 (c), we can tell that the accuracy of the state estimation on partial measurements still satisfies. However, link congestions may also make the system unobservable. To resist this negative impact, advanced dynamic routing schemes could be implemented to auto-detect the saturation level of the communication link and proactively distribute the data flows.

### 4) *Denial of Service (DoS) Attack*

The DoS attack depletes the available resources of the target by deliberately generating large amount of redundant data or inquiries. In the all-PMU state estimator system, DoS attacks can occupy partial computational power of critical gateway routers leading to packets dropping and longer delay. In DoS attack scenario, we assume that 10 compromised computers within the system start to send malicious data to the router located at bus 16 to deplete its resources at 0.2 seconds. The simulation results in Fig. 7 (d) show that the behavior of the state estimator becomes intermittent starting at 0.4 seconds. The system state switches between unobservable and observable. Due to the large amount of DoS attack at the router in a short period of time, the router can be overloaded and the packet dropping may occur. In the worst case, the system can be unobservable and non-recoverable. A prominent solution to prevent DoS attack is to setup backup routers in a dual-router structure. Associated preventing schemes such as the malicious traffic detecting and filtering or labeling the data packets with priorities will also increase the robustness of the system.

## V. Case II: PMU-Based Out-of-Step Protection

### A. *Case Description*

To prevent the power system equipment from accidental damages, relays and other protection devices may proactively perform protection schemes. Out of Step (OOS) may occur when the rotor angles of one generator or a group of generators separate from the rest of the system. If such groups of generator cannot isolate from the power system timely, it would result in large mechanical vibrations that could potentially damage these generators [14]. By observing the rotor angles at the interface between these out-of-synchronous areas and the rest of the network, there are huge transient swings. Thus, in traditional OOS protection schemes, relays need to detect these swings and open the tie lines [15]. With the availability of PMUs and WAMS, even in the event of drastic change in operating points, these schemes can be made adaptive so that they are both dependable and secure.

A PMU based adaptive OOS scheme has been proposed in [16][17]. In this scheme, the rotor angles of all the major generators are monitored by distributed PMUs and collected into a central control center. If a major disturbance occurs, coherent groups of generators are recognized using the trajectory of the rotor angles. When the centers of the rotor angles of these coherent groups distinguish by over 120 degree, OOS protection operations are performed to island the two out of synchronous systems. In order to develop and assess the OOS protection scheme, we implement the whole OOS process on our co-simulation platform.

### B. *Co-Simulation Results*

A set of OOS protection system is implemented in GECO based on the WAMS in Fig. 6. PMUs are placed at each generator bus to directly measure the rotor angles. A central OOS controller is equipped in the Super PDC at bus 16 to collectively monitor the rotor angle trajectories. To simulate an OOS condition, a sequence of events provided in [18] is used. A three-phase fault is initialized on line 21-22. After clearing the fault, machines on bus 35 and 36 lose synchronism. By the nature of their rotor angle trajectory, it is concluded that these two machines form a coherent group of machines. The rotor angles of all the machines are recorded and the centers of angles of all the coherent groups are calculated. When the difference between the centers of angles reaches 120 degree, existence of OOS conditions is confirmed and the lines 16-24 and 16-21 are opened, forming an island consisting of generators on bus 35 and 36. The circuit breaker operating times are not included in this co-simulation.

#### 1) *The OOS condition in normal condition*

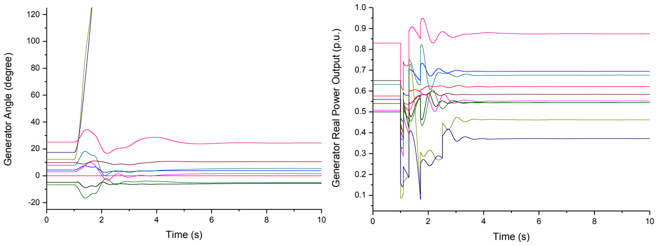


Fig. 8. (a) Generator angels showing OOS condition (BW=1 Gbps, D=5 ms); (b) Generator real power outputs (BW=1 Gbps, D=5 ms)

The reaction of the generators to the events described above can be observed in the real power outputs and rotor angles of the generators. When the communication bandwidth is set to 1 Gbps and delay is 5 ms, these co-simulation results of the OOS protection are shown in Fig. 8 (a) and (b). After placing the three-phase fault at 1.0 second and the subsequent fault clearing by opening line 21-22 around 1.1 seconds, the generators experience transient oscillations. The rotor angle plots in Fig. 8 (a) show that the generators on bus 35 and 36 split from the rest of the generators. The phenomenon is recorded by the central controller and an islanding command is issued to the circuit breakers at line 16-24 and 16-21. These two lines are finally open around 1.71 seconds. After the OOS separation, the rotor angle trajectories of these two generators separate faster from the rest of the system. This event can be observed in the power output of the generators as a spike at the same time, as shown in Fig. 8 (b). The real output of the power will come back to a stable condition at 4.95 seconds after a series of oscillations. After separation, the bigger island operates at a reduced frequency of 59.87 Hz, while the smaller island is over-generated resulting in a frequency of 60.6 Hz.

#### 2) *The OOS condition on an inferior network*

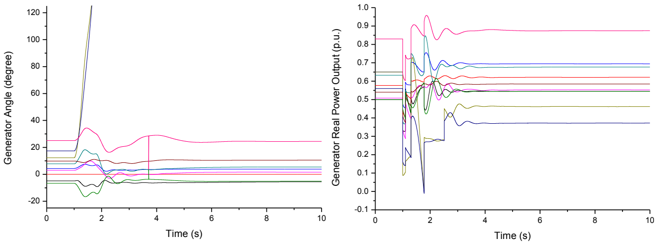


Fig. 9. (a) Generator angels (BW=100Mbps, D=10ms); (b) Generator real power outputs (BW=100Mbps, D=10ms)

To assess the cyber impacts on the OOS protection scheme, we do an experiment on an inferior communication network where the communication bandwidth is decreased to 100Mbps and the transmission delay keeps 10ms for each link. The co-simulation results used this condition are plotted in Fig. 9 (a) and (b). The simulation results show that the OOS protection scheme can still restore the system but with a slower response. The OOS separation occurs at 1.77 seconds and the system comes back to stable around 5.02 seconds. This slower response results in larger spike compared to Fig. 8 (b) which could potentially damage system.

### *3) The OOS condition with link failure*

Even if for the same network setting with communication link failure being considered, the results will be completely different. When a communication link failure follows the short circuit fault, the rotor angles and the real power of the generators are shown in Fig. 10 (a) and (b). It can be inferred from Fig. 10 (b) that the OOS protection fails to form the island. This is because the average phasor delay will be increased when the network loses an important path. Some of the angle measurements cannot arrive at the central controller before its timer expires. To solve this problem, we should improve the communication network or longer timeout setting of timer to secure the OOS protection scheme under this scenario.

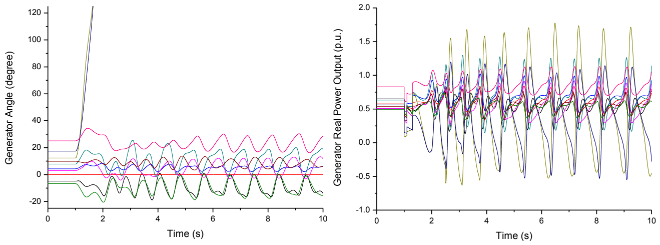


Fig. 10. (a) Generator angels with link failure (BW=100Mbps, D=10ms); (b) Generator real power outputs with link failure (BW=100Mbps, D=10ms)

## VI. Conclusion

In this paper, we describe two commonly used simulation mechanisms: dynamic simulation for power systems and event-driven simulation for network systems. Then we analyze the root cause of synchronization errors existing in most co-simulation platforms and review our global event-driven co-simulation framework GECO, which is able to provide better synchronization accuracy. The co-simulation framework then is applied in two WAMS applications to test the validity of the framework. In the first case study, an all-PMU state estimation system is implemented on the New England 39-bus system. Potential network failures and cyber-attacks have been applied to the all-PMU state estimator. The co-simulation results reveal that the state estimator is vulnerable to link failure, router-saturation, and DoS attacks. In the second case study, the co-simulation framework is used to help developing an OOS protection scheme. Generator angles tracking functions are implemented in the co-simulation platform and the OOS criteria are assessed associated with certain network environments.

## VII. References

[1] A. G. Phadke and R. M. de Moraes, "The Wide World of Wide-area Measurement," Power and Energy Magazine, IEEE, vol. 6, pp. 52-65, 2008.

[2] H. Lin, S. Veda, S. Shukla, L. Mili, and J. Thorp, "GECO: Global Event-Driven Co-Simulation Framework for Interconnected Power System and Communication Network," Smart Grid, IEEE Transactions on, vol. PP, pp. 1-13, 2012.

[3] K. Hopkinson, W. Xiaoru, 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," Power Systems, IEEE Transactions on, vol. 21, pp. 548-558, 2006.

[4] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated Hybrid-Simulation of Electric Power and Communications Systems," in Power Engineering Society General Meeting, 2007. IEEE, 2007, pp. 1-8.

[5] K. Zhu, M. Chenine, and L. Nordstrom, "ICT Architecture Impact on Wide Area Monitoring and Control Systems' Reliability," *Power Delivery, IEEE Transactions on,* vol. PP, pp. 1-1, 2011.

[6] W. Li, A. Monti, M. Luo, and R. A. Dougal, "VPNET: A co-simulation framework for analyzing communication channel effects on power systems," in *Electric Ship Technologies Symposium (ESTS), 2011 IEEE*, 2011, pp. 143-149.

[7] C. M. Davis, J. E. Tate, H. Okhravi, C. Grier, T. J. Overbye, and D. Nicol, "SCADA Cyber Security Testbed Development," in *Power Symposium, 2006. NAPS 2006. 38th North American*, 2006, pp. 483-488.

[8] Hua Lin, Yi Deng, Sandeep Shukla, James Thorp, Lamine Mili, "Cyber Security Impacts on All-PMU State Estimator – A Case Study on Co-Simulation Platform GECO," SmartGridComm, Taiwan, 2012.

[9] L. Hua, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, 2011, pp. 1-6.

[10] S. S. Hua Lin, Sandeep Shukla, James Thorp, Lamine Mili, "A Study of Communication and Power System Infrastructure Interdependence on PMU-based Wide Area Monitoring and Protection," PES General Meeting, San Diego, 2012.

[11] I. Hiskens, "Significance of Load Modeling in Power System dynamics," presented at the x symposium of specialists in electric operational and expansion planning, 2006.

[12] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, "Synchronized Phasor Measurement Applications in Power Systems," Smart Grid, IEEE Transactions on, vol. 1, pp. 20-27, 2010.

[13] K. Jones, "Three-Phase Linear State Estimation with Phasor Measurements," Master of Science, Electrical and Computer Engineering, Virginia Tech, Blacksburg, 2011.

[14] J. De La Ree, Y. Liu, L. Mili, A. G. Phadke, and L. DaSilva, "Catastrophic Failures in Power Systems: Causes, Analyses, and Countermeasures," Proceedings of the IEEE, vol. 93, pp. 956-964, 2005.

[15] L. Li and L. Yutian, "Out-of-step splitting framework based on adaptive separation detecting criterion," in Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009, 2009, pp. 1-5.

[16] A. G. Phadke and J. S. Thorp, "Communication needs for Wide Area Measurement applications," in Critical Infrastructure (CRIS), 2010 5th International Conference on, 2010, pp. 1-7.

[17] F. G. V. Cedeno, "Multiple Swing Out-of-Step Relaying," Doctor of Philosophy, Electrical Engineering, Virginia Tech, 2010.

[18] A. Sauhats, J. Kucajevs, L. Leite, G. Bockarjova, and A. Utans, "Out-of-step automation device model validation methodology," in Environment and Electrical Engineering (EEEIC), 2011 10th International Conference on, 2011, pp. 1-6.

# Communication Simulations for Power System Applications

J. C. Fuller, S. Ciraci, J. A. Daily, A. R. Fisher, M. Hauer
Pacific Northwest National Laboratory
Richland, WA

*Abstract*— **New smart grid technologies and concepts, such as dynamic pricing, demand response, dynamic state estimation, and wide area monitoring, protection, and control, are expected to require considerable communication resources. As the cost of retrofit can be high, future power grids will require the integration of high-speed, secure connections with legacy communication systems, while still providing adequate system control and security. The co-simulation of communication and power systems will become more important as the two systems become more inter-related. This paper will discuss ongoing work at Pacific Northwest National Laboratory to create a flexible, high-speed power and communication system co-simulator for smart grid applications. The framework for the software will be described, including architecture considerations for modular, high performance computing and large-scale scalability (serialization, load balancing, partitioning, cross-platform support, etc.). The current simulator supports the ns-3 (telecommunications) and GridLAB-D (distribution systems) simulators. A test case using the co-simulator, utilizing a transactive demand response system created for the Olympic Peninsula and AEP gridSMART demonstrations, requiring two-way communication between distributed and centralized market devices, will be used to demonstrate the value and intended purpose of the co-simulation environment.**

***Keywords—co-simulation; modeling; power systems; communication systems; distribution; smart grid***

## I. Introduction

Smart grid technologies and concepts, such as demand response (DR) and wide area monitoring, protection, and control (WAMPAC), and new sensing and measurement equipment, such as phasor measurement units (PMU) and advanced metering infrastructures (AMI), are expected to provide unique insights into the performance and real-time operations of the U.S. electrical system. However, these applications are also expected to require relatively intensive data transfer and considerable communication resources as compared to current resources. Future power grid operations will require the seamless and secure integration of legacy communication systems with new communication technologies and protocols, as the cost of system-wide upgrades is prohibitive. As traditional power systems transition into more dynamic systems where both energy and data flows are managed and transacted in a more distributed manner, moving away from classical, centralized control architectures, decision makers need to understand how investments in communication systems can be leveraged across multiple applications to reduce cost.

It is generally understood that high bandwidth communications are required to overlay the grid of the future, especially to enable high-speed, wide area control and protection [1]. However, not all applications require continuous, high-speed communications (e.g., AMI networks may need to only update billing information once per day). Understanding the requirements of the desired applications, and whether dedicated communications are required versus a shared network configuration, can help utilities reduce long term investment and costs, and accelerate smart grid technology deployment. One of the main barriers to this has been the lack of design tools that are capable of simulating power and communication systems in a single environment, as the performance of each affects the other.

Some work has already been done in this area. The GridSim simulator was designed to co-simulate power (TSAT) and communication (GridStat) systems for wide area control and protection applications [1]. This tool was designed to run in real-time. EPOCHS was developed by [2] to link the PSCAD/EMTDC electromagnetic transient simulator with the PSLF electromechanical transient simulator and ns-2 communication simulator. This work was based on integrating off-the-shelf tools to understand the effects of communication systems on electromechanical scenarios. Baran, et al., extended the PSCAD/EMTDC simulator to simulate agent-based distributed protection and control applications [3]. Nutaro, et al., have developed a co-simulator to investigate hybrid discrete and continuous, power and communication simulations using THYME and ns-2 [4][5]. This framework has also been used to investigate the communication effects on real-time markets and frequency sensing and actuating loads [6][7]. Lin, et al., developed a co-simulator using PSLF and ns-2 to evaluate an agent-based remote backup relay system [8].

Each of these applications has created a co-simulation environment to understand the effects of communication system impacts on transmission related issues. However, for the most part, the effects of communication on distribution systems have been ignored. Distribution systems model power flows in a different manner than transmission systems, focusing more on unbalanced load flows and localized control and protection issues, requiring radically different assumptions. A majority of new smart grid applications, from AMI to DR to self-healing feeder reconfiguration, are being deployed and controlled at the distribution level. The aforementioned tools are not capable of evaluating these network applications, as they currently stand. Some work has been performed using

OpenDSS to co-simulate with OMNet++ to understand how a low-cost communications infrastructure affects plug-in electric vehicle (PEV) charge scheduling for voltage stabilization [9]. OpenDSS has also been co-simulated with ns-2 to evaluate whether community energy storage (CES) devices can respond fast enough to adjust for “cloud transients” in high penetration distributed photovoltaic systems [10]. These works have shown some of the consequences of not properly accounting for the latency in control signals in simulation. However, there are a number of smart grid applications that these tools are not capable of evaluating.

As part of a three year project within the Future Power Grid Initiative (FPGI) at the Pacific Northwest National Laboratory, work is being performed to address some of these gaps [11]. The Next Generation Network Simulator is being created as a platform for integrating communication and power system simulators across multiple computing elements, so that simulations are highly scalable. The end goal is to create a simulation environment capable of integrating discrete event based power and communication simulators, capturing detailed behaviors from the lowest level of power systems (end use load) all the way to the generation plant, and all of the communication systems in between.

This paper will be organized as follows. In Section II, the general framework will be described. In Section III, specifics of the software implementation will be discussed, including current development progress. In Section IV, an example model will be explored, using an actual test case from the deployment of a real-time demand response pilot as part of the AEP gridSMART® Demonstration Project. Section V will provide conclusions and discuss future directions.

## II. Integration Framework

The framework is designed to integrate and allow the sharing of information across multiple simulation domains, specifically for event-driven, steady-state, time-series analysis. From a user’s perspective, this should be as simple as creating models in each of the domain specific environments, and then using a simple configuration to define the linkages between the models. To model and simulate a system of this complexity, the framework shall need to support certain key requirements:

(1) **Scalability**. Simulations will be inherently large and dynamic, and users are expected to seek out advanced computing options. The framework will support hundreds to thousands of components simultaneously.

(2) **Simulation speed**. Large-scale simulations are expected to result in significant simulation times. Reducing this time is of upmost importance. Partitioning and load balancing techniques will be employed to maximize performance across widely dispersed computing options, with a focus on distributed computing.

(3) **Time synchronization**. Each of the domain specific simulators will require different time scales and time steps to perform their functions. The framework will provide a means for integrating time scales using event-based queues and a “heartbeat” to keep each of the simulators synchronized in time. When appropriate, simulations will move ahead independently until synchronization is required.

(4) **Model heterogeneity**. Models from different domains are expected to use a variety of solution techniques. The framework will handle the widely disparate computational workloads between different domains.

(5) **Output and diagnostics**. While each of the component simulators is expected to handle its own output of domain specific data, the framework will need to provide a means for users to access the flow of information through the framework.

(6) **Fault resiliency**. Applications on large scale computational networks must be resilient to system failures. A “checkpoint” system is required to periodically save and restart the simulation from a saved point with minimum loss of data.

(7) **Serialization and De-serialization of Objects**. The ability to transfer messages across components distributed on different machines, check-pointing, and restarting the simulation state in an object oriented code all requires the ability to serialize and de-serialize objects. Writing serialization and de-serialization methods and functions on a large code base can be time consuming and error prone. A simulation framework should provide an infrastructure to facilitate this task.

(8) **Modularization and Intellectual Property Protection**. Frameworks and the simulated components must be decoupled by design. The framework has to be able to instantiate different components that are not necessarily maintained and developed together with the simulation engine. Components should be compiled standalone (without requiring the framework) to allow distribution as dynamic libraries. This could allow developers to protect their software properties by encrypting the compiled object and distributing them without source code. A set of fixed APIs should be exposed to allow this mechanism, and a language that allows flexibility, speed, and modularization such as C++ should be used.

A conceptual model of an example simulation setup is shown in Figure 1. This particular model would need to support a transmission simulation, multiple distribution simulations, and multiple communication simulations. A user would need to create the individual models in each domain then define the interconnections between the models. For example, the distribution solver would need to share the total feeder demand at a given moment in time with the transmission solver to replace the static substation load. The transmission solver, in turn, would need to communicate the current voltage at that node. The serialization and de-serialization of this information is handled by “wrapper” modules created for each new simulation tool introduced to the integration framework. The wrapper module contains mechanisms for translating time requirements between the framework and the simulator, converting message information into proper formats, and accessing/modifying the requisite data in each simulator.

Currently, the integration framework is in various stages of development to support the given requirements. The next section describes an initial integration using two specific simulators, ns-3 for communication systems and GridLAB-D for power systems.

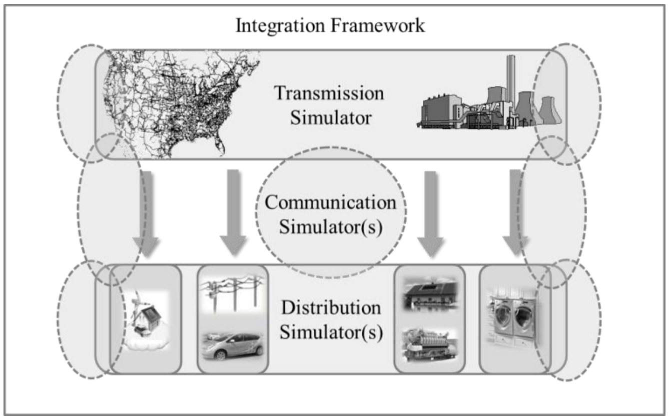


Fig. 1. Conceptual model of integration framework

## III. Current Implementation

The two main requirements for integrating GridLAB-D and ns-3 (and other simulators) are reuse and separation. The goal is to make the modules of both simulators available for integrated simulations, and where necessary, allow the different simulators to pass information into or out of one another. In this way, users can implement and experiment with different power and communication hardware/protocol combinations.

It is possible to integrate the simulators by adapting the modules of one simulator for use with the other one. Unfortunately, such adaptations can complicate the code base of the simulators, can introduce errors, and often recreates work than has already been done. Hence, it is important to keep the code bases of both simulators separate, allowing for independent but parallel development. For example, modules of ns-3 could be adapted to work with GridLAB-D by directly linking the ns-3 functionality through the core operations of GridLAB-D. However, this complicates the code base of GridLAB-D, as the developers also have to maintain these modules. Moreover, the adaptations can introduce errors to the existing code leading to crashes or incorrect results.

Considering these requirements integration middleware was implemented, in which both simulators run independently and communicate only when they need to exchange messages. The middleware facilitates the message exchange and time synchronization. Figure 2 provides a typical setup of an integrated simulation with this middleware.

The middleware consists of three modules: interface, network application, and time synchronization. The interface module is integrated to the GridLAB-D simulator. It provides methods for sending and receiving messages and interpreting those messages into the format required by GridLAB-D. These methods are used by other GridLAB-D modules for exchanging messages. An important aspect of these methods is that they hide all of the details of the communication network making it easier to implement integrated simulation models, allowing domain experts to develop models within familiar environments.

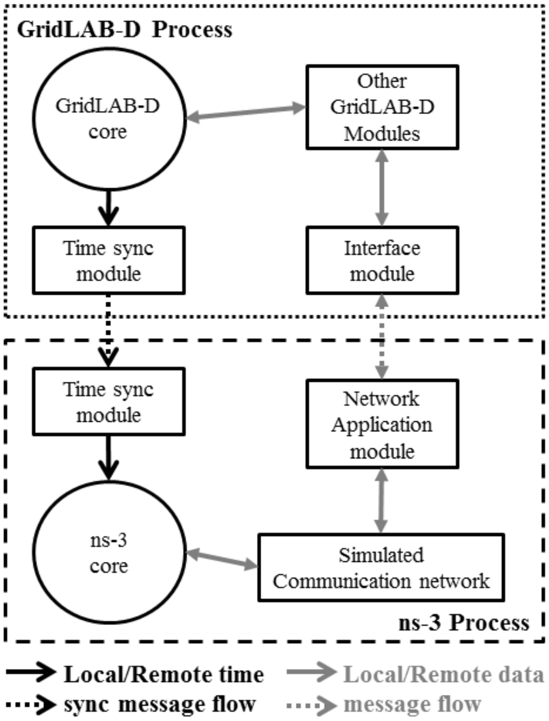


Fig. 2. Integration middleware and various modules

The network-application module is integrated with the ns-3 simulator. It is responsible for setting up the communication network. It builds the routing table and tries to re-establish broken connections. The network-application module also acts as the bridge between the interface module and the communication network. It passes the messages from the interface module to the communication network and forwards the messages sent by the communication network to the interface module.

The time synchronization module keeps the internal times of both simulators in sync. This is a critical operation in order to coordinate the message exchanges. For example, a message sent by a GridLAB-D module at time $t$ should also be forwarded to the communication network at time $t$. If ns-3's internal time is less than $t$, then the message can be forwarded back to GridLAB-D at $t_1 < t$. This may result in errors during calculations.

The time synchronization module keeps counters for the number of sent and received messages. If the number of sent messages is equal to the number of received messages, then there are no messages in transit. Hence, it allows both simulators to progress in time independently. If, on the other hand, the number of received messages is less than the number of sent messages, then there are messages in transit. The synchronization module brings each simulator to the same time step and allows them to progress at most one step (e.g., 1 second).

It is possible for the communication network to "lose" packets (e.g., by simulating network errors). In such a case, the counters for sent and receive messages would never be equal. Therefore, the synchronization algorithm described above would force the simulators to progress one step until completion, severely impacting the performance. To prevent this, the time synchronization module was programmed to declare packets as lost when the sent and received counters do not stabilize after *n* time steps after it is sent. The user can set the value for *n*.

A de-/serialization code generator was created as part of the framework. The users just to need to point out which classes (or specific attributes) of the GridLAB-D modules need to be serialized for the messages that are sent to ns-3. An important aspect of the code generator is the ability to automatically extract the properties required for serializing attributes. For example, it can trace pointers and automatically identify the size allocated for a dynamic array. Automatic extraction of properties is required so that users without knowledge of GridLAB-D code can integrate ns-3 with custom GridLAB-D modules. The code generator is implemented as part of the C++ compiler, and thus it can be easily integrated to build procedures [12].

## IV. Demand Response Example

A demand response (DR) use case is investigated using the integrated simulation environment. GridLAB-D is capable of simulating a DR strategy known as transactive control, first demonstrated on the Olympic Peninsula [13] and now being used as part of the AEP-Ohio gridSMART™ Demonstration Project [14].

The control system works by creating a near real-time, double-auction, retail market applied at the distribution level. A double-auction market is a traditional market strategy that can be described as a two-way market, where both suppliers and end-use loads submit bids for price and quantity into a single energy market simultaneously. The auction resolves the supply and demand bids into a common cleared market price and quantity, and delivers this information back to the participants. The supply curve is defined by the Locational Marginal Price (LMP) and the capacity limit of the distribution feeder substation. The demand curve is formulated from the devices on the system as shown in Figure 3.

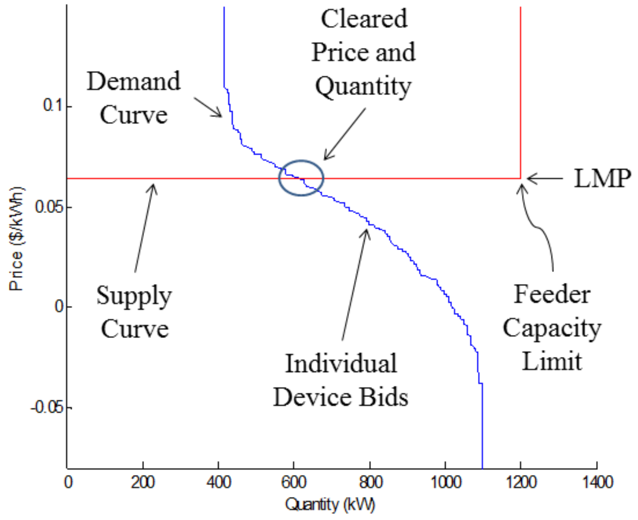


Fig. 3. Demand and supply curve formulated in double-auction market – a single supply bid represents the LMP and the feeder capacity constraints, while the demand curve is made up of hundreds of bids from AC units

Devices within the customer residence translate consumer desire for comfort into a bid price and quantity. This information is sent to a centralized “auction” every five minutes (minimally). In the case of a smart thermostat, the customer is able to set their desired thermostat setpoint and their preference for “savings” versus “comfort” on a simple slider that limits the setpoint adjustment allowed. The customer’s thermostat responds to the market clearing price broadcasted by the auction and adjusts the energy consumption as a function of that price (high prices equate to lower energy consumption). The goal is to settle the double-auction market every five minutes to limit the amount of energy delivered to the feeder and reduce wholesale energy costs. More detailed explanations about the control method can be found in [13][15][16]. The communication requirements of the system are as follows (shown pictorially in Figure 4):

1) The auction determines the cleared price and broadcasts this and historical information to all controllers on the system (100s-1000s of devices per feeder).
2) The controllers receive the information and translate the price to adjust the thermostatic setpoint. It then forms a bid using the historical data and current local states and transmits to the auction.
    a) If the thermostat has a state change (ON/OFF or consumer updates preferences), a new bid is transmitted using the id generated in (3).
3) The auction transmits a receipt record to each controller with a unique identifier after it receives a bid.

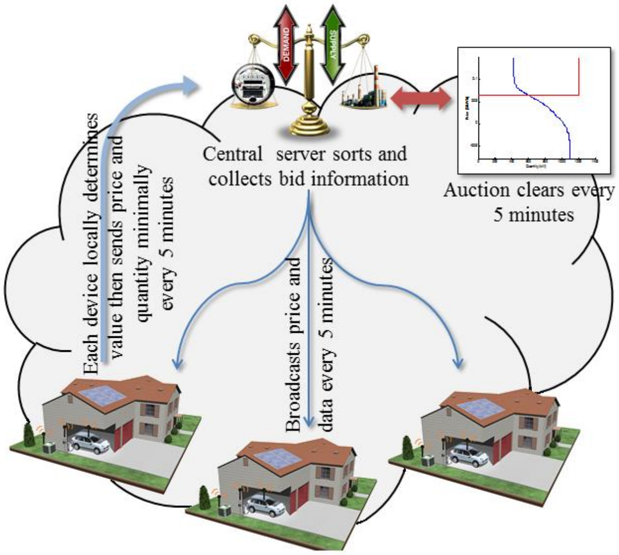


Fig. 4. Pictorial description of double-auction transactive market

Steps 1-3 are repeated every five minutes as each new market is formed. Step 2a is repeated as many times as necessary within the five minute market cycle, and is used when the Air Conditioning (AC) system switches current states (e.g., on to off or vice versa) or the consumer makes a manual modification to the setpoints (e.g., turns the thermostat up a couple degrees). Five minute market intervals allow the cycling behavior of the controllable loads to participate in load reduction and load recovery, as this is close to the natural duty cycle of major loads (mainly ACs and water heaters) and exploits natural behavior [13]. Shorter time intervals may "short-cycle" the devices, causing long term degradation of components, while longer time intervals (such as those seen in the bulk power system) decrease the dependability of individual loads to perform as bid over the given time interval. Longer time intervals, say 15 minutes, may be used, but five minutes was chosen as this is the same time interval used in both demonstration projects.

Previous work with GridLAB-D to evaluate the performance of this control system has made the assumption that communication delays are non-existent and information between the different elements are exchanged near instantaneously [16]. However, this is obviously not the case in an actual system. In some cases, it may be expected that a system like this would be used over existing communication infrastructure, such as an Advanced Metering Infrastructure (AMI). To evaluate the effects of communication delay on this control system, the integrated simulation engine discussed in Section III was used. A GridLAB-D power system model was created using a representative distribution feeder, the IEEE 13-node system [17]. The static loads were replaced with thermodynamic building models, as discussed in [18][19], resulting in 900 individual residential building objects. Each building had its own thermostatic control. In the base simulation (Fixed price), none of the homes were engaged in demand response. In the response cases, 600 of the homes were simulated as a control group, i.e., no price response, while the last 300 were part of a double-auction market.

In the first response case, no communication delays were assumed. In the second, the message packets from each of the AC controllers were routed through a simple radial WIFI network in ns-3 to the auction. Communication delays were increased until there was a noticeable effect on the cleared market price. It should be noted that this is a contrived case to explore the simulation environment and not to point out deficiencies in the demand response control. However, this type of signal delay may be representative of an undersized communication system, or one that is overlapped with other control systems. An adverse reaction within the control system occurred when the average round trip delay reached 100 seconds. Results from these simulations are shown in Figure 5. Only the six hour window of interest is shown here.

At approximately 4:30 pm, the total feeder load approaches the pre-defined capacity limit of the market. In the response cases, the cleared market price increases to encourage more demand response and flatten the demand at the capacity limit. Of note in the context of the communication delay, however, is the effect at 6:30 pm. Note the difference between the "no delay" and "with delay" power demand. While this is not significant, do note the effect it has on the price; in the "with delay" case, the cleared market price jumps to the price cap, greatly affecting the cleared price (~0.10 $/kWh to ~4 $/kWh) for nearly an hour and a half. This is caused by a significant portion of the bids being delayed, particularly the re-bids later in the market cycle. This is driven by the fact that communication requirements (re-bids) increase during stressed system conditions. Because the re-bids within the cycle do not actually make it to the auction before clearing, the auction uses out-of-date information to formulate the bid curve. While in most market clearings this minor error does not affect the outcome, because of the already high prices and relatively stressed system at 6:30 pm, the error becomes large enough to affect the outcome. In this case, consumers could be exposed to relatively high energy prices (~4 $/kWh) due to a communication delay within the control system, not due to a scarcity in energy production.

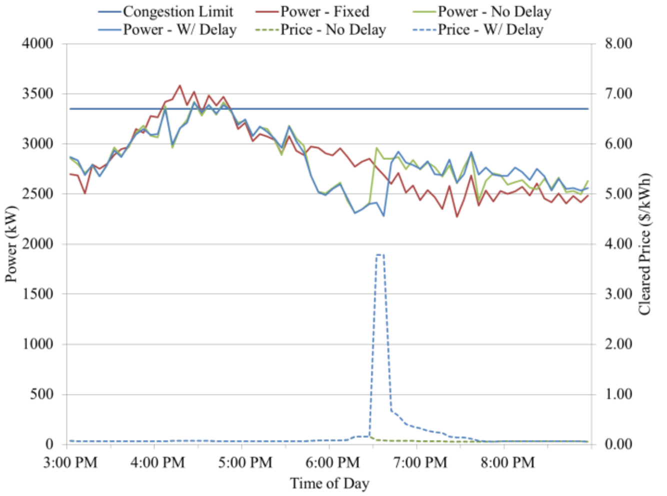


Fig. 5. Comparison of simulated power demand and cleared market price with and without communication delays

It is understood that this is a relatively extreme example, using a fairly contrived signal delay, and is not expected to occur in actual deployments. However, it was designed to highlight some of the extra design steps and considerations that may need to be addressed when designing a demand response control system, especially one that requires closed loop control and bi-directional flows of information in near real-time. It also highlights the need for co-simulation environments that can evaluate the co-dependent effects of power and communication systems. In turn, a co-simulation environment such as this could be used to help determine the most economic means of deploying smart grid technologies, specifically in terms of communication requirements for successful system operations. For example, in this case, it is clear that a communication system with greater than 100 second round-trip delay is unacceptable (i.e., most AMI networks would not support it). This knowledge can be used to inform the system planners about the potential cost (versus benefit) of the proposed demand response technology.

## V. Conclusions and Future Work

This paper presents an integrated communication and power system simulation engine. The current framework supports some functions of GridLAB-D and ns-3, and an example test case using demand response is presented here. Continuing work will focus on expanding the general capabilities of the integration framework, generalization of serialization/de-serialization processes, creation of a more generic GridLAB-D interface, incorporation of a transmission solver (both sub-second machine dynamics and static power flow), and enhanced support for distributed computing.

## Acknowledgments

The authors of this paper would like to thank Henry Huang, Jeff Dagle, and Bora Akyol of the FPGI leadership team for funding and supporting this work.

## References

[1] Anderson, D., Chuanlin Zhao; Hauser, C., Venkatasubramanian, V., Bakken, D. and Bose, A. Intelligent Design Real-Time Simulation for Smart Grid Control and Communications Design. Power and Energy Magazine, IEEE, vol.10, no.1, pp.49-57, Jan.-Feb. 2012.

[2] Hopkinson, K., Xiaoru Wang, Giovanini, R., Thorp, J., Birman, K. and Coury, D. EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. IEEE Transactions on Power Systems, vol.21, no.2, pp. 548- 558, May 2006.

[3] Baran, M., Sreenath, R. and Mahajan, N.R. Extending PSCAD/EMTDC for simulating agent-based distributed applications. IEEE Power Engineering Review, vol. 22, no. 12, pp. 52-54, 2002.

[4] Nutaro, J. Designing power system simulators for the smart grid: Combining controls, communications, and electro-mechanical dynamics. 2011 IEEE Power and Energy Society General Meeting, vol., no., pp.1-5, 24-29 July 2011.

[5] Nutaro, J., Kuraganti, T. and Shankar, M. Seamless Simulation of Hybrid Systems with Discrete Event Software Packages. 2007 Simulation Symposium. ANSS '07. 40th Annual, vol., no., pp.81-87, 26-28 March 2007.

[6] Nutaro, J. and Protopopescu, V. Discrete sensing and actuation in a simulation of frequency responsive loads. 2012 IEEE Energytech , vol., no., pp.1-6, 29-31 May 2012.

[7] Nutaro, J. and Protopopescu, V. The Impact of Market Clearing Time and Price Signal Delay on the Stability of Electric Power Markets. IEEE Transactions on Power Systems, vol.24, no.3, pp.1337-1345, Aug. 2009.

[8] Lin, H., Sambamoorthy, S., Shukla, S., Thorp, J. and Mili, L. Power system and communication network co-simulation for smart grid applications. Proc. IEEE PES Conf. on Innovative Smart Grid Technologies, Anaheim, 2011, pp.1-6.

[9] Levesque, M., Xu, D. Q., Joos, G. and Maier, M. Co-Simulation of PEV coordination schemes over a FiWi Smart Grid communications infrastructure. IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, vol., no., pp.2901-2906, 25-28 Oct. 2012.

[10] Godfrey, T., Mullen, S., Dugan, R.C., Rodine, C., Griffith, D.W., and Golmie, N. 2010 First IEEE International Conference on Modeling Smart Grid Applications with Co-Simulation. Smart Grid Communications, vol., no., pp.291-296, 4-6 Oct. 2010.

[11] Future Power Grid Initiative. [Online]. Available: http://gridoptics.pnnl.gov

[12] Ciraci, S., Villa O. Exploting points-to maps for de-/serialization code generation. To appear in proceedings of ACM Symposium on Applied Computing, 2013.

[13] Hammerstrom, D. J., et. al., “Pacific Northwest GridWise Testbed Demonstration Projects: Part I. Olympic Peninsula Project”, Technical Report PNNL-17167, Pacific Northwest National Laboratories, Richland, WA, 2007.

[14] gridSMART Demonstration Project. [Online]. Available: http://www.gridsmartohio.com/

[15] Pu Huang, et. al., “Analytics and Transactive Control Design for the Pacific Northwest Smart Grid Demonstration Project,” in Proc. First IEEE Int’l Conf. on Smart Grid Communications, p. 449, Oct. 2010.

[16] Fuller, J. C., Schneider, K. P., Chassin, D. Analysis of Residential Demand Response and double-auction markets. 2011 IEEE Power and Energy Society General Meeting, vol., no., pp.1-7. 24-29 July 2011.

[17] Kersting, W.H. Radial distribution test feeders. IEEE Transactions on Power Systems , vol.6, no.3, pp.975-985. Aug 1991.

[18] Schneider, K. P., Fuller, J. C., Chassin, D. P. Multi-State Load Models for Distribution System Analysis. IEEE Transactions on Power Systems, vol.26, no.4, pp.2425-2433. Nov. 2011.

[19] Fuller, J. C., Prakash Kumar, N., and Bonebrake, C. A. Evaluation of Smart Grid Investment Grant Project Technologies: Demand Response. PNNL-20772. Pacific Northwest National Laboratory, Richland, WA, 2011.

# Modelica-Enabled Rapid Prototyping of Cyber-Physical Energy Systems Via The Functional Mockup Interface

Atiyah Elsheikh, *Member, IEEE,* Muhammed Usman Awais, Edmund Widl, *Member, IEEE,*
Peter Palensky, *Senior Member, IEEE*

Austrian Institute of Technology, Energy Department, Vienna, Austria

{givenname.surname}@ait.ac.at

***Abstract*—Modelica has achieved a great success in the last decade. Universal modeling concepts, object-oriented facilities and large set of libraries in several physical domains allow for rapid prototyping of multidisciplinary applications. A larger community can benefit from these capabilities if Modelica-based components can be integrated into their favourite simulation tools. This work addresses the impact of transferring Modelica prototyping capabilities into different classes of simulation tools: general-purpose modeling tools, domain-specific tools and academical research-oriented simulation environments. In particular, it shows that the realization of model-based research of cyber-physical systems shall benefit from the convergence of such efforts using the functional mockup interface.**

***Index Terms*—functional mockup interface FMI, functional mockup unit FMU, cosimulation , Modelica, agent-based modeling, GridLAB-D, HLA, TRNSYS**

## I. Introduction

CYBER-PHYSICAL ENERGY SYSTEMS are facing rapid developments concerning energy resources, communication technologies, sensor devices and others. The variety of components out of which such complex systems are composed makes model-based research a challenging task. Components can be not only of physical types like e.g. power generation, transportation, networks, power consumption, cities and buildings but can be also of virtual nature like communication, statistically random behavior, controls with sensors, national politics, oil prices etc [1]. Many actual challenges can be viewed from two perspectives: specification and implementation perspectives.

From the model specification perspective, models need to be described using well-established specification covering many aspects like continuous-time, discrete events, statistics and artificial intelligence domains [2], [3]. From implementation perspective, these specifications need to get translated to simulation code adequately. Nevertheless, while a wide set of highly-advanced simulation tools exists each providing sophisticated functionalities covering a specific aspect of energy systems [4], [5], a tool that covers all possible mentioned aspects does not exist for practical reasons.

Consequently, investigating future cyber-physical systems of new technologies would certainly benefit from a comprehensive modeler-oriented methodology that is capable of performing the following tasks for the components of such complex systems that are modeled with various tools:

- rapid prototyping
- arrangement within well-defined hierarchies possibly according to hybrid paradigms
- distributed cosimulation in different (hardware) platforms
- exploiting high-performance computing resources

One step towards this goal is to enable the combination of many types of tools within a cosimulation platform. These kinds of tools, can be classified, among others, as follows:

1) General-purpose modeling languages
2) Domain-specific tools capable of modeling highly sophisticated aspects
3) Self-developed tools and methodologies from academia by which research-oriented questions are investigated

A natural approach is to exploit standardized cosimulation capabilities for enabling the usage of desired tools and hence combining the advantages of all underlying approaches. Additionally, flexible prototyping capabilities of new components would certainly enhance this approach. Here, we constructively suggest the Modelica language as a rapid prototyping platform for other simulation tools. In Elsheikh et al. [2], many features of the Modelica language were primarily examined in the context of complex energy systems. The main elements of such systems were abstracted in a model example. The prototyping capabilities of the Modelica language as well as were the advantages of employing Modelica were primarily emphasized in the context of complex energy systems.

In this work, the impact of transferring Modelica capabilities into other tools is addressed. This is discussed along three tools categorized according to the mentioned tools classification. The integration of Modelica into other tools is done via the functional mockup interface (FMI) [6]. In this aspect, FMI can be viewed as a medium for enabling Modelica technologies for other arbitrary simulation tools and hence extend the scope of their possible applications.

The rest of this work is structured as follows: Section II gives an overview of the Modelica language followed by Section III for introducing the FMI technology. Sections IV, V and VI demonstrate the advantages of integrating Modelica-based components into GridLAB-D, TRNSYS and HLA, respectively. Finally, Section VII provides a brief summary.

## II. Rapid prototyping with Modelica

### A. Background

Modelica [7] is one of the modern state of the art equation-based modeling and simulation languages. It is based on universal domain-independent modeling concepts adequate for multidisciplinary applications [8], [9]. Object-oriented facilities enabling encapsulation for component reuse and object inheritance for hierarchical modeling are fundamental characteristics of Modelica. The object oriented modeling philosophy relies on the fact that any system, no matter how complex it is, can be decomposed into a finite set of smaller components. Modelica can describe components corresponding to differential algebraic equations (DAEs) with intuitive language constructs for implicit equations.

The Modelica language was initiated 1997 as a specification language for exchanging models among different working groups. Many well-established and promising features of many existing modeling languages and developed concepts have been adopted. It was and is still subject to intensive discussions concerning its maintenance and development within the Modelica association (www.modelica.org). An international conference is organized every 18 months with ever continuously increasing number of participants from industry and academia.

### B. Modeling concepts

The key concept behind Modelica is that it employs non-causal modeling concepts by which input/output relationship among components is usually absent. That is, data flow among model components needs not to be explicitly defined. Component variables are not necessarily supposed to be declared as inputs or outputs of each others[1]. A component is typically encapsulated with well-defined interfaces to the external world. These interfaces, called connectors, work as communication ports enabling the connection to other components. The connectors can be viewed as energy carriers usually characterized by two types of variables, flow variables (say $E$) and potential variables (say $P$). By connecting two identical connectors together, two types of equations are generated, sum to zero equations for flow variables and identity equations for potential variables, see Figure 1 [2]. The former type of equations emulates conservation laws present in physical domains, that is the sum of all flows of conserved quantities (e.g. energy) into and out of a component must be equal to zero. In this way, the modeling of large-scale systems becomes the task of dragging, dropping component icons and connecting them together. Resulting models are visually one-to-one map to the conceptual reality.

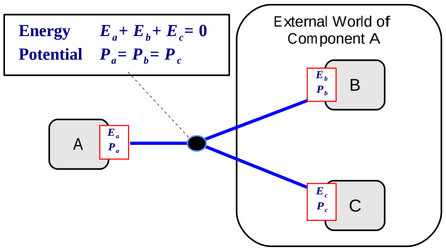

Energy $E_a + E_b + E_c = 0$
Potential $P_a = P_b = P_c$
External World of Component A
A
$E_a$ $P_a$
B
$E_b$ $P_b$
C
$E_c$ $P_c$

Fig. 1. Connecting components in Modelica assembles common conservation laws of Physics

### C. Modeling and simulation with Modelica

Components from different physical domains are implemented by identifying the right quantities for potential and flow variables. For example, typical quantities for flow variables are current, particles flow rate and heat flow rate for the electrical, hydraulic and thermal fields, respectively. The corresponding potential quantities are voltage, pressure and temperature, respectively. A significant advantage of Modelica is the presence of a large-set of free and commercial libraries in many physical domains. The Modelica association continuously integrates further libraries into an ever growing Modelica standard library (MSL). By developing sophisticated new applications, there is no need to start development from scratch. Many already implemented components and types can be reused.

Having constructed Modelica models, many Modelica-like simulation environments like Dymola [10], MapleSim (www.maplesoft.com), OpenModelica [11], Wolfram SystemModeler (www.wolfram.com) and others exist for modeling, compiling and simulating Modelica models. Such Modelica simulation environments are responsible for transforming graphical models into efficient simulation code using sophisticated symbolic algorithms for manipulating and simplifying resulting large-scale equation systems. Such generated equations are usually sparse, i.e. in each equation, only few variables are present. Consequently, modern DAE solvers exploiting the Jacobian sparsity are usually utilized for efficient and robust numerical integration. Moreover, additional capabilities for detecting and handling events are present. More features of Modelica are explicitly emphasized in [2].

## III. Transferring Modelica via FMI

### A. Background

FMI, a standardized unified model interface, is a result of the MODELISAR project (www.modelisar.com) aiming at improving the design and interoperability of simulation tools for embedded system applications. FMI is becoming the current trend for tools interoperability among simulation-based software. Currently, more than 40 simulation tools are supporting FMI (www.fmi-standard.org/tools).

A simulation tool supporting FMI is capable of exporting its models as functional mock-up units (FMU)s. A FMU is a zip file containing the following components:

1) The implementation of a model as a compiled C shared

[1] Nevertheless, some causal standard constructs similar to those present in classical procedural languages are also supported. Modelica specification is flexible enough that also the classical block-diagram approach (e.g. as in Simulink) can be emulated, if needed

libraries following a specific API in C, optionally accompanying the source code

2) A model-description XML file, containing among others a description of the model inputs, parameters, outputs, used types and physical units
3) Other optional data e.g. icon images, user interface specification and relevant documentation

Such FMUs can be simulated as stand-alone programs or imported by other FMI-supporting simulation tools.

### B. Supported FMI operations

A FMU implements a model mathematically described by a hybrid ordinary differential equations system (ODE) as a mixture of continuous state and discrete variables, see Figure 2. Additionally, a set of algebraic equations depending on state variables $x$ could be also present making the underlying model conceptually equivalent to a DAE of index one [12]. A typical cosimulation master importing FMUs as a slave black-box model would typically perform the following FMI operations on a FMU in the same order emphasized by Figure 2:

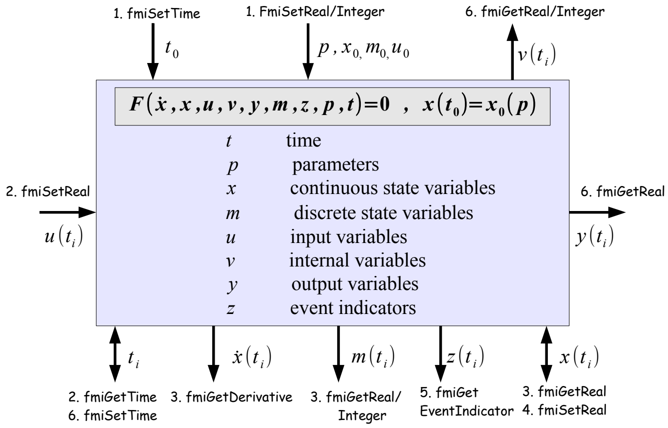


Fig. 2. Mathematical description of a FMU. Contents of FMUs can be retrieved and updated using naturally ordered elementary FMI calls

1) Initialization step: setting up start time $t_0$, model parameters $p$, start values $x(t_0)$ and input variables $u(t_0)$

Having an initialized FMU, the numerical integration is iteratively performed at discrete time steps $t_i, i = 1, 2, \ldots$ as follows:

2) Pre-processing step: set step $t_i$ and set the inputs $u(t_i)$
3) Integration step: update values of $\dot{x}(t_i), x(t_i)$ and $m(t_i)$ and perform numerical integration
4) Post-processing step: update state variables $x(t_i)$
5) Event handling step: report the presence of events if $z_j(t_i) * z_j(t_{i-1}) < 0$ and handle them adequately
6) Output steps: Compute the outputs $y(t_i)$ and other intermediate variables $v(t_i)$, if needed

and finally:

7) Finalization step: memory deallocation and processing of results

Step number 3 is done using either an external solver or an internal solver that comes with the FMU. The former case is performed if the FMU only supports FMI for model exchange (FMI-ME) while the latter is referred to as FMI for cosimulation (FMI-CO). In FMI-CO, additional FMI routines are provided for performing self numerical integration.

### C. Common advantages of FMI

There are many drivers for considering the FMI technology within common as well as own self-developed simulation tools. Firstly, simulation tools can import model components implemented by other simulation environments capable of exporting their models as FMUs. Secondly, self exported FMUs can be imported by other modeling, simulation and analysis tools capable of importing FMUs such as PySimulator [13] and JModelica [12]1. Many tools for assisting the implementation, validation and simulation of FMUs exist such as the FMI library (www.jmodelica.org/FMILibrary), FMI SDK development kit (www.qtronic.de/en/fmusdk.html) and FMU compilance checker (www.fmi-standard.org/downloads).

In summary, although the original motivation behind FMI is to support Modelica-based developments for embedded systems, nevertheless, FMI can be viewed as a medium for transferring the Modelica capabilities into other modeling and simulation tools. Overall, this definitely enhances the multidisciplinary collaboration potentials among several working groups and specialists in several domains. In particular, several tools based on long decades of experiences and developments are definitely not easily reproducible in favourite languages.

## IV. Modelica-based FMUs for Universal simulation languages: GridLAB-D

### A. Background

GridLAB-D (www.gridlabd.org) is one of the leading developer-oriented open-source modeling environment for simulating discrete event-based systems [14]. This specification language is typically used for describing several aspects of energy systems like weather, market, power systems, grids, distribution automation modules and many others for power system analysis applications. Already a large repositories of such modules exist and are developed, utilized and exchanged among different research groups.

GridLAB-D adopts an agent-based simulation approach. Each agent type is characterized by a set of property states and actions for updating states. The specification of an agent type is realized by either one of the following:

1) An intuitive GridLAB-D specification languages (or)
2) A C++ class according to templates

Using the GridLAB-D language, a *module* of interacting agents can be specified. Each instantiated agent dynamically calculates next time point for updating its states. The simulation engine of GridLAB-D is responsible for tracking the states of agents, managing the communication among agents by advanced scheduling algorithms and controlling the progress of time through a universal global clock. The modeler can construct and efficiently simulate large-scale complex systems with thousands of interacting agents by organizing the agents into hierarchies of subsystems and working subgroups.

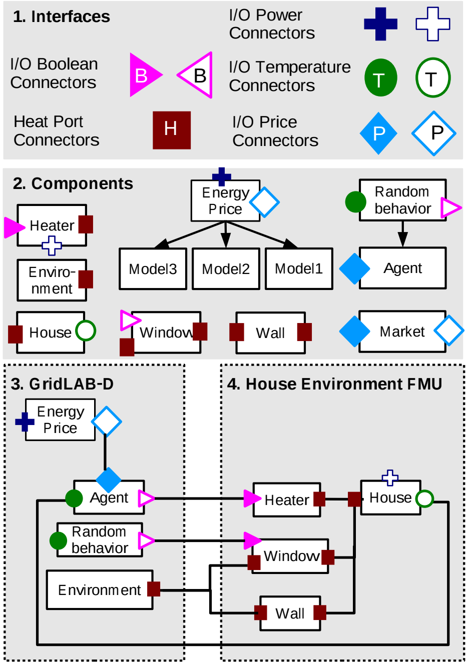


Fig. 3. Coupling GridLAB-D modules with a FMU component. The continuous-time part is modeled with Modelica and the discrete part is modeled with GridLAB-D

## B. Advantages

Many advantages for integrating FMU-based Modelica components within GridLAB-D are visually demonstrated in Figure 3. The physical part of the model describes the thermal behaviour of an energy consumption unit (a house with a heater, a window and an isolation wall), cf. [2] for implementation details in Modelica. The discrete part implemented by GridLAB-D describes the external conditions influencing the power consumption. This includes a weather module, statistically random behaviour agent (opening/closing windows), an intelligent agent controlling the power consumption of a house (e.g. heater settings), influenced by a market agent supplying the energy prices.

*1) Benefits for the GridLAB-D community:* The benefits of the presented coupling can be directly extracted from this simple model. Both Modelica and GridLAB-D together can be used for efficient prototyping and simulation of complex hybrid systems. The benefits to the GridLAB-D community can be viewed as follows: when modeling cyber physical systems, the continuous part can be rapidly prototyped with Modelica. Moreover, despite of the fact that GridLAB-D is a simulation language, it does not inherit any capabilities for numerical integration of ODEs. On the other side, FMUs can be numerically integrated in a straightforward way, even if FMI-CO is not supported. FMI-ME design is adequately relevant for interfacing with common ODE solvers.

*2) Benefits for the Modelica community:* On the side, GridLAB-D is also a useful tool for FMU-based modeling. GridLAB-D can be considered as an easy-to-use user interface for setting and instantiating FMU-based components. Existing powerful statistical tools can be used for initializing FMUs using e.g. uniform, normal and exponential distribution among many others. In this way, statistical studies based on parameter variations can be performed on the fly. Another significant aspect is the powerful capabilities of GridLAB-D for high performance simulation of very large-scale systems that cannot be simulated with Modelica alone due to their complexity [3]. Easy-coupled large-scale systems can be decomposed into thousands of smaller interacting FMUs on a discrete simulation basis. The interactions among FMUs, their execution order and the exchange of data can be specified with GridLAB-D.

In summary, adjoining the non-causal modeling approach with the discrete agent-based approach results in a powerful modeling environment with a larger scope of applications, as done in Stifter and et al. [15].

# V. Modelica-based FMUs for domain specific tools: TRNSYS

## A. Background

TRNSYS [16] is a domain-specific tool for simulating the thermal behaviour within energy-efficient buildings among many other domain-related applications. It provides high-level facilities for constructing building models with comprehensive details including multi-zoning description (e.g. internal structure and rooms), exact architectural specifications and geothermal conditions (e.g. typical weather conditions, sun light directions and duration). TRNSYS additionally provides a wide extensive set of model components for controllers, electrical storages, HVAC, solar thermal collectors and many others. A sophisticated highly-detailed GUI is available for the modeling task.

TRNSYS is based on a block-diagram approach for modeling technical systems. Each component computes output variables from given input variables and parameters. Systems are assembled by connecting components together by which the inputs of some components are the outputs of others. For components implementing ODEs, the numerical integration can be performed with the TRNSYS simulation engine, the *kernel*. The kernel additionally employs a block-wise successive iteration method suitable for handling potential loops in the specified model.

## B. Developing new components

The underlying modular architecture of TRNSYS separates between a model component specification interface, its implementation and the numerical integration process. The components are implemented as dynamically loaded libraries (DLLs) to be loaded by the kernel. This gives the opportunity for supplying user-implemented components not provided by the TRNSYS standard library. This is done by implementing standard template routine in C++ or Fortran. A typical routine implements a specific set of operations like:

- setting initial values and model parameters
- computing model outputs and state derivatives at given step-sizes
- post-processing and a finalization step

These operations are then queried by the kernel for performing the numerical integration[2]. While low-level implementation of small-size models could be a straightforward task, realizing large-scale physical models within a procedural language is a tedious task. Firstly, the developer needs to translate the model into a set of equations. This is not a straightforward task for complex large-scale models.

### C. Advantages

Alternatively, the physical modeling can be rather done by Modelica and integrated into TRNSYS via FMI, see Wetter [17] for a comparison between Modelica and TRNSYS w.r.t. model development time. The required operations for realizing user-defined types in TRNSYS are ideally compatible with typical FMI operations. That is, it is possible to build FMU-based types for TRNSYS as done in Elsheikh et al. [18]. In this work, a single wrapper FMI-based code file is provided for integrating TRNSYS-conform[3] FMUs into TRNSYS. This has a lot of advantages briefly demonstrated in Figure 4. While TRNSYS already contains a comprehensive subset of model components, model-based investigation for innovative research studies of new concepts and emerging technologies can most benefit from the prototyping capabilities of Modelica. Advanced complex Modelica models can be integrated into basic TRNSYS types without low-level explicit implementation of the mathematical details. Specialized features exclusively present in some Modelica building libraries [19] can be utilized. Meanwhile, many sophisticated extensively tested model components in TRNSYS, that are not easily constructable with Modelica, can be exploited. That is, the horizon of applications scope by combining both sides certainly gets extended.

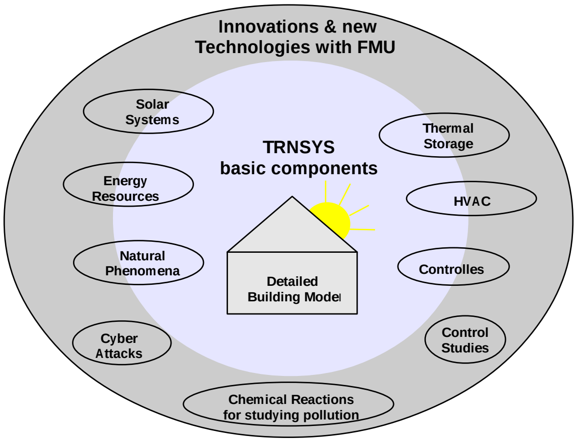


Fig. 4. Rapid prototyping of innovations and new technologies with TRNSYS and FMI. The internal region corresponds to applications that can be implemented with TRNSYS . The external regions corresponds to model components that are rapidly prototyped with Modelica.

## VI. Modelica-based FMUs for the HLA cosimulation environment

### A. Background

The high level architecture (HLA) is a simulation interoperability cosimulation standard [20], designed to support distributed simulations with various different synchronization schemes. It has been used in large-scale applications of industry and defence simulations for more than a decade. Dynamic management, incremental design and development support are fundamental features of the HLA. In the HLA terminology a simulation component is called a ``Federate˝, while the whole distributed simulation is called a ``Federation˝. The most important functional aspect of the HLA is its runtime infrastructure (the RTI). The RTI works as a central communication server for all the individual simulations. The RTI is responsible for synchronizing the federates, providing them timely updates of the federation shared data structures. It also allows the federates to manage the creation, deletion and ownership of data structures. Hence, the HLA can be viewed as a platform supporting parallel discrete event simulations (PDES).

### B. Advantages

By combining FMI into the HLA framework, a PDES platform enhanced with continuous-time simulations is established. The resulting framework is ideal for performing efficient simulations of massively large-scale applications and intensively expensive computations, e.g. energy consumption of a city with various energy resources. Such large simulations would also need flexible-demand of computational power to be assimilated. The idea of extending HLA with continuous-time simulation has been presented in Müller et al. [21] where discrete event network simulators were combined with electricity domain continuous simulators. Awais et al. [22] has presented a preliminary design of using the HLA with the FMI.

Figure 5 presents a corresponding hypothetical scenario of hardware configuration along with the RTI, which may enable a large simulation like this. Figure 5 emphasizes that

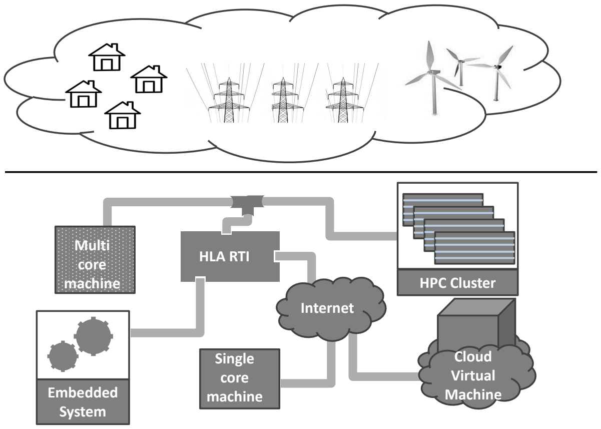


Fig. 5. A distributed simulation for analyzing the energy requirements of a big city

[2] The numerical integration can be also performed internally within the component

[3] Complete declaration for the public items are specified, the inputs and the outputs of the FMU and a parameterized start values for state variables, if required. Any Modelica model can be rewritten in a way to make the resulting FMU to be TRNSYS-conform

an HLA based simulation is completely network independent. Several machines within super-computer clusters are dedicated to simulate the energy producers, like wind mills, hydro electricity and others. Communication of data is simulated by discrete event based network simulators on a parallel computing machine. Actual devices as embedded systems are assisting the simulation of transfer of power and load regulation. The houses as energy consumption units are simulated on cloud virtual machines. In this topology, cloud-based computing enables dynamical computational power resources based on time-varying demands.

To summarize; combining FMU components within the HLA gives many advantages over simulations running as single processes. An efficient exploitation of parallel distributed simulation environment is more scalable, robust and flexible. Currently HLA is one of the reliable standardized interface for parallel and distributed simulation interoperability. Stipulating its flexibility with the power of FMU-based modeling enhances reusability and brings great benefits for model-based investigation of large-scale systems.

## VII. Outlook

This work addresses the impact of transferring Modelica rapid prototyping capabilities into other simulation tools via the FMI technology. For that purpose, three different tools has been chosen as representatives of different classes of modeling and simulation environments. GridLAB-D, a universal modeling language, certainly benefits from enhanced prototyping capabilities from FMU-based Modelica components. Additionally, more descriptive power enhances the underlying agent-based discrete modeling approach. Numerical integration capabilities of FMUs are provided by defaults. Large-scale applications can be efficiently simulated by decomposing complex systems at weak-coupled points into hierarchical subsystems of interacting agents communicating on discrete-time basis. The scope of applications is extended by exploiting the best of both languages.

Enabling Modelica rapid prototyping to TRNSYS, a domain-specific tool, gives a promising approach for implementing new innovative technologies. FMU-based prototyped components can be independently tested and analyzed by a large set of FMU-based tools. Capabilities of domain-specific Modelica libraries can be integrated. These statements are also applicable to many domain-specific tools. HLA, a co-simulation environment for parallel discrete event simulators, provides a perfect ground for establishing high performance computing environment for massively large-scale applications of cyber physical systems. Continuous-time simulations can be integrated by enabling Modelica-based technologies within the HLA platform via FMI.

## References

[1] M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu, "Seamless model-based development: From isolated tools to integrated model engineering environments," in *Proceedings of the IEEE*, vol. 98, no. 4, 2010, pp. 526 – 545.

[2] A. Elsheikh, E. Widl, and P. Palensky, "Simulating complex energy systems with modelica: A primary evaluation," in *DEST'2012, the 6th IEEE International Conference on Digital Ecosystems and Technologies*, Campione d'Italia, Italy, 2012.

[3] E. Widl, P. Palensky, and A. Elsheikh, "Evaluation of two approaches for simulating cyber-physical energy systems," in *IECON'2012, the 38th Annual Conference of the IEEE Industrial Electronics Society*, Montreal, Canada, October 2012.

[4] C. Macana, N. Quijano, and E. Mojica-Nava, "A survey on cyber physical energy systems and their applications on smart grids," in *Innovative Smart Grid Technologies (ISGT Latin America), 2011 IEEE PES Conference on*, October 2011.

[5] M. Ilic, L. Xie, U. Khan, and J. Moura, "Modeling future cyber-physical energy systems," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008.

[6] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf, "The functional mockup interface for tool independent exchange of simulation models," in *Modelica'2011: The 8th International Modelica Conference*, Dresden, Germany, 2011.

[7] H. Elmqvist and S. E. Mattsson, "Modelica - the next generation modeling language: An international design effort," in *ESS97: The 9th European Simulation Symposium*, Passau, Germany, 1997.

[8] F. Casella and P. Colonna, "Dynamic modeling of IGCC power plants," *Applied Thermal Engineering*, vol. 35, pp. 91 – 111, 2012.

[9] F. Casella, F. Donida, and M. Lovera, "Beyond simulation: Computer-aided control system design using equation-based object-oriented modelling for the next decade," *Simulation News Europe*, vol. 19, no. 1, pp. 29 – 41, 2009.

[10] D. Brück, H. Elmqvist, H. Olsson, and S. E. Mattsson, "Dymola for multi-engineering modeling and simulation," in *Proceedings of the 2nd International Modelica Conference*, Munich, Germany, 2002.

[11] P. Fritzson, P. Aronsson, H. Lundvall, K. Nystrm, A. Pop, L. Saldamli, and D. Broman, "The OpenModelica modeling, simulation, and software development environment," *Simulation News Europe*, vol. 44, no. 45, Dec. 2005.

[12] C. Andersson, J. Åkesson, C. Führera, and M. Gäfvert, "Import and export of functional mock-up units in JModelica.org,," in *Modelica'2011: The 8th International Modelica Conference*, Dresden, Germany, 2011.

[13] A. Pfeiffer, M. Hellerer, S. Hartweg, M. Otter, and M. Reiner, "PySimulator – A simulation and analysis environment in Python with plugin infrastructure," in *Modelica'2012, the 9th International Modelica Conference*, Munich, Germany, September 2012.

[14] D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *Transmission and Distribution Conference and Exposition, 2008. IEEE/PES*, april 2008, pp. 1 –5.

[15] M. Stifter, E. Widl, F. Andrén, A. Elsheikh, T. Strasser, and P. Palensky, "Co-simulation of components, controls and power systems based on open source software," in *2013 IEEE Power & Energy Society General Meeting*, Vancouver, Canada, July 2013, accepted.

[16] S. A. Klein, J. A. Duffie, and W. A. Beckman, "TRNSYS: A transient simulation program," *ASHRAE Transactions*, vol. 82, pp. 623 – 633, 1976.

[17] M. Wetter and C. Haugstetter, "Modelica versus trnsys – a comparison between an equation-based and a procedural modeling language for building energy simulation," in *The 2nd SimBuild Conference*, Cambridge, MA, USA, August 2006.

[18] A. Elsheikh, E. Widl, P. Pensky, F. Dubisch, M. Brychta, D. Basciotti, and W. Müller, "Modelica-enabled rapid prototyping via TRNSYS," in *BS'2013, The 13th International Conference of the International Building Performance Simulation Association*, Chambéry, France, August 2013, submitted.

[19] M. Wetter, "Modelica-based modelling and simulation to support research and development in building energy and control systems," *Journal of Building Performance Simulation*, vol. 2, pp. 143 – 161, 2009.

[20] Simulation Interoperability Standards Committee *et al.*, "IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-IEEE std 1516-2000, 1516.1-2000, 1516.2-2000. new york: Institute of electrical and electronics engineers," *Inc., New York*, 2000.

[21] S. C. Müller, H. Georg, C. Rehtanz, and C. Wietfeld, "Hybrid simulation of power systems and ICT for real-time applications," in *3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, Berlin, Germany, 2012.

[22] M. Awais, P. Palensky, A. Elsheikh, E. Widl, and M. Stifter, "The high level architecture RTI as a master to the functional mock-up interface components," in *ICNC 2013 International Workshop on Cyber-Physical System (CPS) and its Computing and Networking Design*, San Diego, USA, January 2013.

# The FMI++ Library: A High-level Utility Package for FMI for Model Exchange

Edmund Widl, *Member, IEEE*, Wolfgang Müller, *Student Member, IEEE*, Atiyah Elsheikh, *Member, IEEE*,
Matthias Hörtenhuber and Peter Palensky, *Senior Member, IEEE*

Austrian Institute of Technology, Energy Department, Vienna, Austria

{givenname.surname}@ait.ac.at

***Abstract*—The success and the advantages of model-based design approaches for complex cyber-physical systems have led to the development of the FMI (Functional Mock-Up Interface), an open interface specification that allows to share dynamic system models between different simulation environments.**

**The FMI specification intentionally provides only the most essential and fundamental functionalities in the form of a C interface. On the one hand, this increases flexibility in use and portability to virtually any platform (even embedded control systems). On the other hand, such a low-level approach implies several prerequisites a simulation tool has to fulfil in order to be able to utilize such an FMI component, for instance the availability of adequate numerical integrators. The FMI++ library presented here addresses this problem for models according to the FMI for Model Exchange by providing high-level functionalities, especially suitable for but not limited to discrete event simulation tools. The capabilities of this approach are illustrated with the help of several applications, where the FMI++ library has been successfully deployed.**

**This approach intends to bridge the gap between the basic FMI specifications and the typical requirements of simulation tools that do not primarily focus on continuous time-based simulation. In other words, this enables such models to be used as de-facto stand-alone co-simulation components.**

## I. Introduction

The FMI (Functional Mock-Up Interface) for Model Exchange [1] defines a standard that allows to exchange dynamic models between different simulation tools. This specification consists basically of two parts:

- Model interface: Each model has to provide an executable (shared library) with strictly defined functionalities, implemented in C.
- Model description scheme: Along with the executable an XML-file has to be provided, that contains all necessary information about the model.

When implemented for a model, these two elements together, wrapped up in a ZIP archive, comprise a Functional Mock-Up Unit (FMU).

Even though the actual specification includes intentionally only very fundamental functionalities, it provides all the essential features to represent e.g. Modelica [2], Simulink[1] and SIMPACK[2] models. These features include for instance access to the model parameters, its actual states and derivatives as well as event indicators. The advantages of such a low-level approach are tool independence, i.e. the model interface includes no simulator-specific functionalities, and platform independence, since C-compilers are available for virtually any operating system and processor.

Due to these advantages, the FMI for Model Exchange specification becomes increasingly popular. Several well established simulation tools already offer the possibility to import and simulate FMUs, either natively (e.g. OpenModelica [3], JModelica [4], Dymola[3] or SimulationX[4]) or via third party tools (e.g. Modelon's FMI Toolbox[5] for MATLAB/Simulink).

However, the FMI specification is primarily intended for models that comprise (sets of) hybrid ordinary differential equations (ODE). Therefore, virtually all tools that currently support the import or export of models by means of FMI for Model Exchange are simulation environments that focus on continuous time-based simulation. Hence all of these tools provide their own high-level functionalities (e.g. numerical integrators) which are necessary to handle such FMUs.

For applications that lack these features the FMI for Co-Simulation specification has been developed. As the name implies, this specification requires the models to provide their own internal utilities for simulation. However, most modelling tools only offer the possibility to export models according to the FMI for Model Exchange definition, while fewer support FMI for Co-Simulation. For this reason, the actual usage of FMUs is still mostly limited to simulation environments that are well capable of dealing with continuous time-based models anyway.

The goal of the FMI++ library is to bridge this gap, by providing model-independent functionalities for the simulation of Model Exchange FMUs, including numeric integration and event handling. This offers the possibility to include FMUs with relatively small effort into simulation environments, that by itself do not support the simulation of hybrid ODE-based models. Developers using FMI++ do not need to focus on low-level details of common FMI functionalities, but can concentrate on the task at hand.

This paper is organized as follows: Section II gives an overview of the related work regarding software for FMI

[1] http://www.mathworks.com, MathWorks

[2] http://www.simpack.com, SIMPACK Multi-Body Simulation

[3] http://www.dymola.com, Dassault Systèmes

[4] http://www.itisim.com/simulationx, SimualtionX

[5] http://www.modelon.com, Modelon FMI Toolbox for MATLAB

support. In Section III the functionalities of the FMI++ library are covered in detail. Section IV gives examples of the successful deployment of the FMI++ library, demonstrating its practicality and flexibility. Finally, Section V provides the conclusions and an outlook.

## II. Related Work

Convenient high-level approaches like object-oriented interfaces are intentionally left out from the FMI specifications, in order to achieve the possibly highest degree of platform independence. Also the questions of how to unzip an FMU or retrieve the model information from the XML-file lie within the user's responsibility.

There are however free open-source development tools available that implement generic methods for interacting with FMUs: QTronic's FMU Software Development Kit[6] (FMU SDK) and JModelica's FMI Library[7] (FMIL). Both are intended to serve as a starting point for applications that export or import FMUs. They are written in C and offer support for unzipping, (meta) information retrieval, dynamic model loading, setting of model parameters and evaluation of model equations.

The FMIL package has been used as basis for the development of the FMU Compliance Checker[8], a free software package that checks any given FMU's compliance with the FMU standard. Also the PyFMI library[9], a Python package offering an interface for interaction with FMUs, relies on the FMIL.

Also worth mentioning is JFMI[10], a Java wrapper for FMI. Even though it does not provide functionality beyond the scope of the FMI specification, it extends the scope of FMI from C/C++ applications to Java, thus effectively adding another level of platform independence.

However, even though these packages offer lots of convenient functionalities for the practical handling of FMUs, their main features are still fairly basic. For simulation environments not focused on the numerical integration of continuous time-based components, i.e. especially discrete event-based simulators, these packages do not offer the tools needed to easily integrate FMUs. This is where the FMI++ library tries to step in, by offering generic but advanced numerical integration and event handling capabilities for FMUs.

## III. Implementation

Since the FMI++ library intends to promote the FMI specification and its use in modern simulation environments, it is a freely available open-source project. This will hopefully also encourage other developers to contribute to the code base for future improvements.

The FMI++ library is written in C++, offering an object-oriented solution to interacting with FMUs. It is easily portable and has so far been tested on Linux architectures and on Windows (using MinGW/GCC). Fig. 1 gives an overview of the library.

### A. Dependencies

FMI++ relies on a few well-established open-source software packages, which already provide validated concepts and solutions.

*XML parsing and information retrieval:* For the retrieval of model (meta) information from the description file, the FMI++ library uses the corresponding FMU SDK functionalities. The FMU SDK itself depends on eXpat[11] for the parsing of the XML model description file.

*Numerical integration:* For numerical integration FMI++ relies on the ODEINT library [5], which has recently become an official part of the Boost library collection [6]. It is a highly flexible and top performing C++ library for numerically solving ordinary differential equations. Since ODEINT is a header-only template library, it imposes no further dependencies at runtime on FMI++.

*FMIL extension:* The FMI++ library can be used on top of the FMIL, in which case the FMIL has to be already properly installed.

### B. The self-integrating FMU class

The most obvious obstacle for using a bare FMU for Model Exchange is its lack of an integrator. For this reason, the FMI++ library provides a generic method for integration, encapsulated into a dedicated object. The resulting object owns the actual FMU instance and is able to advance its current state up to a specified point in time. It also provides utilities for convenient input and output handling and includes the proper handling of FMU-internal events.

*Class FMUBase:* This is the base for objects implementing self-integrating FMUs. It is implemented as a pure virtual interface and contains prototypes of all the functions needed by the numerical integrator and for advanced event handling (see Section III-C).

The most important features are:

- initialize/instantiate: These functions are responsible for the instantiation and initialization of the FMU and all corresponding necessary internal actions.
- integrate: Advances the state of the FMU to the specified point in time, with either a specified number of integration steps or a fixed integration step size.
- raiseEvent/handleEvents: These functionalities are the prerequisite for proper event handling. Whenever an event occurs, be it either a change of external inputs or an update of the internal state, the internal FMU instance has to be notified (via raiseEvent) and then the necessary actions have to be taken (by calling handleEvents).
- rewindTime: Event handling may in some cases involve the necessity to reset the internal FMU to a previous state. With this function the FMU-internal clock can be set back. This affects only the value of the internal time,

[6] `http://www.qtronic.de`, QTronic FMU Software Development Kit

[7] `http://www.jmodelica.org/FMILibrary`, FMI Library

[8] `https://www.fmi-standard.org`, FMU Compliance Checker

[9] `https://www.jmodelica.org`, PyFMI

[10] `http://ptolemy.eecs.berkeley.edu/java/jfmi`, JFMI

[11] `http://expat.sourceforge.net/`, The eXpat XML Parser

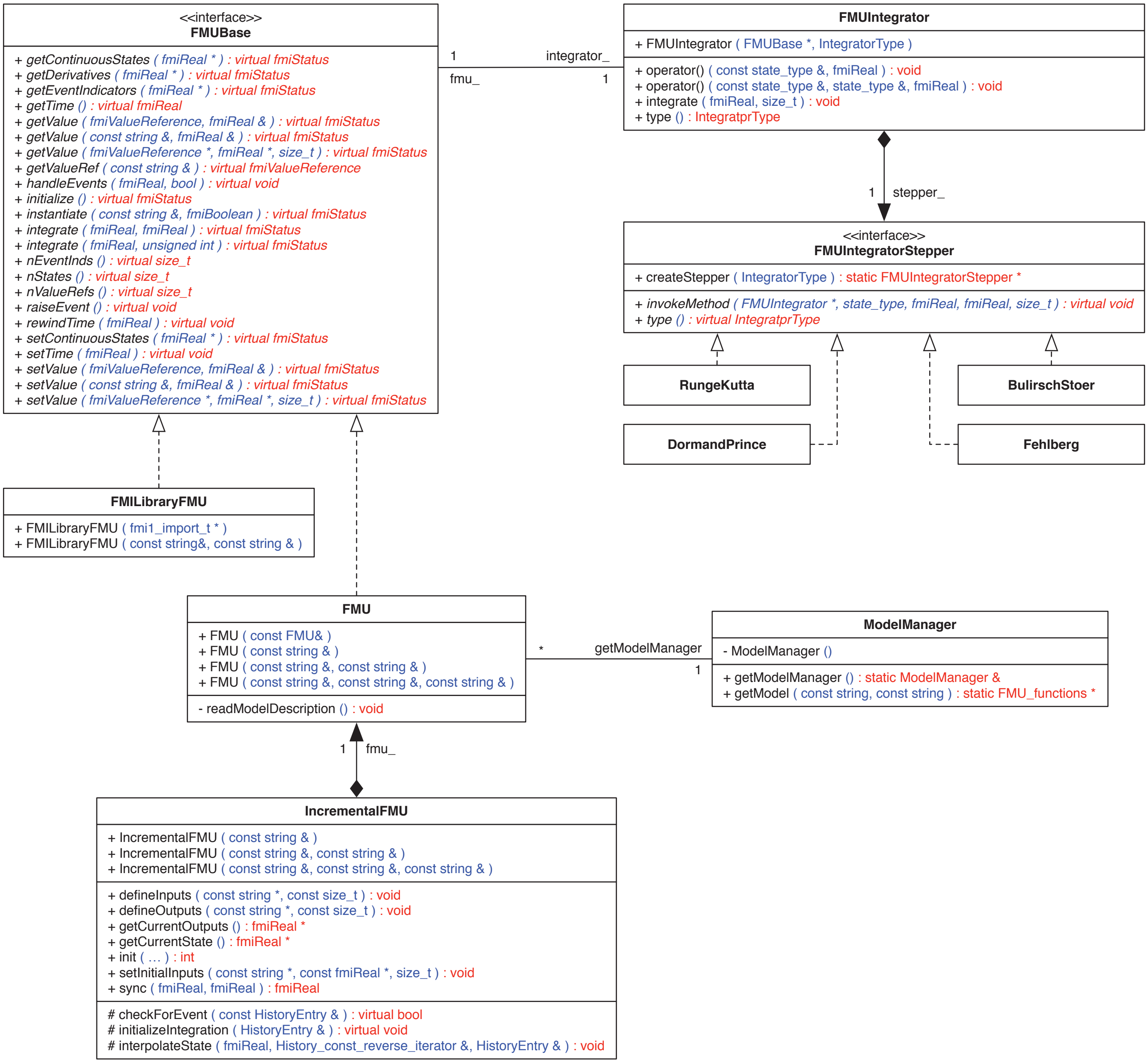

<<interface>>
FMUBase
+ getContinuousStates ( fmiReal * ) : virtual fmiStatus
+ getDerivatives ( fmiReal * ) : virtual fmiStatus
+ getEventIndicators ( fmiReal * ) : virtual fmiStatus
+ getTime () : virtual fmiReal
+ getValue ( fmiValueReference, fmiReal & ) : virtual fmiStatus
+ getValue ( const string &, fmiReal & ) : virtual fmiStatus
+ getValue ( fmiValueReference *, fmiReal *, size_t ) : virtual fmiStatus
+ getValueRef ( const string & ) : virtual fmiValueReference
+ handleEvents ( fmiReal, bool ) : virtual void
+ initialize () : virtual fmiStatus
+ instantiate ( const string &, fmiBoolean ) : virtual fmiStatus
+ integrate ( fmiReal, fmiReal ) : virtual fmiStatus
+ integrate ( fmiReal, unsigned int ) : virtual fmiStatus
+ nEventInds () : virtual size_t
+ nStates () : virtual size_t
+ nValueRefs () : virtual size_t
+ raiseEvent () : virtual void
+ rewindTime ( fmiReal ) : virtual void
+ setContinuousStates ( fmiReal * ) : virtual fmiStatus
+ setTime ( fmiReal ) : virtual void
+ setValue ( fmiValueReference, fmiReal & ) : virtual fmiStatus
+ setValue ( const string &, fmiReal & ) : virtual fmiStatus
+ setValue ( fmiValueReference *, fmiReal *, size_t ) : virtual fmiStatus
1
integrator_
fmu_
1
FMUIntegrator
+ FMUIntegrator ( FMUBase *, IntegratorType )
+ operator() ( const state_type &, fmiReal ) : void
+ operator() ( const state_type &, state_type &, fmiReal ) : void
+ integrate ( fmiReal, size_t ) : void
+ type () : IntegratprType
1
stepper_
<<interface>>
FMUIntegratorStepper
+ createStepper ( IntegratorType ) : static FMUIntegratorStepper *
+ invokeMethod ( FMUIntegrator *, state_type, fmiReal, fmiReal, size_t ) : virtual void
+ type () : virtual IntegratprType
RungeKutta
BulirschStoer
DormandPrince
Fehlberg
FMILibraryFMU
+ FMILibraryFMU ( fmi1_import_t * )
+ FMILibraryFMU ( const string&, const string & )
FMU
+ FMU ( const FMU& )
+ FMU ( const string & )
+ FMU ( const string &, const string & )
+ FMU ( const string &, const string &, const string & )
- readModelDescription () : void
*
getModelManager
1
ModelManager
- ModelManager ()
+ getModelManager () : static ModelManager &
+ getModel ( const string, const string ) : static FMU_functions *
1
fmu_
IncrementalFMU
+ IncrementalFMU ( const string & )
+ IncrementalFMU ( const string &, const string & )
+ IncrementalFMU ( const string &, const string &, const string & )
+ defineInputs ( const string *, const size_t ) : void
+ defineOutputs ( const string *, const size_t ) : void
+ getCurrentOutputs () : fmiReal *
+ getCurrentState () : fmiReal *
+ init ( … ) : int
+ setInitialInputs ( const string *, const fmiReal *, size_t ) : void
+ sync ( fmiReal, fmiReal ) : fmiReal
# checkForEvent ( const HistoryEntry & ) : virtual bool
# initializeIntegration ( HistoryEntry & ) : virtual void
# interpolateState ( fmiReal, History_const_reverse_iterator &, HistoryEntry & ) : void

Fig. 1. Overview of the most important features of the FMI++ library.

but not the internal state of the FMU (which has to be changed via setContinuousState, etc.).

- getValue/setValue: Several convenient getter and setter functions are defined, allowing e.g. to set or get values by referring to their names.

*Class FMUIntegrator:* This class provides the link between ODEINT's numerical integration routines and all classes inherited from FMUBase. It is implemented as a functor object, that provides the necessary inputs (i.e. the FMU's continuous states and the according derivatives) to ODEINT. It also updates the internal state of the FMU with the corresponding result.

*Class FMUIntegratorStepper:* The actual integration algorithms provided by ODEINT are encapsulated in objects inheriting from this class. Currently implemented are a basic 4th-order method with constant step size (class RungeKutta), a 5th-order method with adaptive step size (class DormandPrince), an 8th-order method with adaptive step size (class Fehlberg) and a high-precision method with controllable order and adaptive step size (class BulirschStoer).

*Class FMU:* With this class, the FMI++ library provides a completely implemented realization of the interface defined by FMUBase. At construction time it dynamically loads the model, thus avoiding the need to define the model identifier as macro at compile time. It also has access to the complete set of information from the model description.

*Class ModelManager:* This is a utility class designed for the use by instances of class FMU. It is implemented as a singleton and handles the dynamic loading of FMU models

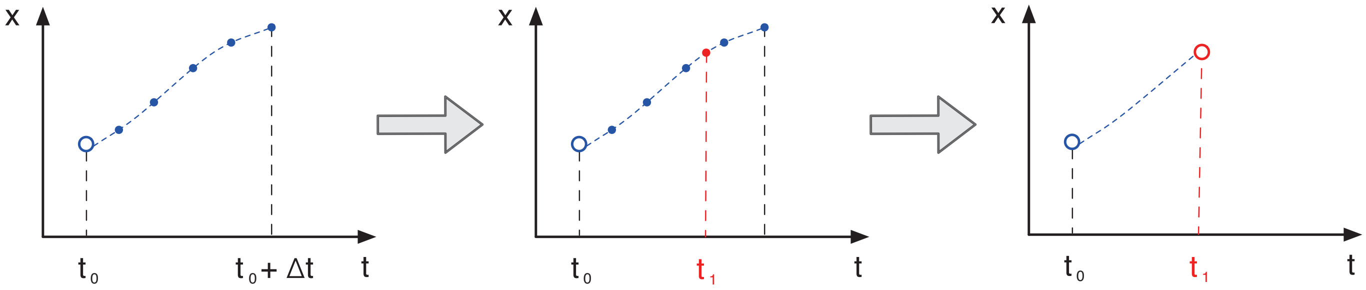


Fig. 2. Schematic view of an incremental update.

and the parsing of the corresponding model description. The singleton instance actually loads and parses each model only once during runtime and stores this information in case another instance of the same model is instantiated. This saves time in cases where simulations include many instances of the same model.

*Class FMILibraryFMU:* The abstract definition provided by class FMUBase makes it possible to combine other FMI tools with the FMI++ library. The class FMILibraryFMU gives an example for such a hybrid use, by utilizing the FMIL implementation for the internal handling of FMUs. This allows for example to integrate the functionalities provided by FMI++ into PyFMI (see Section IV-B).

### *C. The advanced event-handling FMU class*

The FMI++ library offers the possibility to combine the basic ability to integrate the state of an FMU with advanced event handling capabilities. This is especially useful when using FMUs within discrete event-based simulation environments, where the time difference between updates is not constant.

The class IncrementalFMU implements a lookahead mechanism, where predictions of the FMU's state are incrementally computed and stored. In case an event occurs, these predictions are then used to interpolate and update the state of the FMU. If no event occurs, the latest prediction can be directly used to update the FMU's state.

Fig. 2 shows a schematic view of such an incremental update. Shown on the left, at time $t_0$ the FMU's state $x$ is represented by a blue circle. According to this state, several predictions (blue dots) up to the time $t + \Delta t$ are computed and stored, with $\Delta t$ referred to as *lookahead horizon*. In the current implementation the time steps between these internal predictions are constant and have to be specified at instantiation time. Next, depicted in the middle, an (external) event occurs at time $t_1$. Since the exact time of the event does in general not coincide with one of the predictions, the state at that time is interpolated using the available predictions, depicted by the red dot. Finally, shown on the right, this interpolated prediction is used to update the actual state of the FMU, depicted by a red circle, and the old predictions are deleted.

It is important to note that the actual state of the FMU is not changed when the predictions are calculated. This is only done during the next update.

The most important features of class IncrementalFMU are:

- sync: This function call updates the associated FMU from time $t_0$ to time $t_1$. It first uses the previous predictions to update the state of the FMU. Subsequently it calculates a new set of predictions according to the current inputs.
- checkForEvent: This function checks for each new prediction whether an FMU-internal event has occurred. In case it returns true, no further prediction is computed. It is implemented as a virtual function, which enables the user to customize its behaviour.
- handleEvent: This function is called in case checkForEvent has returned true. It is implemented as a virtual function, which enables the user to customize its behaviour.
- initializeIntegration: This function initializes the integration by defining the first prediction. By default, this is the current state of the FMU. It is implemented as a virtual function, which enables the user to customize its behaviour.

The default implementation of class IncrementalFMU recognizes FMU-internal events and stops the prediction at the corresponding time. This implementation uses a linear interpolation technique to estimate the state from the stored predictions. By inheriting from class IncrementalFMU and customizing checkForEvent, handleEvent and initializeIntegration, it is possible to extend this functionality. Section IV-A gives an example where this feature is used to implement an additional controller.

## IV. Examples

The functionality of the FMI++ library has been tested by applying it within various simulation environments. The following examples give a brief glimpse of the accomplished results.

### *A. Inclusion into GridLAB-D*

GridLAB-D [7] is a discrete event-based micro-simulation tool with a focus on power distribution systems. It comes with a variety of plug-in modules for modeling and simulating energy generation, distribution and consumption as well as related topics such as controls, network communication or markets. By including the functionalities offered through FMI++ it is possible to include continuous time-based simulation components via FMI.

For the purpose of illustration a simple thermal system has been simulated. It consists of a simplified building model,

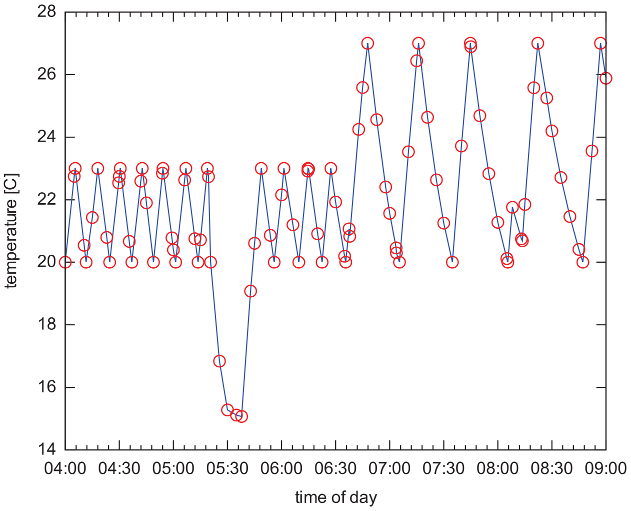


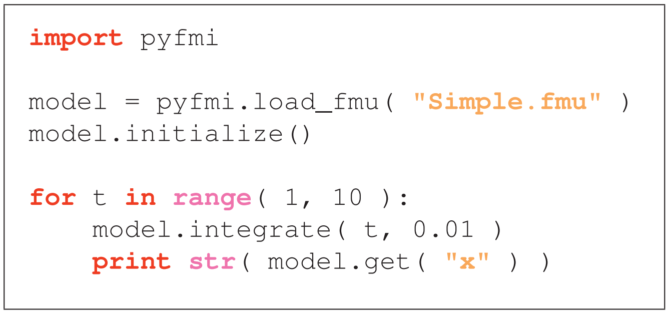
Fig. 3. Temperature profile of a building simulated in GridLAB-D using an FMU model via FMI++. Each red circle corresponds to an update (sync) of the FMU.

whose heater is turned on/off by a two point controller. The controller's set point can be altered by an agent, and at random times the building's windows are opened and closed, effectively changing the thermodynamic properties of the building. A detailed description of the model can be found in Reference [8].

Fig. 3 shows the results of such a simulation for an individual building. The thermal volume was modeled in Modelica, and exported as an FMU with the help of OpenModelica. The FMU was wrapped by a dedicated class derived from IncrementalFMU in order to be usable with GridLAB-D. The two point controller was not part of the FMU, but has been implemented by customizing the functionalities of checkForEvent, handleEvent and initializeIntegration (see Section III-C). Furthermore, the effects of external events have been included this way, i.e. the opening/closing of a window or changes due to the decisions made by the controller's agent.

### B. Extension of PyFMI

Using class FMILibraryFMU (see Section III-B) it is possible to include the features offered by FMI++ in PyFMI. Adding the possibility to integrate an FMU only needs 5 additional lines of code in PyFMI's source code (plus several changes in its setup to compile properly). Fig. 4 shows a simple Python script that uses the integrator feature of this modified version of PyFMI.



```
import pyfmi

model = pyfmi.load_fmu( "Simple.fmu" )
model.initialize()

for t in range( 1, 10 ):
    model.integrate( t, 0.01 )
    print str( model.get( "x" ) )
```

Fig. 4. Simple example script for PyFMI, testing FMI++'s integrating feature.

### C. Interaction with Ptolemy II's DE domain

Ptolemy II [9] is a generic open source simulation framework for studying the interplay of concurrent processes. These concurrent processes are represented by so-called actors, whose implementations have to obey certain guidelines (referred to as *abstract semantics*) to ensure a well-defined behaviour. The principles behind the design of the FMI++ library allow to define a dedicated actor for Ptolemy II's discrete event (DE) domain, that respects these abstract semantics.

Fig. 5 shows a basic discrete event-based Ptolemy II model containing such an FMU actor (labelled events FMU). In the upper left corner a green box visualizes the DE director that governs the execution of the model. The FMU actor is connected to a plotter, that observes and records the FMU's output. In this particular case the FMU contained within the FMU actor has been generated with the help of OpenModelica. The model simply integrates a constant, thus producing a linear output. However, the model raises internal events, that periodically reverses the constant's sign, which effectively changes the slope of the output signal. Fig. 6 shows the resulting output, where each of the red dots correspond to the firing of the actor. As can be seen, the actor is either fired once the integration's lookahead horizon is reached or when an internal event occurs.

The FMU actor uses internally an instance of class IncrementalFMU, so that the (potentially) costly numerical integration is completely executed in C++. The bindings between Java and FMI++ have been done using SWIG [10]. See Reference [11] for details on the implementation of the FMU actor.

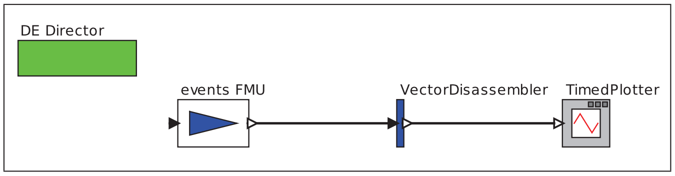


Fig. 5. Ptolemy II model including an FMU actor based on FMI++.

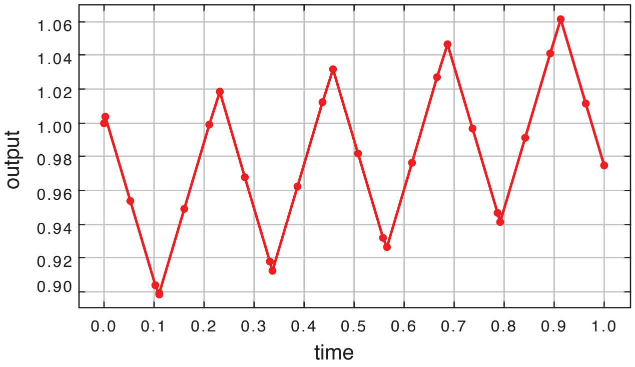


Fig. 6. Resulting output from the FMU actor. Each red dot corresponds to a firing (sync) of the FMU actor.

## V. Conclusions and Outlook

This work presents the basic functionalities of the FMI++ library that extends the functionalities provided via the FMI for Model Exchange specifications. Its features allow to utilize FMUs in a convenient way, simplify their integration in existing simulation environments and enable their use as independent co-simulation components.

Currently, the FMI++ library is still in an early prototype phase. The concepts presented here still need to be thoroughly validated. Further improvements are already planned, e.g. support for automatically unzipping FMUs or improved interpolation algorithms for lookahead predictions.

The FMI++ library intends to promote the FMI specification and encourages the use of FMUs in modern simulation environments. The source code will soon be publicly available, and valuable contributions are heartly welcome.

## References

[1] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clau, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf, "The functional mockup interface for tool independent exchange of simulation models," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, 2011.

[2] P. Fritzson, *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*, 1st ed. Wiley-IEEE Press, 2011.

[3] P. Fritzson, P. Aronsson, H. Lundvall, K. Nystrm, A. Pop, L. Saldamli, and D. Broman, "The OpenModelica Modeling, Simulation, and Software Development Environment," *Simulation News Europe*, vol. 44, no. 45, Dec. 2005.

[4] J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit, "Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problem," *Computers and Chemical Engineering*, vol. 34, no. 11, pp. 1737–1749, Nov. 2010.

[5] K. Ahnert and M. Mulansky, "ODEINT - Solving Ordinary Differential Equations in C++," in *AIP Conf. Proc.*, vol. 1389, 2011, pp. 1586–1589.

[6] B. Schäling, *The boost C++ libraries*. XML Press, 2011.

[7] D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *Transmission and Distribution Conference and Exposition, 2008. IEEE/PES*, april 2008, pp. 1 –5.

[8] P. Palensky, E. Widl, and A. Elsheikh, "Simulating cyber-physical energy systems: challenges, tools and methods," *submitted to IEEE Transactions on Systems, Man and Cybernetics*, 2012.

[9] J. Eker, J. Janneck, E. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong, "Taming heterogeneity - the Ptolemy approach," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 127 – 144, 2003.

[10] D. Beazley, "Automated scientific software scripting with SWIG," *Future Generation Computer Systems*, vol. 19, no. 5, pp. 599 – 609, 2003.

[11] E. Widl and W. Müller, "Linking FMI-based components with discrete event systems," in *Proceedings of the International IEEE Sytems Conference (SysCon)*, 2013, accepted.

# Steady-State Co-Simulation with PowerFactory

Matthias Stifter, *Member, IEEE*, Roman Schwalbe, *Student Member, IEEE*, Filip Andrén, *Member, IEEE*
and Thomas Strasser, *Senior Member, IEEE*

AIT Austrian Institute of Technology, Energy Department, Vienna, Austria
{givenname.surname}@ait.ac.at

***Abstract*—Power system analysis applications like PowerFactory make it possible to investigate research questions within a dedicated domain specific environment. With the increasing complexity in cyber-physical systems the need for coupling models or systems for simulation becomes eminent. By utilizing and extending existing interfacing mechanisms the pros and cons for different coupling approaches under different simulation time scales (steady state, transient) are compared. The tight coupling using steady-state simulation together with external simulators have a significant increase in performance and usability. This paper shows the different possibilities of coupling a power system simulation application, namely PowerFactory, with other continuous and discrete event models and simulators. Selected examples for co-simulation applications are discussed.**

## I. Introduction

The operation of electric power systems is often a complex interaction between the continuous dynamics of physical laws and discrete events (event-driven) behavior. Examples for the first case are physical devices like generators, machines, capacitors, or lines and controllers, market based rules, or on-load tap changers for discrete events.

Such a continuous and discrete dynamic interaction can be for example the tap change process initiated by the automatic voltage controller, which changes the continuous power flow equations before and after the event. Another demonstrative example is given by a line breaker, which disconnects the line and the equations have to be removed from the system model.

While continuous effects can be adequately described by ordinary differential equations (ODE), discrete events can be described by Petri nets and finite state machines [1]. According to [2] the general model for representing power systems is given as

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\varphi}(\boldsymbol{\xi}, \boldsymbol{u}, t)$$

where $\boldsymbol{\xi}$ is the vector of state variables, $\boldsymbol{u}$ the vector of discrete variables, $t$ the time and $\boldsymbol{\varphi}$ the vector of differential equations. Since phenomena in the electric power system (from switching transients over long term dynamics to dispatching) and controls (from power electronics over on-load tap changer to supervisory operation) spans over various time scales, the vector of general state variables can be split into three sub-domains [2]:

- characterized by slow dynamics (big time constants),
- the phenomena under investigation, and
- fast dynamics (small constants).

While the first domain can be considered as quasi-static, or so slow that the effects can be neglected, the third can be considered as algebraic variables (taking instantaneous effect), thus can be discontinuous. The resulting models are a set of non-linear differential algebraic equations (DAEs) with discrete variables.

The system can be formulated as a hybrid dynamical system, or hybrid automation, if it is written as continuous DAEs for every discrete variable change. Hybrid systems are characterized by: continuous and discrete states, continuous dynamics, discrete events, defined transitions from one state to another caused by an event [3].

Nowadays power system simulation software is capable of handling the interaction of continuous and discrete dynamics, like the mentioned example of tripping a line due to a fault and simulating the resulting transients. Usually modern domain specific tools have also some (restricted) possibility to couple other models, simulators or controls, which are defined in an external environment, within the internal system model. This is especially interesting and necessary when more complex system behavior, reflecting the increase of autonomous participants, has to be modeled. In principle the coupling with external or user defined models and the coupling with other simulators are distinguished.

Many co-simulation approaches for power systems analysis have been introduced recently [4], but this paper focuses on coupling or interfacing with PowerFactory.

The combination of stability simulation (RMS) with transients (EMS) within the PowerFactory environment is introduced in [5]. Two simulators are running in master slave mode, where the current is determined depending on the given voltage calculated in the master and injected back. A drawback is that other frequencies than the base can not be considered.

A hybrid simulation coupling of PowerFactory with ICT for real-time applications is presented in [6]. Based on the high level architecture (HLA) [7] this platform brings the domain of wide-area monitoring, protection and control together with power systems (on the bay level) to design and analyst applications in respect to real-time constraints. PowerFactory is coupled via OPC on a 10 ms time scale RMS simulation.

A standardized approach for co-simulation is the specification of functional mock-up interfaces (FMI) for co-simulation and model exchange [8].

This paper will investigate existing program interfaces and their extensions to couple the power system analysis software PowerFactory with external continuous and discrete event systems. The remaining part of it is organized as follows: Section II provides a brief overview of the PowerFactory simulation environment whereas the available interfaces to third-party tools are explained in Section III. The main part of this paper—the co-simulation mechanisms—are discussed in Section IV and some examples are provided in the following Section V. The main findings and conclusions are presented in Section VI and an outlook about further activities is provided.

## II. Power System Simulation

### A. About PowerFactory

DIgSILENT PowerFactory, is a power system simulation and analysis tool with several built-in system models and control algorithms for simulation [5]. Among other functionality, it is capable of doing power flow, short circuit, harmonic, stability simulations like transients as well as steady state simulations for balanced and unbalanced systems. With the DIgSILENT programming language (DPL) the scripting of certain tasks and functionality, like manipulating objects, performing a certain number of power flows for a given time interval, exporting results are possible. While DPL is not available in stability analysis (RMS) mode, the DIgSILENT simulation language (DSL) is used for implementing algorithms and functionality, like controllers, governors.

### B. Simulation time domain

Different time scales or time constants are considered when investigating power system phenomena and controls. They can be divided into the following main simulation types [9].

*1) Electro-magnetic transients (EMT):* The system is fully modeled by state variables and transients are considered by determining all natural and forced frequencies in the systems. The models are described by differential equations for mechanical as well as rotor electrical dynamics.

*2) Root mean square (RMS):* For large-scale stability simulation, only the base (synchronous) frequency is considered and the electrical network parameters are represented as phasors. Especially the differential equations of the synchronous machine model are simplified and the terminal voltage can be represented by a phasor. This can be used for the algebraic equations of the network to solve the stability simulation. While fast transients in the network are not able to be accurate determined, with the internal representation of the synchronous generation based on differential equations (state variables), the machine flux and rotor stability can be investigated.

*3) Power flow (steady state):* In the quasi-static equilibrium of the system, the derivatives of all state variables are zero. The main parameters are the magnitude and phase of each bus voltage and the active and reactive power flow in each line.

### C. Simulation time constraint

*1) Offline simulation:* No constraints are on the computation time for the solution. The main limitation is the performance of the hardware system. If coupling with external models exist, the synchronization and calculation and simulation of this model also effect the throughput.

*2) Real-time simulation:* As it is defined for a real-time constraint simulation, the simulation step has to be finished in a certain time period, otherwise the synchronization is lost. Depending on the phenomena investigated, the time constraint can be from microseconds (for power hardware-in-the-loop) up to minutes and hours (demand or weather forecast). Often also referred as "emulation" [10]. The main advantage of time-synchronized simulation is, that no other simulation control is necessary, to synchronize the individual simulators.

### D. Simulation steps

*1) Integration step size:* Automatic step size adaptation, based on the local discretisation error and the dynamic of the state variables, can be used to increase the step in case of decayed transients [11]. If fixed step size is choosen, the integration step size has to be provided.

*2) Event handling:* In case an event has occurred, the state variables are interpolated to the moment of the event time and the simulation is restarted (see IV-B) [11]. To increase performance, a time interval can be provided, where events are accumulated.

## III. Interfaces for Co-Simulaton

The different possibilities which exists to couple PowerFactory with other models and simulators are explained and their adequacy is investigated in detail in the following sections. An overview of the interface mechanism provided by PowerFactory is given in Figure 1.

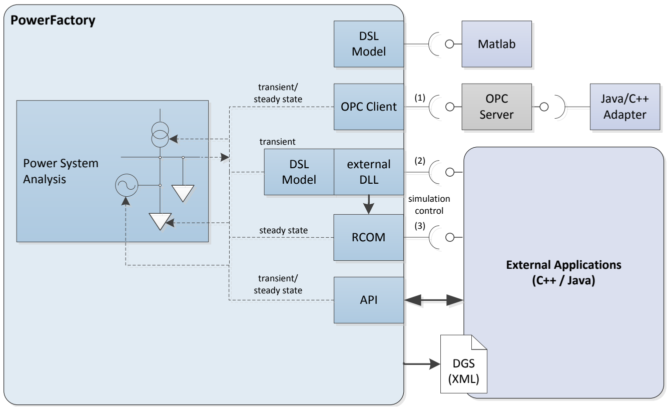


Fig. 1. Overview of the interfaces provided by PowerFactory.

### A. MATLAB/Simulink

A direct interface to MATLAB/Simulink based on PowerFactory frame and slot technique with a built-in RPC interface is provided [12]. It directly invokes the MATLAB/Simulink model which runs in MATLAB runtime engine mode. It has to be noted that this communication link is not standardized as generally proposed in this work.

### B. DLL

The integration of an external 'event driven' C/C++ dynamic linking library (DLL) — `digexdyn.dll`—is possible. It can be integrated via DSL blocks (means only available in RMS simulation) by calling the external defined function (e.g. shunt controller). The result of the evaluated code is back injected into the RMS simulation by emitting events. This event is passed on as a string with the appropriate parameters specified and values assigned (see Listing 1) [13].

```
// step capacitor
void _cdecl StepCapacitor(double tEvnt,double** dParams,...)
{
  // create a PowerFactory paramter event to set
  // the actual step of the capacitor at delta time = Tdelay
  sprintf(eventstr[0],”create=EvtParam target=Shunt name=
         CapStep dtime=\%f variable=ncapa value=3”,Tdelay);
}
```

Listing 1. Returning the result as an event into transient simulation (RMS).

## C. OPC

The OPC [14] client interface is an asynchronous communication and data exchange mechanism used in process interaction [9]. The data transfer is synced automatically in RMS simulation according to the update frequency, but which is only practicable in combination with the real-time PC-clock synchronization. Otherwise the simulation is too fast to reflect the data exchange update (see Section IV-A). During power flow simulation, the OPC data exchange has to be carried out through an explicit DPL command (see Listing 2).

```
  ! connect OPC client
  status_OPC = Link:isLinkStarted;
  if (status_OPC = 0){
     Link.Execute();
     Link.SetOPCReceiveQuality(255);
  }
  ! initialise if not part of calculation relevant objects
  Variable.InitTmp();
  nDataNum = Link.ReceiveData(1);
  if(nDataNum > 0) {
     Variable.GetMeaValue(value);
  }
  ! disconnect OPC client
  Link.Execute();
```

Listing 2. Manually receiving OPC data with DPL.

Every writable variable needs a so called “trigger” object which sets the value of the variable (see Listing 3).

```
  oTriPFready:ftrigger=value;
  Variable.SetMeaValue(value);
  nDataNum = Link.SendData();
```

Listing 3. Manually sending OPC data with DPL.

## D. RCOM

A command server functionality is available for remote procedure calls (RPC) with client/server communication. Most of the DPL scripts and direct commands are available, like activating projects and performing simulations. This is used to run PowerFactory in “engine mode” to perform various batch jobs for simulation (Listing 4).

```
  digrcom -c=InitSimulation
  digrcom -c=ldf
  ...
  digrcom -d -p ncacn_ip_tcp -n 127.0.0.1 -e 2001 -f=”
         example.cmd”
```

Listing 4. Invoking a single or a batch command file via the RPC interface.

## E. API

An application programming interface (API) encapsulates the internal data model and various simulation functionality in PowerFactory. The application singleton instance provides the access to all project related data objects and their corresponding values (of variant—e.g. string, double, vector). This interface has a low level abstraction and functionality for accessing topology and performing simulations, but has been extended with a wrapper to provide more convenient functionality. The “PFSim” wrapper can read and write attributes, automatically assign load profiles according to the column header of a given csv source file and perform state machine TCP/IP socket synchronizable offline and real-time simulation (see Figure 2).

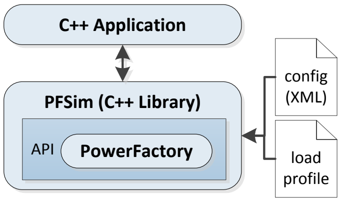
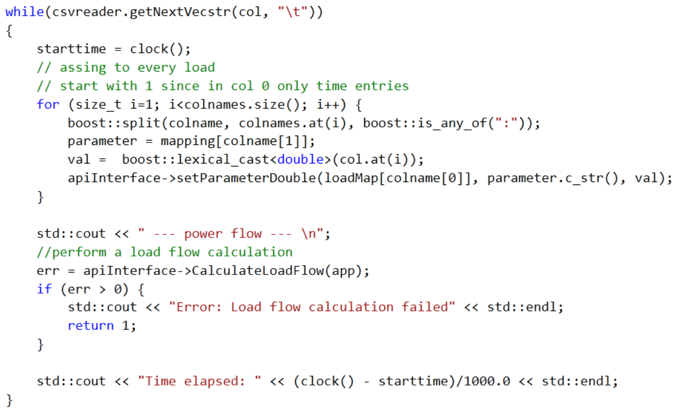


Fig. 2. Stand-alone application using the PowerFactory API via PFSim wrapper.

Figure 3 shows an example code of performing a simulation with the developed PFSim wrapper.



```
while(csvreader.getNextVecstr(col, "\t"))
{
    starttime = clock();
    // assing to every load
    // start with 1 since in col 0 only time entries
    for (size_t i=1; i<colnames.size(); i++) {
        boost::split(colname, colnames.at(i), boost::is_any_of(":"));
        parameter = mapping[colname[1]];
        val =  boost::lexical_cast<double>(col.at(i));
        apiInterface->setParameterDouble(loadMap[colname[0]], parameter.c_str(), val);
    }

    std::cout << " --- power flow --- \n";
    //perform a load flow calculation
    err = apiInterface->CalculateLoadFlow(app);
    if (err > 0) {
        std::cout << "Error: Load flow calculation failed" << std::endl;
        return 1;
    }

    std::cout << "Time elapsed: " << (clock() - starttime)/1000.0 << std::endl;
}
```

Fig. 3. Listing example for performing power flow for a given load profile with PFSim wrapper.

# IV. Co-Simulation Mechanisms

## A. Synchronization

In general the following two mechanisms are distinguished when a model is coupled with PowerFactory:

*1) Sequential co-simulation:* In one case the simulation of the model is directly invoked sequentially from within the simulation loop. This means that it is an integral component in DSL coupled to MATLAB/Simulink or an external DLL. When this model equation is solved, the execution thread runs into the function representing the model and blocks until the simulation terminates. Figure 4 shows the invocation of the external model, blocking the execution of the simulation. In this figure the model is implemented in an external C++ DLL and can return results only via events emitted into the RMS simulation. Note that if the external model is realized with the direct MATLAB/Simulink interface, the DSL model is continuously updated and the output is immediately used in the simulation (model needs not to emit events).

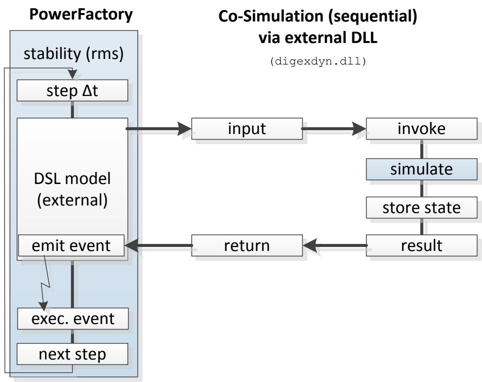


Fig. 4. Sequential execution of every simulators for a simulation step from the simulation master (simulation control).

*2) Parallel co-simulation:* In the second case, the simulation of the model runs in parallel and the input and output data are exchanged via a communication link. Since the external simulation has to be controlled in terms of the simulation step, both have to be synchronized via flags (or similar mechanisms). Figure 5 demonstrates the parallel simulation processes, waiting of each other's input to perform the simulation.

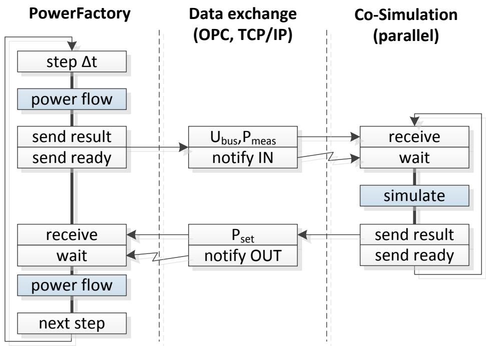


Fig. 5. Coupled simulation in parallel running processes depend on synchronization.

### B. Iterations and accuracy

As described in Section II, if an internal event occurs the simulation time is set to the exact moment in time where it took place, thus preventing any integration error or model error (see Figure 6). If the event happens in between a simulation step, than the simulation is interpolated at the exact event time.

When the parallel co-simulation is controlled by the iterative execution of power flows from a DPL script, the actual simulation time step can be simulated again. Since the power flow is a steady-state for a certain moment in the simulation time, a controller's change on the network model (e.g. change the tap position, or the active power of a load) can be immediately taken into account for the same time step (see Figure 7, upper part). It is even possible to have several iterations of the same time step, until the controller's output has converged. Note here, that no dynamics of the controller transients can be simulated in this case. The resulting steady state simulation is valid for this moment in time.

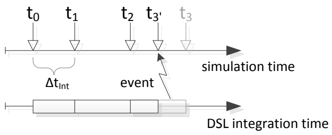


Fig. 6. If an event occurs (in sequential co-simulation), the simulation is carried out at the exact moment in time, even between a time step or back to the moment where the event took place.

When a continuous time model is co-simulated, the time interval of the simulation step is needed to accurately integrate the model and represent its internal states. If the resulting output is communicated to PowerFactory before the next simulation step, there will be no simulation error due to lagging behind. (see Figure 7b).

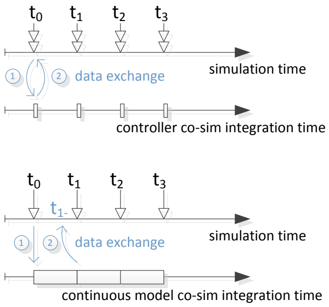


Fig. 7. In case of parallel co-simulation when the execution of the simulation steps are synchronized: a) the changes of the model or controller can be exchanged immediately, or b) changes are exchanged before the next simulation step takes place.

Parallel co-simulation within real-time constraints, synchronized e.g. via PC clock, is affected by the lag in communicating the results from the co-simulator back. Because of the time constraints for every simulation step, the changes imposed by the co-simulated model or controller only effect when the next simulation step takes place. Figure 8 illustrates the asynchronous time steps of the time synchronized parallel simulation of the two simulators and the lag $\Delta t_{lag}$ until the effects are reflected. This lag results from the delay of the communication, the computational performance of the co-simulation and the step size of the simulation.

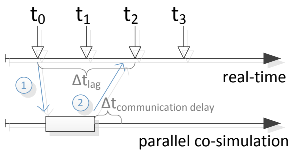


Fig. 8. If there are no synchronisation mechanism (e.g., PowerFactory and controller in a real-time) the parallel running simulators can lag behind, due to communication delay and asynchronous time steps.

## V. Application Examples Coupling PowerFactory

### A. Model validation

For automatically assigning smart meter active and reactive power measurement data to a low voltage network model and for validating the simulation results with the measured voltages, an analysis environment has been introduced in [15]. For this purpose a Java application's input and output stream are connected to the RCOM interface.

Although in this specific case measurements of the real process and not another simulation environment were coupled with PowerFactory and demonstrates merely a way to interface the simulation. A possible use case with this interface for co-simulation could be the implementation of building models for demand side management applications and shifting energy demand in time according to meet given objective. This interface based on RCOM is not applicable for transient analysis, since there is not possibility to provide state variables and there derivatives. Therefore the models implemented in this setup are restricted to steady state investigations and benefits are: easy to implement, RPC (remote procedure call) based interface, controllability of simulation.

### B. Distributed energy resources and energy management

In [16] a framework for co-simulating a continuous model of a electric energy battery in MATLAB/Simulink with connection to the electric network modeled in PowerFactory is presented. The battery model is simulated on a smaller time scale and synced with PowerFactory, where the simulation control and the iteration of the power flow for the actual simulation time step with the results from the battery model is performed. The MATLAB/Simulink based model of the battery gives much more flexibility of the implemented battery control and energy management system and allows to use existing models (e.g., model of a vanadium-redox-flow battery, based on chemical equations). The interface is based on exchanging state variables and derivatives, therefore the model can be represented in differential equations. This coupling is appropriate for transient simulations.

Due to internal restrictions of running the MATLAB/Simulink model instance, all parameters have to be sent for each step. If the internal states are stored in the workspace, the number of necessary state variables and parameters exchanged with PowerFactory can be reduced. Figure 9 shows MATLAB/Simulink, which runs in 'engine mode', exchanging results with PowerFactory.

### C. Communication and control

In [17] co-simulation of a power system and a voltage controller is presented, which takes the underlying communication system model into account. This setting makes a co-simulation setup with other simulators possible. The PFSim library (PowerFactory API wrapper) supports power flow and other functionality as well as offline and real-time RMS simulation. In case of offline RMS simulation the simulation control forces the next simulation time step $\Delta t$ and PFSim advances the RMS simulation about that time interval. After the simulation has finished, the data objects' attributes can be accessed.

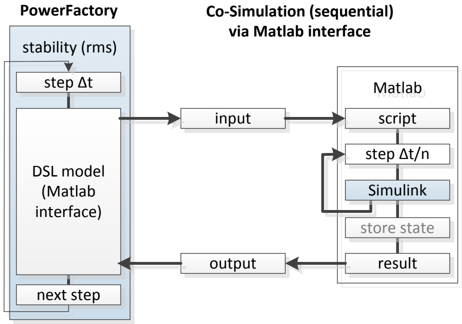


Fig. 9. Co-simulation with MATLAB/Simulink using the PowerFactory's build interface. Note that internal states have to be stored explicitly.

In case of real-time simulation, the blocking RMS command [sim] has to be started in a dedicated thread. While reading the data objects' attributes is possible, the changing or writing of attributes is only possible via thread-synchronized queues. The external DLL realization of the DSL function injects the new value via PowerFactory's asynchronous event support (see Figure 10).

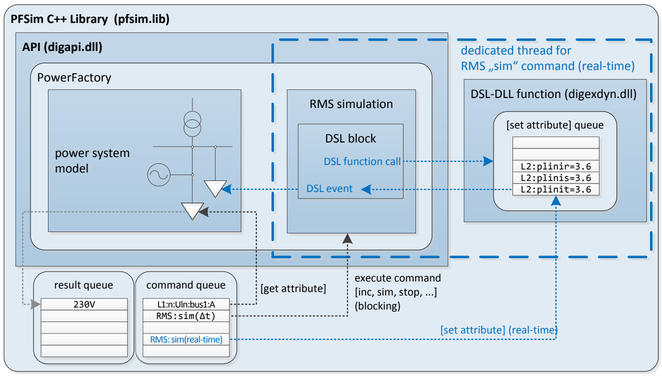


Fig. 10. Co-simulation with the API wrapper 'PFSim' for offline and real-time simulation (emulation). Data objects' attributes cannot be set during RMS simulation via API. The RMS [sim] command is blocking and needs to be executed in a separate thread. Data exchange needs to be handled via synchronised queues.

Since the PFSim library itself can be used in co-simulation environments, results are put into thread-safe queues, where they can distributed over communication links with other simulation components. Additionally the library supports basic simulation controls, like initialization, step, start, stop for power-flow and stability simulation (RMS) both for offline and real-time simulation.

### D. Voltage control

Simulation and validation of a coordinated voltage controller under real-time constraints for medium voltage networks has been published in [18]. Offline as well as scaled real-time RMS simulation via OPC interface has been used to successfully develop and deploy the control algorithm. The controller, any implemented prototype or functional device (e.g. protection relay) can be coupled with the simulated system in a real time based controller hardware-in-the-loop (C-HIL) setup. Real time simulations constraints allows the coupling of physical and simulated models.

### E. SCADA/control system validation

Another interesting example is the validation of the implementation of a supervisory control and the data acquisition (SCADA) system and connected distributed controllers for power distribution systems [19]. For coupling the PowerFactory simulation environment with the distributed controllers the OPC interface with an OPC server (according to Figure 1) has been used. The communication between the controller and the SCADA tool was carried out using standard Ethernet-based communication protocols (e.g. TCP/IP, UDP/IP, IEC 61850, IEC 60870-5-104). An overview of this example is provided in Figure 11. The controls can be directly deployed to the target platforms (e.g., embedded system) and are purely event-driven, which is not possible to implement in PowerFactory.

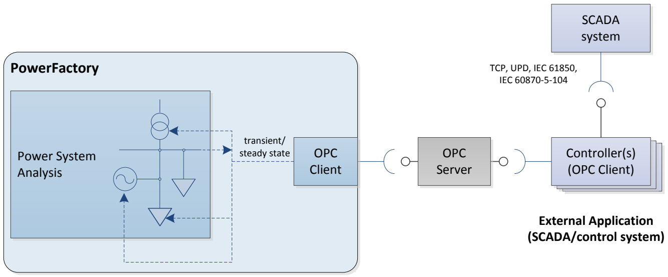

SCADA system
TCP, UPD, IEC 61850, IEC 60870-5-104
PowerFactory
Power System Analysis
transient/ steady state
OPC Client
OPC Server
Controller(s) (OPC Client)
External Application (SCADA/control system)

Fig. 11. Co-simulation for SCADA/control system validation.

### F. Comparision and applicability

A comparison of the different possibilities to couple models and simulators with PowerFactory is given in Table I.

TABLE I. COMPARISON OF SIMULATION COUPLING FOR POWERFACTORY.

| PowerFactory interface | power flow | RMS | EMT | simulation control | coupling | |
|---|---|---|---|---|---|---|
| | | | | | sequ. | parallel |
| MATLAB | - | × | × | - | × | |
| DLL | - | × | × | - | × | |
| OPC | × | $\times^1$ | - | - | × | $\times^2$ |
| RCOM | × | $\times^3$ | - | × | | × |
| API | × | $\times^1$ | × | - | $\times^4$ | |
| API+DLL+RCOM | × | × | × | × | × | $\times^5$ |

[1] timing constraints, when it comes to real time simulation
[2] synchronization necessary
[3] no access to variables during RMS simulation
[4] API can change variables between simulation time steps
[5] multiple threads and queued synchronization are necessary

## VI. Summary and Conclusions

The popular PowerFactory simulator for power systems analysis has been introduced and an overview of possibilities to interface other external models, controllers or simulators have been given in order to cope with the increasing complexity in power systems. The majority of presented co-simulation approaches in this paper is applicable for simulating steady-state models and controllers. Simulating transient phenomena would require the exchange of state variables, like PowerFactory's integrated MATLAB/Simulink interface mechanism provides. Moreover, the overview of selected simulation examples has shown the applicability of the different interfacing possibilities. Future work will investigate the possibility to couple dynamic transient models with PowerFactory.

## Acknowledgement

This work is supported by the Austrian Climate and Energy Fund and by the Austrian Research Promotion Agency (FFG) under the project *"DG-EV-HIL"*.

## References

[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, Nov. 2007.

[2] F. Milano, *Power System Modelling and Scripting*. Springer, Jan. 2010.

[3] I. Hiskens, "Power system modeling for inverse problems," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 3, pp. 539–551, March 2004.

[4] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, Jan. 2011, pp. 1–6.

[5] DIgSILENT, "Combined Transient and Steady State Simulation (RMS-EMS)," DIgSILENT GmbH, Tech. Rep., 2007.

[6] S. C. Müller, H. Georg, C. Rehtanz, and C. Wietfeld, "Hybrid simulation of power systems and ICT for real-time applications," in *3rd IEEE PES Innovative Smart Grid Technologies (ISGT) Europe Conference*, Berlin, Germany, October 2012.

[7] IEEE, "IEEE Standard for Modeling and Simulation (M & S) High Level Architecture (HLA) - Framework and Rules," *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000) - Redline*, pp. 1–38, Aug. 2010.

[8] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clau, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf, "The functional mockup interface for tool independent exchange of simulation models," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, 2011.

[9] DIgSILENT, "PowerFactory User's Manual 14.1 - General Information, Volume I," DIgSILENT GmbH, Tech. Rep., 2012.

[10] M. Steurer, C. Edrington, M. Sloderbeck, W. Ren, and J. Langston, "A megawatt-scale power hardware-in-the-loop simulation setup for motor drives," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 4, pp. 1254–1260, April 2010.

[11] DIgSILENT, "PowerFactory User's Manual 14.1 - Power System Analysis Functions, Volume II," DIgSILENT GmbH, Tech. Rep., 2012.

[12] MATLAB/Simulink, "The language of technical computing," Access Date: February 2013. [Online]. Available: http://www.mathworks.com

[13] DIgSILENT Support, "Example of a simple shunt controller," DIgSILENT GmbH, Tech. Rep., 2010.

[14] Y. Shimanuki, "OLE for process control (OPC) for new industrial automation systems ," in *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 6, 1999, pp. 1048–1050.

[15] M. Stifter, B. Bletterie, D. Burnier, H. Brunner, and A. Abart, "Analysis environment for low voltage networks," in *Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on*, Oct. 2011, pp. 61–66.

[16] F. Andrén, M. Stifter, T. Strasser, and D. Burnier de Castro, "Framework for co-ordinated simulation of power networks and components in smart grids using common communication protocols," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Nov. 2011, pp. 2700–2705.

[17] F. Kupzog, P. Dimitriou, M. Faschang, R. Mosshammer, M. Stifter, and F. Andren, "Co-Simulation of Power- and Communication-Networks for Low Voltage Smart Grid Control," in *D-A-CH-Konferenz Energieinformatik*, Oldenburg, Germany, May 2012.

[18] F. Andrén, S. Henein, and M. Stifter, "Development and validation of a coordinated voltage controller using real-time simulation," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Nov. 2011, pp. 3713–3718.

[19] T. Strasser, F. Andrén, F. Lehfuss, M. Stifter, and P. Palensky, "Online Reconfigurable Control Software for IEDs," *Industrial Informatics, IEEE Transactions on*, vol. accepted for publication, 2013.

# Integrating Households into the Smart Grid

Andrea Monacchi*, Dominik Egarter*, Wilfried Elmenreich*†

*Institute of Networked and Embedded Systems,
Alpen-Adria-Universität Klagenfurt / Lakeside Labs, Austria
andrea.monacchi@aau.at, dominik.egarter@aau.at, wilfried.elmenreich@aau.at

†Complex Systems Engineering, University of Passau, Germany
wilfried.elmenreich@uni-passau.de

*Abstract*—**The success of the Smart Grid depends on its ability to collect data from heterogeneous sources such as smart meters and smart appliances, as well as the utilization of this information to forecast energy demand and to provide value-added services to users. In our analysis, we discuss requirements for collecting and integrating household data within smart grid applications. We put forward a potential system architecture and report state-of-the-art technologies that can be deployed towards this vision.**

**Keywords: smart energy, smart home, smart appliances, semantic sensor networks**

## I. Introduction

The smart grid is a cyber-physical system combining the electricity infrastructure with an informative channel to connect producers and consumers to become more efficient and reliable. The presence of this data flow will result in modifications to the current energy market, such as new tariff plans based on the actual availability of energy. This aspect is getting more important, as the grid will have to cope with fluctuations produced by a dynamic energy demand and with the integration of renewable-energy plants, which might depend on the availability of sun and wind. Banks of batteries might be used to mitigate such variations, by storing energy when demand is low and a high amount of energy is available. In this way, the use of conventional (i.e., nuclear, fossil fuel) power plants can be kept to a minimal level by relying on them only in case of high energy demand.

Households play an important role in the grid, as users can aim to contribute to an idea of sustainable living and receive tailored services, for instance, appliances can be scheduled to postpone certain tasks and avoid running in periods of peak demand (when the energy might come at a higher price).

Persuasive technologies, such as feedback systems providing real-time energy monitoring and recommendations, have been shown to raise the awareness of users about their energy consumption and inducing a long-term change in their behavior and lifestyle [1]. However, studies show that the effectiveness of these systems in making people responsible depends on their sensitivity and motivation [2]. In addition, intelligent controllers can be used to optimize the running costs of the household, while simultaneously increasing the comfort of inhabitants. This requires taking continuous changes in the grid into account, which is only possible in presence of a reliable and shared data infrastructure.

To achieve this, we argue that openness of protocols and data is necessary to provide the stakeholders of the smart grid with a shared development framework. Indeed, the household is an ecosystem of highly heterogeneous digital devices, using different protocols and data representations. The creation of an advanced metering infrastructure that is able to collect and analyze energy consumption information down to the device level is expected to extend such applications to the global market.

In this paper we advocate for the integration of data produced within domestic environments into the smart grid. Our contribution is twofold: presenting design guidelines and identifying technologies that can be deployed towards this vision. In Section 2, we outline potential requirements for a data infrastructure that can mediate the interaction between the stakeholders of the smart grid, such as users in domestic environments, energy providers and developers of applications. To effectively implement the architecture, the paper proposes potential solutions which emerge from current research trends. In particular, we identify open protocols that can enhance embedded devices with networking capabilities. Section 3 regards current solutions to integrate devices into the Internet. In addition, we propose using techniques from the semantic web for allowing devices to describe their data. Section 4 suggests ways of annotating data and devices using the Resource Description Framework (RDF) model, as well as mechanisms to retrieve descriptions in a wide scale network.

## II. Requirements

A reliable and flexible infrastructure able to tackle different demand is required to handle the amount of data produced by heterogeneous devices in highly dynamic environments. We have identified the following requirements:

- **Plug & play mechanism:** The architecture should support discoverability of services and resources in the network so that they can be used as soon as they become available [3]. Services are required to provide a description of their characteristics that can be advertised to other peers or retrieved when needed.
- **Accessibility of data:** To increase interoperability and reduce maintainance costs, the architecture should support a data-driven abstraction. This data space should be accessible through a uniform interface, such as standard query languages and a well defined API. This requires sensed data to be semantically annotated

according to well-known design patterns and vocabularies, and enhanced by situational information such as time and space.

- **Reliable and neutral data infrastructure:** Knowledge and information about properties and context should be stored in a repository that can ensure availability and continuity of service. Moreover, it should support integrity of data and avoid any discrimination that is not strictly required to guarantee Quality of Service.
- **Confidentiality:** The architecture should provide mechanisms to avoid unauthorized disclosure of information. In particular, it should secure access to the home network and the repository (e.g., using authentication and encrypted communication) and use a sandbox mechanism for applications when accessing user data (e.g., OAuth protocol[1]).
- **Quality of data:** The architecture should ensure appropriateness of data, i.e., consistence with respect to time. For instance, event-oriented systems may not be able to meet strong real-time constraints, as events are queued for an unpredictable time before being dispatched.

### A. *A potential architecture*

A potential architecture addressing all the requirements should firstly support openness of data and protocols. The household can be considered as a network of self-describing sensors and actuators, that can dynamically join and leave the network with the help of service discovery mechanisms. Such devices are producers and consumers of data. Therefore, the household can be seen as data space, where data are described with respect to shared vocabularies and well-known design patterns, and can be retrieved using standard query languages. For instance, data can be produced by meters, appliances and generators and related to situational information such as time and space. A smart appliance is aware of its consumed power, based on voltage and current measurements or built-in profiles [4]. This requires a machine-readable description to be deployed with the smart device to describe its properties and functionalities, so that they can be automatically used by other machines. In addition, such description can also be used as a reference for describing data produced by the appliance. For instance, a washing machine might be reachable at a certain URI (Uniform Resource Identifier) and return metadata describing the device, such as profile listing the type and the expected consumption for a certain task. The appliance may provide various features, such as measuring its actual consumption and returning it as a stream annotated with respect to the device profile.

A gateway is used to bridge the home network to the Internet, and to ensure security when accessing home devices. Moreover, it plays also a crucial role in the integration of smart and legacy devices, which can not provide their metadata, and thus, would not be accessible within this architecture. Non-intrusive load monitoring (NILM)[5] can be applied to disaggregate the power profile of attached loads out of the overall household consumption. Accordingly, metadata can be dynamically associated to the power profile of running appliances in order to be directly exploited by applications.

The data space should not be localized to the house, as such data can be useful to build a more accurate understanding of the system and offer tailored functionalities, which can take dynamic variations of the market and the demand into account. We look at cloud-computing technologies to cope with the huge amount of data and computing power required to manage the repository. The development of batch and real-time analytics solutions, such as Hadoop[2] and Storm[3] respectively, enables the stakeholders of the smart grid to perform complex data analysis, such as forecasting of future demand. Therefore, future home management systems will be able not only to optimize energy consumption but also to track the inhabitants' activities, and offer complex tailored functionalities. Rule-based semantic reasoning can be used to infer further knowledge and produce more abstract situational information that can be directly exploited by decision makers implemented as applications. This will provide users a set of value-added services and help data providers pay off the operational costs of the repository. A sketch of our architecture is shown in Fig. 1. In the following sections we describe in greater detail potential technologies that could be employed to implement this architecture, in terms of communication infrastructure and ways of semantically annotating data and resources so that they can automatically be consumed in a network of heterogeneous entities.

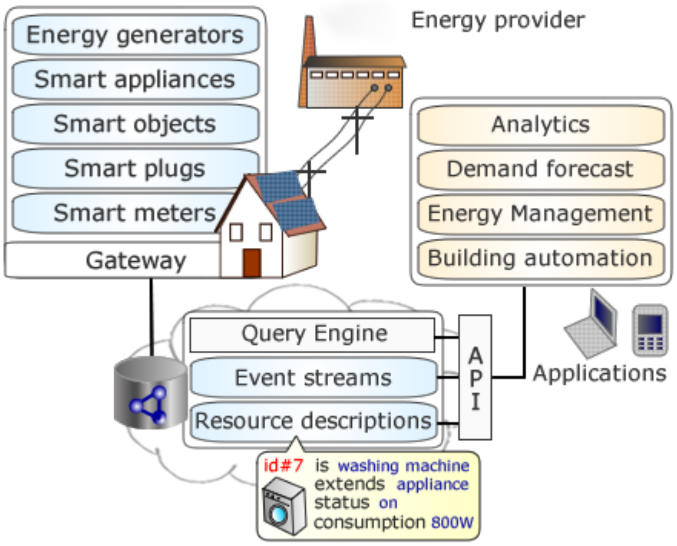


Fig. 1. Architecture to provide data interoperability in the smart grid: data produced in households is semantically annotated and fed to the cloud. A query engine provides a uniform interface to data that can be exploited by applications.

## III. The communication infrastructure

As shown in [6], a web-enabled smart grid can be realized by deploying inexpensive devices to monitor energy-related aspects such as power quality. Bringing the web into

[1] http://oauth.net/

[2] http://hadoop.apache.org/

[3] http://storm-project.net/

constrained networks requires embedded devices to use a lightweight and interoperable stack of protocols that is fully compatible with the existing Internet. This is already possible, as some lightweight versions targeting embedded systems have already been implemented. The constrained application protocol (CoAP)[4] has been proposed as alternative to HTTP that can be run by deeply embedded systems (e.g., 8-bit microcontrollers). It uses UDP with a simple retransmission mechanism where each GET request is associated with a unique identifier. In addition, it provides an eventing mechanism that allows subscription to updates of a certain resource. In the REST model [7], resource discovery takes place by accessing an index page listing links leading to resources on the same or different servers. Therefore, it may be argued that service-oriented architectures such as the DPWS[5] are more powerful than classical REST designs, especially in terms of service discovery. CoAP tries to remedy this problem by using the CoRE link format (RFC6690)[6], where resources can be explained by meaningful links. Since links define relationships between web resources, assigning a meaning to links enables machines to automatically calculate how to use resources on a server. In particular, a client can contact the server using a GET to a predefined location, i.e., /.well-known/core, which returns the list of resources exposed by the server and their media type. This means that nodes can act as resource directories listing links to resources stored on other nodes. To be visible to other clients, resource providers can POST a link to their resources to the "/.well-known/core" position of the chosen directory node. For instance, a washing machine could post its description on the gateway, in order to be visible within the network. An energy management system might need to retrieve the current consumption of the appliance by performing a GET to the sensing resource of the node "washing machine" (e.g., *coap://DNS-name-or-IPv6-address-of-washing-machine/consumption-sensor*). Similarly, the application could act on the appliance by using POST to modify the state of a certain resource. For instance, the system could update the information of the current energy cost to enable localized decision-making tasks, as well as directly controlling the device (e.g., switching it on/off) when certain situations occur. CoAP has been implemented in different programming languages: C (libCoap[7][8] and Erbium [8][9]), C++ (evCoap[9]), Java (Californium[10], JCoAP[11]), Javascript (Copper[12]). Along with 6LoWPAN[13], CoAP is the ideal candidate to integrate real world services into the standard internet network (Fig. 2). Therefore, web servers can be run on embedded devices (e.g., washing machine) and used to interact with humans (i.e., providing a service), as well as machines (i.e., returning machine-readable information). Security in real world objects is also crucial. A secure CoAP (CoAPs) using a compressed DTLS over 6LoWPAN networks is discussed in [10].

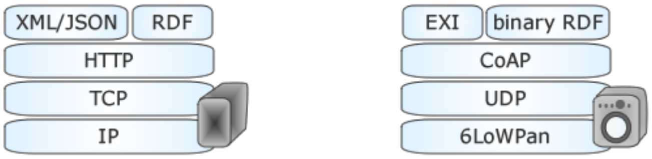


Fig. 2. The HTTP-CoAP mapping: integrating embedded systems in the internet

### A. Reducing the data representation overhead

Data interoperability in the interaction is ensured by standard data-interchange formats, such as XML and the lighter Javascript Object Notation (JSON). Binary variants have been proposed to reduce the overhead of managing these formats in deeply constrained embedded devices. The Efficient XML interchange (EXI) [14] provides an efficient way of processing XML information. It is shown to provide the highest compression and compactness compared with respect to other representations [11]. An implementation is proposed by [12]. Its counterpart is the binary JSON format[15].

### B. A CoAP gateway for the home

In our proposed architecture, the gateway acts as sink and resource directory for the network. Networked devices, such as smart appliances, can interact with the gateway by registering themselves and providing their description when necessary. The *if* (interface description) attribute of the CoRE Link format can be used to refer to an external machine-readable document describing the resource. The description is usually specified in formats such as WSDL or WADL, which can only capture aspects related to the interface. However, developing applications as workflows of CoAP web services requires human involvement. According to [13], machines should be able to figure out which operations can be executed on resources, and which effects operations produce in a certain situation. In this way, they can use a uniform interface to retrieve descriptions and autonomously manipulate resources according to a certain goal. RESTdesc[16] is a mechanism according to which RESTful webservices can be described in terms of pre- and post-conditions in the Notation-3[17] format. For instance, it can be used to define the effects of performing a POST method, which is strictly dependent on the application logic and the data passed with the request. In [14], RESTdesc is used to show the composition of sensor web services for the reservation of tables in a restaurant. However, to truly integrate household appliances and devices within the household, and consequently in the smart grid, the gateway will have to mediate between smart and legacy devices. Therefore, it will have to connect to a sensing system in order to detect running devices. On one hand, smart outlets can be connected to running devices to track their power consumption. On the other hand, non-intrusive load monitoring (NILM) [5] exploits optimization [15] and machine-learning techniques to disaggregate the power profile of loads out of the overall household consumption. For instance, a refrigerator can be identified by

[4] http://tools.ietf.org/html/draft-ietf-core-coap-12

[5] http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

[6] http://tools.ietf.org/html/rfc6690

[7] http://sourceforge.net/projects/libcoap/

[8] http://www.contiki-os.org/

[9] https://github.com/koanlogic/webthings/tree/master/bridge/sw/lib/evcoap

[10] https://github.com/mkovatsc/Californium

[11] http://code.google.com/p/jcoap/

[12] https://github.com/mkovatsc/Copper

[13] http://tools.ietf.org/wg/6lowpan/

[14] http://www.w3.org/TR/exi/

[15] http://bjson.org/

[16] http://restdesc.org/

[17] A RDF serialization format. http://www.w3.org/TeamSubmission/n3/

its periodic power profile, whereas a certain sequence of state changes can help detect a multi-state appliance, such as a coffee machine. Detected appliances can be represented on the gateway as virtual resources, returning their actual and expected consumption. This requires the definition of resource profiles, which can be dynamically filled out by the gateway to represent these resources. In this way, the architecture is capable of collecting energy consumption information from any device in the household. Consequently, applications can already exploit this information, without waiting for the market to provide effective development frameworks for producers of devices.

## IV. Enabling data interoperability

Linked data is a paradigm which aims at leveraging the web from a collection of documents to a network of interlinked data called the Web of Data. Beside the web of hyperlinked HTML (HyperText Markup Language) documents, Linked Data suggests the use of RDF (Resource Description Framework) for describing the content of documents, in terms of relationships between data. In the RDF data model, the basic unit of information can be described as *subject*, *predicate* and *object* triple. A predicate is a property of the subject entity and provides a connection to another entity (object) or a literal value (e.g., string). This means that various triples denote a directed graph, where nodes can represent real-world entities (e.g., people, places) or abstract concepts. In order to avoid naming ambiguities, RDF uniquely assigns URIs to resources. Since many URIs are also URLs (i.e., the set of names used to identify web resources), many people erroneously use them interchangeably although they do not denote the same set of resources (i.e., URLs $\subset$ URIs). Links can be used to specify properties of certain resources, by providing connections to vocabularies written in the RDFS (RDF Vocabulary Definition Language) and OWL (Web Ontology Language) formats. In addition, links can also refer to entities residing on other data sources, so that the web of data can be navigated by humans and crawlers following RDF links. This is a clear advantage over other kinds of web services, as rather than using customized mashups built from statically chosen data sources, data can be directly discovered and integrated in the workflow without human intervention [16]. We refer to [17] for a thorough introduction to linked data.

Ontologies can be used as agreed vocabularies between data consumers and data producers, as ontologies represent knowledge in a certain domain. Therefore, data can be semantically annotated with respect to concepts defined in such vocabularies. For example, a description of a washing machine could be defined in Turtle[18] as:

```
@prefix ns: <http://myrepository.com/houses/12345/> .
@prefix en: <http://example.com/ontologies/appliances.owl#> .

ns:washing-machine en:model         "XYZ123456" ;
                   en:manufacturer  "Bob Inc." ;
                   en:type          en:washing-machine ;
                   en:consumption   "800" .
```

This could be deployed with the appliance and retrieved at the URI "http://myrepository.com/houses/12345/washing-machine", which might link to the actual IPv6 of the washing machine. A common strategy[19] is to retrieve resources using their URI and exploit headers to specify whether retrieving a web page or an RDF description. In this way, humans (via a web browser) and machines can access two different representations of the same resource, identified by its URI. In HTTP, the URI of the representation is returned to the client using the code 303 (i.e., see others). The client can then retrieve it through a common GET request. A straightforward way to exploit this semantic description would be to dereference URIs and following RDF links to navigate the graph and discover other resources on the fly. For instance, in the proposed architecture the URI of the gateway could be an informative point to discover resources in the household. However, this might be a slow process when done in a large scale network, such as the Web. A common solution used in search engines is to navigate the graph off-line using a crawler. Collected data is processed and stored in a persistent storage such as a triple store. In our architecture, smart devices maintain a description of their capabilities and properties that can be retrieved at a certain URI, as well as streams of linked sensor data that refer to such descriptions. However, to provide applications with an efficient way to retrieve stream data from the network of constrained devices, we collect sensor data in a reliable data repository (i.e., a triple store) and we provide an endpoint where queries can be executed. In fact, managing heavyweight knowledge representation techniques on resource constrained devices might not be possible, as they would produce obsolete results in highly dynamic environments. The introduction of lightweight semantic tools and models, such as the binary RDF[20] representation, is therefore required to bring semantic technologies to these networks.

### A. Towards a widely accepted ontological framework

The definition and agreement of common vocabularies is crucial to extend an ontological framework to a global scale, which means that everyone sharing a certain vocabulary will be able to interpret information. Indeed, data might use different vocabularies to represent the same concept. Therefore, to provide a homogeneous annotation of data it is important to converge to a single vocabulary (e.g., translating data coming from different data sources), as well as defining correspondences between concepts defined in different vocabularies (i.e., ontology alignment). The W3C Semantic Sensor Network Incubator group addresses this problem by providing a standardized vocabulary. The SSN-XG ontology[21] is an OWL ontology that describes sensors in terms of accuracy and capabilities, as well as observations that can be drawn from the recorded data [18]. The ontology is designed according to the stimulus-sensor-observation design pattern, which separates environment stimuli from the kind of sensor used for the data collection [19]. This allows the reuse of stimuli, sensors and observations in different contexts. Information on how to enrich sensor data within the SSN-XG ontological framework is shown in [20]. In particular, they provide information about the type of data collected: where, when and under which conditions. A specific ontology for capturing event data in smart grids is proposed in [21], whereas [22] introduces the

[18] Terse RDF Triple Language, a RDF serialization format. http://www.w3.org/TeamSubmission/turtle/

[19] http://www.w3.org/TR/cooluris/

[20] http://www.w3.org/Submission/2011/03/

[21] http://purl.oclc.org/NET/ssnx/ssn

*Smart Appliances Ontology Model*, an ontological framework for smart appliances.

### B. *Querying the web of sensor data*

Linked data is described as set of RDF triples denoting a labeled graph. The possibility of retrieving data from such graphs by expressing complex queries across diverse data sets is therefore the key for automatic composition of data and services. SPARQL[22] (SPARQL Protocol and RDF Query Language) is the most diffuse query language for retrieving data defined in the RDF format. Starting from version 1.1, SPARQL provides the possibility of adding, deleting and changing data triples. These kinds of features are expected to provide enough flexibility to RDF definitions. For instance, applications might require updates of certain properties of running devices, as well as addition and removal of certain relationships between data to encode certain rules in the knowledge base. For this reason, [23] foresee the evolution of the Web to a read-write space, where machines can collaborate. In this scenario, potential concerns related to the provenance and trust of this data arise, as well as the permission to manipulate them in applications.

As shown in [24], SPARQL can also be used to perform complex event processing over RDF data. A mechanism to use SPARQL for defining rules and constraints on semantic-web graphs is SPIN[23] (SPARQL Inferencing Notation). Accordingly, data constraints can be verified using SPARQL ASK, while SPARQL CONSTRUCT and UPDATE can be used to create new data triples when certain conditions occur. This might be used to perform reasoning tasks as alternative to specific languages (e.g., RIF) and rule engines. New triples might be added as result of RDFS and OWL inferencing, as well as business rules and computations.

SPARQL works effectively in the web of linked data where information changes infrequently and queries produce the same results for a certain time window. However, this is not appropriate for the real world – a source of continuously changing data. Thus a different query language able to capture this dynamism is needed. Several other alternative query languages have been proposed, such as C-SPARQL [25], SPARQLstream [26], EP-SPARQL [27], and CQELS [28]. In the *Linked Stream Middleware* [29] data triples from over 110 000 sensors are collected, and the CQELS language handles both queries on time independent properties and streams of sensor data. The possibility to exploit stream data collected in real environments plays a crucial role in cyber-physical systems such as the smart grid. To this extent the *Super Stream Collider* [30] provides tools to create complex mashups out of linked streams and linked data.

## V. Related work

An architecture for a Semantic Web of Things is proposed in [31]. The authors developed an ontology out of concepts defined in existing ontologies, such as in the SSN-XG ontology. In addition, they implemented the architecture using the Jena[24] semantic web framework. A crawler periodically scans the CoAP network and stores sensor metadata in a centralised triple store. The data can be queried using SPARQL, although sensor data streams are not stored into the triple store and can not be queried. The authors refer to future work for this feature, perhaps exploiting a cloud-based infrastructure. Moreover, semantic entities are proposed as solution to map sensor data to their high-level state. In particular, entities are implemented as virtual sensors that expose a RESTful interface to manage the high-level state. In our proposed architecture, we advocate for the completeness of SPARQL for data management and complex event processing. [32] aims at improving the approach presented in [31]. The authors show that the classic CoAP resource discovery can provide a syntatic matchmaking mechanism. Thus, semantic matchmaking is provided by extending the set of standard CoRE attributes with properties that semantically describe the resource (e.g., latitude and longitude). In our architecture, information describing the resource is defined in the resource description file, rather than as CoRE attributes. As the description consists of RDF statements, it can be stored in the triple store along with the context data, and directly queried using SPARQL.

## VI. Conclusion

This paper outlines a software architecture for integrating homes into the smart grid, with particular focus on data interoperability. We have discussed architectural requirements to be met when dealing with such scenarios and identified potential technologies that could be employed towards this vision. Several companies such as Bidgely[25], EnergyHub[26] and Ecofactor[27] are already proposing complete solutions to collect and analyse energy consumption in households. Data mining and disaggregation techniques (i.e., NILM) are used to provide users with direct and indirect feedback, such as current and predicted costs, as well as information to help and engage them. However, such systems are specifically built for data sources and applications handling energy consumption data. We expect future real world data repositories to be better integrable with other domains data sets (e.g., social networks, weather channels). For instance, the presence of a SPARQL endpoint on each dataset would allow applications to run federated queries. This would facilitate the exploitation of this collective knowledge for producing more accurate data analytics and value-added tailored services. An immediate solution might be to use a cloud-based triple store such as Dydra[28], which exposes data through a SPARQL endpoint and promises flexibility and scalability, as well as basic authentication features. However, we expect further steps towards the standardization of the semantic framework to allow stakeholders the annotation of data using well-defined patterns and vocabularies. In addition, advances in non-intrusive load monitoring will enhance legacy devices with a basic semantic description, which will make disaggregated information accessible within the architecture. In this way, applications can exploit this information without waiting for the market to provide effective development frameworks for producers of devices. In our future research we will try to bridge the gap between producers and consumers of data in the smart grid by developing concrete solutions to facilitate data integration.

[22] http://www.w3.org/TR/sparql11-query/

[23] http://www.w3.org/Submission/spin-overview/

[24] http://jena.apache.org/

[25] http://www.bidgely.com/

[26] http://www.energyhub.com/

[27] http://www.ecofactor.com/

[28] http://dydra.com

## VII. Acknowledgements

This work is supported by Lakeside Labs, Klagenfurt, Austria and funded by the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF 20214 — 23743 — 35470. We would like to thank Kornelia Lienbacher and Lizzie Dawes for proofreading the paper.

## References

[1] S. Darby, "The effectiveness of feedback on energy consumption: a review for DEFRA of the literature on metering, billing and direct displays," Environmental Change Institute, University of Oxford, Tech. Rep., 2006.

[2] Y. A. Strengers, "Designing eco-feedback systems for everyday life," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2135–2144.

[3] S. Pitzek and W. Elmenreich, "Plug-and-play: Bridging the semantic gap between application and transducers," in *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE, 2005, pp. 799–806.

[4] W. Elmenreich and D. Egarter, "Design guidelines for smart appliances," in *Proceedings of the Tenth Workshop on Intelligent Solutions in Embedded Systems (WISES)*, 2012, pp. 76 –82.

[5] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76 –84, 2011.

[6] N. Bui, A. Castellani, P. Casari, and M. Zorzi, "The internet of energy: a web-enabled smart grid system," *IEEE Network*, vol. 26, no. 4, pp. 39–45, 2012.

[7] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[8] K. Kuladinithi, O. Bergmann, T. P. M. Becker, and C. Görg, "Implementation of coap and its application in transport logistics," in *Proceedings of the Workshop on Extending the Internet to Low power and Lossy Networks*, 2011.

[9] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power CoAP for Contiki," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2011)*, 2011.

[10] S. Raza, D. Trabalza, and T. Voigt, "6LoWPAN compressed DTLS for COAP," in *Proceedings of 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE Computer Society, 2012, pp. 287–289.

[11] S. Sakr, "XML compression techniques: A survey and comparison," *Journal of Computer and System Sciences*, vol. 75, no. 5, pp. 303–322, 2009.

[12] R. Kyusakov, "Towards application of service oriented architecture in wireless sensor networks," Ph.D. dissertation, Luleå University of technology, 2012.

[13] R. Verborgh, E. Mannens, and R. Van de Walle, "The rise of the Web for Agents," in *Proceedings of the The First International Conference on Building and Exploring Web Based Environments*, Jan. 2013, pp. 69–74.

[14] R. Verborgh, V. Haerinck, T. Steiner, D. Van Deursen, S. Van Hoecke, J. De Roo, R. Van de Walle, and J. Gabarró Vallés, "Functional composition of sensor Web APIs," in *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, Nov. 2012.

[15] D. Egarter, A. Sobe, and W. Elmenreich, "Evolving non-intrusive load monitoring," in *Proceedings of 16th European Conference on the Applications of Evolutionary Computation*. Springer, 2012, to appear.

[16] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.

[17] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[18] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, no. 0, 2012.

[19] K. Janowicz and M. Compton, "The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology," in *3rd International Workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010)*, 2010.

[20] J.-P. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Semantic sensor data search in a large-scale federated sensor network," in *4th International Workshop on Semantic Sensor Networks. Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[21] Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna, "Semantic information modeling for emerging applications in smart grid," in *Proceedings of the 2012 Ninth International Conference on Information Technology - New Generations*. IEEE Computer Society, 2012, pp. 775–782.

[22] Y. Zhnag, Z. Wei, Y. Yang, and C. Song, "Ontology description of smart home appliance based on semantic web," in *Proceedings of International on Conference Computer Science Service System (CSSS)*, 2012, pp. 695–698.

[23] S. Coppens, R. Verborgh, M. Vander Sande, D. Van Deursen, E. Mannens, and R. Van de Walle, "A truly Read-Write Web for machines as the next-generation Web?" in *Proceedings of the SW2012 workshop: What will the Semantic Web look like 10years from now?*, nov 2012.

[24] M. Rinne, H. Abdullah, S. Törmä, and E. Nuutila, "Processing heterogeneous rdf events with standing sparql update rules," in *OTM Conferences (2)*, ser. Lecture Notes in Computer Science, R. Meersman, H. Panetto, T. S. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz, Eds., vol. 7566. Springer, 2012, pp. 797–806.

[25] D. F. Barbieri, D. Braga, S. Ceri, and M. Grossniklaus, "An execution environment for C-SPARQL queries," in *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 2010, pp. 441–452.

[26] J.-P. Calbimonte, O. Corcho, and A. J. G. Gray, "Enabling ontology-based access to streaming data sources," in *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*. Springer-Verlag, 2010, pp. 96–111.

[27] D. Anicic, P. Fodor, S. Rudolph, and N. Stojanovic, "EP-SPARQL: a unified language for event processing and stream reasoning," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 635–644.

[28] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth, "A native and adaptive approach for unified processing of linked streams and linked data," in *Proceedings of the 10th international conference on The semantic web - Volume Part I*. Springer-Verlag, 2011, pp. 370–388.

[29] D. Le-Phuoc, H. N. M. Quoc, J. X. Parreira, and M. Hauswirth, "The linked sensor middleware - connecting the real world and the semantic web," in *Semantic Web Challenge 2011 of 10th International Conference on Semantic Web (ISWC) 2011*, 2011.

[30] H. N. M. Quoc, M. Serrano, D. L. Phuoc, and M. Hauswirth, "Super stream collider–linked stream mashups for everyone," in *Proceedings of the Semantic Web Challenge co-located with ISWC2012*, Boston, US, November 2012.

[31] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant, and R. Richardson, "Spitfire: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40 –48, nov 2011.

[32] M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, A. Pinto, S. Ieva, and E. Di Sciascio, "A logic-based coap extension for resource discovery in semantic sensor networks," in *5th International Workshop on Semantic Sensor Networks. Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, vol. 904, nov 2012, pp. 17–32.

# Ontologies for Smart Homes and Energy Management: an Implementation-driven Survey

Marco Grassi, Michele Nucci and Francesco Piazza
Università Politecnica delle Marche
Department of Information Engineering
Ancona, Italy
Email: m.grassi — m.nucci — f.piazza@univpm.it
http://www.semedia.dii.univpm.it

***Abstract*—Semantic Web technologies have become a reference technology for information modelling and reasoning support in Smart Homes. This paper provides an extensive review of the ontologies developed in this scenario. Also, it discusses how they can be connected and expanded to create a complete framework that covers all the aspects of a Smart Home, ranging from device description to energy management, under a unifying holistic vision.**

## I. Introduction

Energy conservation represents nowadays a crucial challenge for a world increasingly starved of energy and threatened by pollution and climate changes. Future Smart Homes (SH) can significantly contribute not only integrating small-scale energy generation system and but also properly scheduling tasks to maximise the consumption of locally-generated energy and reduce the energy demand to the Grid during peak hours when energy is more expensive and mostly relies on fossil sources. Intelligent task management requires several heterogeneous information and parameters of a SH, ranging from device features and status to energy consumption and generation, to be managed together. This paper provides an implementation-driven survey on existing ontologies for the SH scenario discussing their possible reuse and extension to create a complete framework that could model all aspects of a SH. The main purpose of this work is to provide a comprehensive vision that could foster the reuse of existing knowledge and serve as background in the development of an ontology framework for the Semantic Smart Home System we are currently working on [1].

## II. Existing Ontologies for Smart Home Related Concepts

In the last decade, Semantic Web technologies have become a reference technology for information encoding and reasoning support in Smart Home scenario to grant the flexibility and extensibility required by such a dynamic and complex scenario. Several efforts have been done to create on purpose ontology frameworks for the SH. In particular, the ThinkHome project [3] and the BONSAI ontology [2] constitute significant reference for this work.

In this section, a review of the most interesting ontologies, classified according to their domain, is provided. In Sec. 3, we discuss how starting from these ontologies it is possible to create a complete framework for Smart Home modelling[1].

### A. Device and Services

Semantic device descriptions play a fundamental role in SH. They indeed constitute the building block for devices interoperability - enabling smarter device search and discovery - and are required to support service composition. Since the development of FIPA[2] and CC/PP[3] ontologies, several efforts have been done to standardise device description. DReggie ontology[4] describes services and devices in DAML language[5]. The CoBrA ontology[5] extends FIPA ontology to enable a basic description of a device hardware[6]. Chuong Wen keeps separated the device profile (location, status, etc.) and the device functionalities modelling them in two distinct ontologies [7]. Togias at al.'s ontology focuses on UPnP device[9]. Bandara et al. introduce specific classes for describing devices' hardware, software and status. Dibowski [8] provides a complex ontology framework for device description that includes building automation domain, device function, platform and manufacturer. They also introduce a device description repository to store and search the device descriptions. The Marine Metadata's ontology[6] focus on oceanographic sensors and bases on several sensor related specifications (e.g. SensorML, NetCDF). CODAMOS[7] ontology introduces the concept of Platform to describe hardware and software capabilities of a device [12]. The DogOnt[8] ontology[10] has been specifically conceived for modelling an intelligent domotic environment describing its devices, their status, functionalities and notifications as well as its architectonic components.

Semantic annotation of the services supplied by devices allows then semantic matchmaking between entities requiring services and entities advertising/offering services. This provides Smart Homes with automatic discovery and composition

[1] Several of the proposed ontologies are not available online and therefore although interesting they could not be reused. For the accessible ontologies the related link is provided.

[2] FIPA Ontology Service Specification http://www.fipa.org/specs/fipa00086/

[3] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0 http://www.w3.org/TR/CCPP-struct-vocab2/

[4] DReggie ontology http://daml.umbc.edu/ontologies/dreggie-ont

[5] http://daml.umbc.edu/ontologies/cobra/0.4/device

[6] MMI Device Ontology http://mmisw.org/ont/mmi/device

[7] http://distrinet.cs.kuleuven.be/projects/CoDAMoS/ontology/context.owl

[8] http://elite.polito.it/files/releases/dog/dogont/DogOnt-1.0.8/dogont.owl

of services. Semantic Web Service description represents by itself a major research topic for the SW community and a great research effort has been done in the recent years[16]. Two main proposals emerged from such an effort: SAWSDL[9] that directly extends WSDL and OWL-S[10] that provides an high-level ontology to describe services. In particular, OWL-S is suitable for the application in the SH scenario and is indeed reused also in CODAMOS and BONSAI ontology for service description. OWL-S provides three main classes: `ServiceProfile` that presents the service, `ServiceModel` that describes how to use the service and `ServiceGrounding` that specifies the details of how an agent can access a service. In particular, `ServiceModel`'s subclass `Process` allows to describe also complex processes specifying the atomic processes they are composed by and the order in which they are executed.

### *B. Context Description: Environment, Users, Location*

Context assumes a broad meaning dealing with a SH that includes environmental conditions as well as information regarding location, users, devices and the energetic status of the system, described more in detail in the next subsection. Context-aware applications, such as energy management systems, can exploit situational information to perform decision making. Context in CONON ontology [11] includes location, user, activity and computational entity. CODAMOS ontology aims to adapt services based on context and models four main concepts: user (role, mood, task to perform, preference profile), platform (device hardware and software that provide services), service and environment (location, time, environmental condition). OntoAmi [13] introduces the concept of event to reflect context changes. ThinkHome ontology framework[11] includes user description (preferences, associated devices, processes to accomplish), environmental conditions as well as current/forecasted weather conditions. BONSAI ontology introduces a set of environmental parameters (temperature, noise, humidity, luminance, etc.) and the location class to describe not only room position but also point in space that could be useful to determine with more accuracy the position of devices or persons in large spaces (e.g. the position of a garden light).

### *C. Energy*

In our vision, to provide intelligent energy management strategies, energy information modelling for a SH needs to cover not only the amount of energy consumed by SH devices but also the energy that can be produced by integrated energy plant [4].

Sesame Demonstrator project [14] proposes the use of ontologies modelling and rule based reasoning to realise energy-efficient smart home that has the ability to interact with the external information and control systems of energy suppliers. Sesame Demonstrator ontology includes concept like Device (consumption per hour, peek power, the switch on/off status), Tariff, Energy Usage Profile, Activity and Energy Profile. Tang et al. [15] propose the use of ontologies and semantic decision tables to support real-time decision about the energetic exchange between Smart Grid and federated Smart Homes. ThinkHome framework allows to describe both the energy produced by an energy plant and consumed by a device according to its state. It also include the concept of Energy Provider, the type of energy produced (green or not) and the Energy Tariff.

## III. The Overall Ontology Framework

At the time of writing, the DogOnt ontology appears as the most complete ontology to model the components of a building and is reused also by ThinkHome project. It provides a wide taxonomical organisation of controllable devices (appliances, home plants and home gateways) together with their functionalities (control, notification and query) and notifications. However, DogOnt doesn't include facilities to describe hardware features of the device that are important for example to determine the quality of offered service. The CoBrA device ontology can be used to provide a minimal description of hardware features like available memory, display resolution, etc. The concept of location is also limited to the `isIn` property that allows only to explicit the position inside a room or a building part and needs to be extend. With such purpose, ThinkHome relies on Building Object Model (BIM) to provide a more accurate description of the environment and the location. BIM model is very powerful, but it not available for most of existing houses and results very complex to use in such cases. With such purpose, similarly to the BONSAI ontology, `Location` concept can be introduced as a superClass of both of a `BuildingEnvironment` and of an exact `Point` in space. CODAMOS ontology can be reused to describe users and their preference profiles. ThinkHome energy ontology provides a solid base to describe energy consumption and production as well as the energy tariff. It could be extended with more detailed descriptions of existing energy plants and forecasted energy production. Finally OWL-S provides a solid backbone to describe services but needs to be extended to take into account service priority and the possibility to delay and schedule its execution as well as the devices that are used by. Fig.1 provides a simplified schema of the resulting overall ontology framework.

## IV. Conclusion and Future Efforts

In this paper, we showed how simply connecting and extending existing ontologies it is already possible to create a complete framework that covers all the facets of a SH. However, this work shall be intended as a starting point for future developments. Some important features are still missing like the possibility to describe device ensembles, to introduce the concept of proximity and to encode different energy saving strategies. Also, the overall device description in the current framework is achieved basing on two large ontologies (CoBrA and DogOnt). We are currently taking into account the development of a lighter-weight ontology with a reduced set of useful concepts.

## References

[1] M. Grassi, M. Nucci, F. Piazza, "Towards a semantically-enabled holistic vision for energy optimisation in smart home environments", Proceedings of the 8th IEEE International Conference Network, Sensing & Control (ICNSC 2011). Delft, The Netherlands, 11-23 April, 2011.

[9] SAWSDL: http://www.w3.org/TR/sawsdl/

[10] OWL-S: http://www.w3.org/Submission/OWL-S/

[11] https://www.auto.tuwien.ac.at/projectsites/thinkhome/ontologies.html

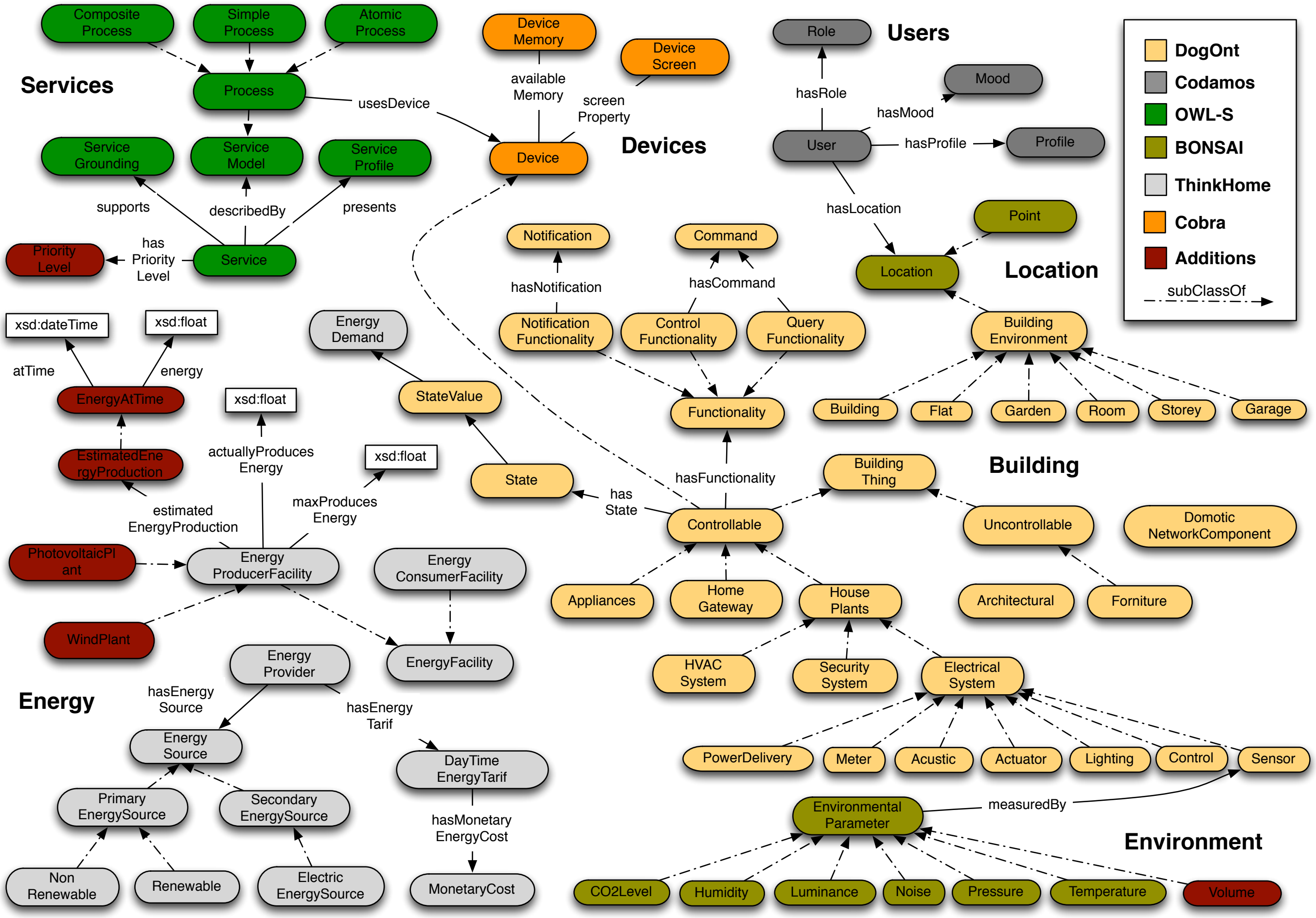


Fig. 1. A simplified schema of the overall ontology framework.

[2] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, “BOnSAI : a Smart Building Ontology for Ambient Intelligence,” in WIMS ’12 Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, 2012.

[3] C. Reinisch, M. Kofler, F. Iglesias, and W. Kastner, “ThinkHome Energy Efficiency in Future Smart Homes,” EURASIP Journal on Embedded Systems, vol. 2011, no. 1, p. 104617, 2011.

[4] M. Grassi, M. Nucci, F. Piazza, ”Towards an ontology framework for intelligent smart home management and energy saving”, Proceedings of the 20th IEEE International Symposium on Industrial Electronics. Gdansk, Poland, 27-30 June, 2011.

[5] D. Chakraborty, “DReggie : Semantic Service Discovery for M-Commerce Applications,” Workshop on Reliable and Secure Applications in Mobile Environment, 20th Symposiom on Reliable Distributed Systems. 2001.

[6] H. Chen and T. Finin, “An Intelligent Broker for Context-Aware Systems,” in Adjunct Proceedings of Ubicomp 2003, Seattle, Washington, USA, October 12-15, 2003.

[7] K. C. Chuong Wen, N. M. Zaini, and E. K. Karuppiah, “Organization of devices information in ontology representation to support service composition,” 2008 International Symposium on Information Technology, pp. 1–8, 2008.

[8] H. Dibowski and K. Kabitzsch, “Ontology-Based Device Descriptions and Device Repository for Building Automation Devices,” EURASIP Journal on Embedded Systems, vol. 2011, pp. 1–17, 2011.

[9] K. Togias, C. Goumopoulos, and A. Kameas, “Ontology-Based Representation of UPnP Devices and Services for Dynamic Context-Aware Ubiquitous Computing Applications,” in 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service, 2010, pp. 220–225.

[10] D. Bonino and F. Corno, “DogOnt - Ontology Modeling for Intelligent Domotic Environments”. 7th International Semantic Web Conference. October 26-30, 2008. Karlsruhe, Germany. Ed. Springer-Verlag, LNCS, pp. 790-803, 2008.

[11] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung, “Ontology based context modeling and reasoning using OWL,” IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, pp. 18–22, 2004.

[12] D. Preuveneers, Y. Berbers, “Automated context-driven composition of pervasive services to alleviate non-functional concerns.” International Journal of Computing and Information Sciences, 3(2), pp. 19–28, 2005.

[13] M.J. Santofimia , F. Moya, F.J. Villanueva , D. Villa, J.C. Lopez, “An agent-based approach towards automatic service composition in ambient intelligence.,” Artificial Intelligence Rev. 29(3– 4), pp. 265–276, 2008.

[14] S. Tomic, A. Fensel, and T. Pellegrini, “SESAME Demonstrator : Ontologies , Services and Policies for Energy Efficiency,” in I-SEMANTICS ’10: Proceedings of the 6th International Conference on Semantic Systems, 2010, pp. 1–4.

[15] Y. Tang and I. G. Ciuciu, “Semantic Decision Support Models for Energy Efficiency in Smart-Metered Homes,” 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Jun. 2012.

[16] K. Mecheri, L. Souici-Meslati, “Semantic interoperability of Web services: A survey,” Machine and Web Intelligence (ICMWI), 2010 International Conference on , vol., no., pp.82-87, 3-5 Oct. 2010

# Simulation and Big Data Challenges in Tuning Building Energy Models

Jibonananda Sanyal, *Member, IEEE,* and Joshua New, *Member, IEEE*

***Abstract*—EnergyPlus is the flagship building energy simulation software used to model whole building energy consumption for residential and commercial establishments. A typical input to the program often has hundreds, sometimes thousands of parameters which are typically tweaked by a buildings expert to "get it right". This process can sometimes take months. "Autotune" is an ongoing research effort employing machine learning techniques to automate the tuning of the input parameters for an EnergyPlus input description of a building. Even with automation, the computational challenge faced to run the tuning simulation ensemble is daunting and requires the use of supercomputers to make it tractable in time. In this paper, we describe the scope of the problem, particularly the technical challenges faced and overcome, and the software infrastructure developed/in development when taking the EnergyPlus engine, which was primarily designed to run on desktops, and scaling it to run on shared memory supercomputers (*Nautilus*) and distributed memory supercomputers (*Frost* and *Titan*). The parametric simulations produce data in the order of tens to a couple of hundred terabytes. We describe the approaches employed to streamline and reduce bottlenecks in the workflow for this data, which is subsequently being made available for the tuning effort as well as made available publicly for open-science.**

***Keywords*—Building energy modeling, parametric ensemble, simulation, big data.***

## I. Introduction

In the United States, buildings consume 39% of all primary energy and about 73% of total electricity of which less than 8% is met by renewable energy resources [1]. To meet the various environmental, social, and financial challenges, the U.S. Department of Energy has set aggressive goals for improving energy efficiency not just in the US buildings sector, but also the industrial and transportation sectors.

Building energy modeling is an important tool for architects and engineers to estimate the energy usage for buildings. The major challenge in designing models for either new buildings or retrofits is to realistically model specific types of buildings and then replicate them across various scales and geographic locations [2, 3, 4, 5]. Each building is unique and unlike the manufacture of consumer goods, there is no single definition of a model that describes the properties for all buildings of a type. As such, it becomes important to be able to accurately model the various physical and material properties of individual buildings to be able to make a realistic assessment of its energy footprint. Having said that, DOE provides a set of standard reference building models [6] that are used nationwide and are representative of the U.S. building stock. These building models are used for normative analysis to determine how policy changes would affect energy consumption in the US, determine tax trade-offs, design building codes, trade-off incentives, and evaluation of the effect of climate change on buildings.

While the simulation engines are not perfect, the outcome of such analyses depends a lot on the correct representation of the physical building in the model. The typical order of business is to employ a building energy modeling expert who painstakingly tunes the various parameters manually. The outcome of the tuning exercise is always a function of the experience and expertise of the modeler. The tuning exercise is very time consuming and sometimes has been observed to take months to "get right".

"Autotune EnergyPlus" [7] is an ongoing research project that aims to employ computers to automatically calibrate the various input model parameters for building retrofit packages (figure 1). In this paper, we describe the Autotune approach and focus on elaborating the technical challenges faced and overcome. Specifically, we focus on elaborating the necessity, experience, and lessons learned using different architectures of supercomputing systems, and the management of large amounts of generated simulation data.

The rest of the paper is divided into the following parts: we begin with describing the Autotune methodology, followed by a description and elaboration of the computational challenge. We follow this by a description of the big-data challenges such as the amount of data generated, its management, storage, and analysis. This is followed by a description of the simulation workflow and the evolving overarching software architecture.

## II. Autotune Methodology

There are about 20 major software tools that perform building energy modeling with each having certain strengths and weaknesses. The primary whole building simulation engine supported by the U.S. Department of Energy is EnergyPlus (E+), which is roughly 600,000 lines of FORTRAN code. A typical input to the simulation engine can contain over 3,000 input parameters for a regular residential building which must be tuned to reflect the building properties. Good sources of data or handy reference values are not easily available for many of these parameters. Experiments by [cite Dewitt] has established that there are often mismatches between product specifications as per the ASHRAE handbook and the manufacturer's supplied data. Again, a finished building often deviates from its design and as a result, the model of the building no longer matches that of the real building. For existing buildings, retrofit

Jibonananda Sanyal and Joshua New are with the Building Technologies Research and Integration Center, Oak Ridge National Laboratory, Oak Ridge, TN, 37830 USA e-mail: {sanyalj, newjr}@ornl.gov

packages are often implemented leading to substantial energy savings. In most cases, there is significant return on investment within a reasonable break-even period for retrofits in existing structures.

All these reasons compel us to adjust a building energy model to match measured data, the tweaking of which is typically performed manually by a building modeling expert. This is a painful, time-consuming, and an error-prone process. The manual tuning approach is also highly subjective and non-repeatable. A very large number of building models start out with the DOE reference buildings (which are most representative of the U.S. building stock) and go through the manual adjustment of geometry, HVAC properties, insulation, fenestration, infiltration properties, etc. Because of the notoriety of the process, energy modeling is expensive and is done primarily on large projects. Unless there are savings in the assessment, professionals in the field will not adopt it as a viable operational practice.

The goal of the "Autotune" project [7] is to save the time and effort spent by energy modelers in adjusting simulation input parameters to match the measured data by providing an easy button (figure 1). To achieve this objective, an initial model of the building and sensor data for a time period is provided to the Autotune software stack, which then spins off the trained machine learning agents [8] and returns a tuned model of the building. At the heart of the Autotune capability is a set of multi-objective machine learning algorithms [8] that characterize the effect of individual variable perturbations on EnergyPlus simulations and adapts the given model to match its output to the supplied sensor data (figure 2). Once machine learning agents are tuned and available, the computational cost of tuning a typical user's building model will be reduced to matter of a few hours using widely available desktop computational resources. This paper focuses on the supercomputing and big-data challenges. Readers interested in the machine language algorithms are referred to [8].

The system is currently in a late stage of development and is being demonstrated to match a subset of 250 sensors of 15-minute resolution data in a heavily instrumented residential building in addition to DOE's standard reference building models [6] for a medium sized office, a warehouse, and a stand-alone retail building. Further, the simulations comprise of three vintages (old, new, and recent) of the DOE commercial reference buildings across 16 different cities representing the different ASHRAE climate zones and sub-zones.

## III. Computational Challenge

The computational space for performing parametric analysis is combinatorially large. Brute-force exploration for 156 of 3000+ input parameters using only minimum, average, and maximum would require $5 \times 10^{52}$ EnergyPlus simulations and $2 \times 10^{28}$ lifetimes of the known universe when running on *Titan*. . . for a single building! For this reason, several numerical algorithms (discussed below) are used which allow convergence to a solution without brute-forcing every combination of inputs and intelligent experimental design can guide parametric analysis to a much smaller sample size.

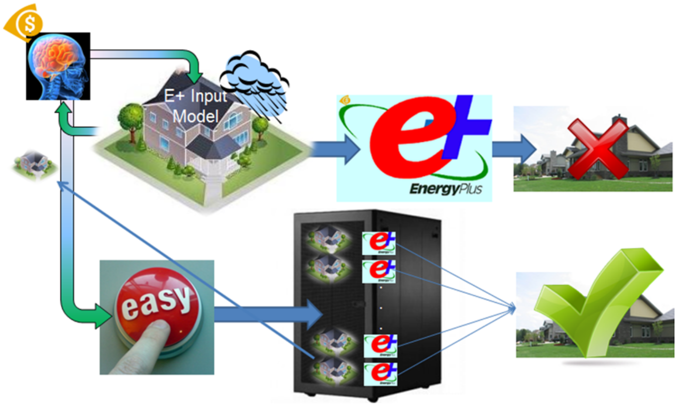


Fig. 1. Autotune workflow for E+ building energy models as a cost-effective solution for generating accurate input models. The conceptual figure illustrates how a conventional calibration process is expensive and inaccurate. The 'easy' button illustrates the 'Autotune' pathway for calibration using computational methods.

### A. Parametric Sampling and Input Generation

We worked with building technology experts to pick only the most important parameters, thereby reducing the count to 156 for a residential building. These are parameters that are most commonly used by energy modelers. The parameters were further ranked into three importance categories. Domain experts further defined realistic bounds and incremental step size values for the parameters. In addition, various meta-parameters were determined which allow several individual parameters to be varied as a function of a single input parameter.

Even with ~156 input parameters and three levels of incremental values for each of the simulations, we are looking at 10 million simulations. Each individual simulation takes roughly 8 minutes which translates to 2 million compute hours accounting for overheads. Using Oak Ridge National Laboratory's *Titan* supercomputer (currently ranked as the fastest supercomputer in the world), this would take 2 months of calendar time to just run the simulations, let alone manage the data, perform machine learning, and subsequent analysis. Effective, scalable methods to sample the input space is crucial.

We use the expert's grouping of important parameters to divide the sampling space into groups of relative importance. We have also used low-order Markov ordered simulations to determine variables with a monotonic effect on sensor data that can reliably be interpolated to estimate impact of a given input. The source of variance of individual variables is being used to guide sampling rates of the more sensitive inputs. Finally, experts in multi-parameter optimization will be investigating computational steering algorithms to determine the optimal sampling strategy for the remaining space beyond the brute-force sampling of higher order Markov chains of Monte Carlo simulations.

Each simulation requires its own input file. The individual input files are generated using two strategies:

*1) A program in Perl:* We wrote a program in Perl that reads a template input file with 156 special markers for variable

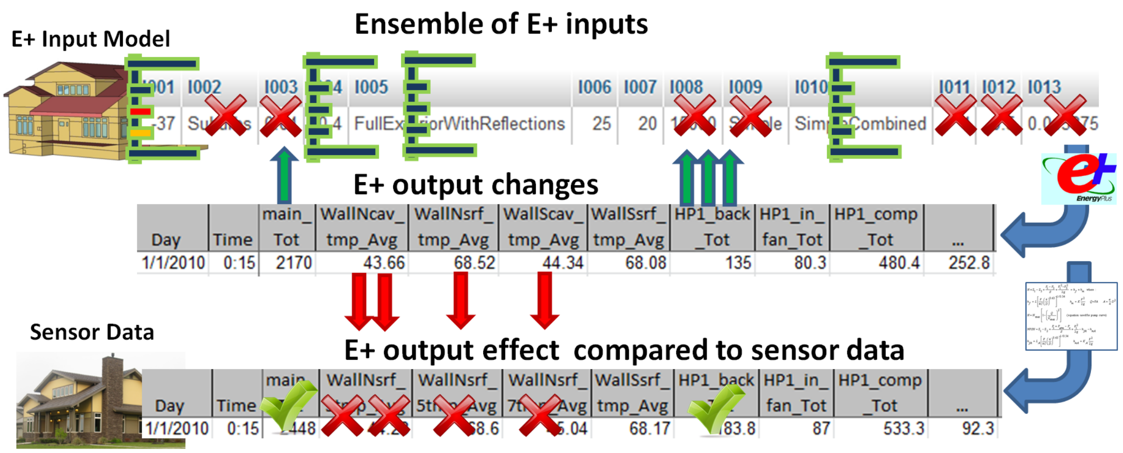

Ensemble of E+ inputs
E+ Input Model
I001 I002 I003 I004 I005 I006 I007 I008 I009 I010 I011 I012 I013
-37 FullExteriorWithReflections 25 20 Simple SimpleCombined 0.0 75
E+ output changes
Day Time main_Tot WallNcav_tmp_Avg WallNsrf_tmp_Avg WallScav_tmp_Avg WallSsrf_tmp_Avg HP1_back_Tot HP1_in_fan_Tot HP1_comp_Tot ...
1/1/2010 0:15 2170 43.66 68.52 44.34 68.08 135 80.3 480.4 252.8
EnergyPlus
Sensor Data
E+ output effect compared to sensor data
Day Time main WallNsrf_ WallNsrf_5th Avg WallNsrf_7th Avg WallSsrf_tmp_Avg HP1_back HP1_in_fan_Tot HP1_comp_Tot ...
1/1/2010 0:15 448 8.6 5.04 68.17 83.8 87 533.3 92.3

Fig. 2. A virtual building model (top) in software space and a real building (bottom) having sensors, when viewed as vectors of numbers, allows a mathematical mapping between vector spaces for direct comparison between sensed world state and sensor space. The layer in the middle illustrates the output of EnergyPlus as a vector of numbers and how output is affected by the ensemble of parametrc input distributions (green bars). The red and green arrows illustrate parameters being tuned down or up during the tuning process. The crosses and checks illustrate progressive acceptability of tuning as the machine learning progresses.

values and another file that specifies value ranges and step sizes. Different sampling algorithms such as maximum and minimum bounds check, Markov Order 1, or Markov Ordering 2 were programmed as modules. When invoked, they would sample the input space and fill in values for the special markers in the template and write it out as individual input files.

*2) E+ supplied parametric preprocessor:* The EnergyPlus simulation tool comes with a parametric preprocessor program which takes an input file embedded with parameters and generates the individual files. The parametric values are supplied within the file where one must list all the possible values, with repetition, for nested block designs. The algorithm scan one line at a time from each set of parameters to generate the individual input files. For example, if we had 4 variables with each having 3 levels, we would have $4 \times 3 = 12$ lines of values for each variable adding a total of $12 \times 4 = 48$ additional lines of input. While convenient for small sampling spaces, it becomes cumbersome very fast. We used this approach initially for the residential building but quickly switched to using our own Perl program. For the DOE reference buildings, we used the parametric preprocessor mainly because we were using meta-parameters to define other parameters and each building type was replicated for three vintages (old, new, and recent) across 16 locations in the United States, which made the management of bulky parametric descriptions somewhat easier.

In summary, a total of 8 million simulations are planned for following building types:

- Residential: ~5 million
- Medium office: ~1 million
- Stand-alone retail: ~1 million
- Warehouse: ~1 million

A set of ~1 million input files (1M each for retail, warehouse, and medium-office buildings) were generated on a 16-core shared-memory compute server using four parallel processes which took about 18 hours to complete.

### B. *Supercomputing*

In the initial stage of the effort, a Qt based tool named “EplusGenius” was developed that leveraged the power of idle desktops to run EnergyPlus simulations and upload the results to a database. It was engineered to provide the choice to run only after office hours and on weekends. While effective in harnessing unused computational power from a network of computers, the computational requirements to achieve a reasonable time-to-solution necessitate the use of HPC resources. Time on several supercomputing systems have been competitively awarded and used to demonstrate the scalability of our algorithms and code for the massively parallel leadership-class computing systems. Systems include the 1024-core shared memory *Nautilus*, 2048-core *Frost*, and 299,008-core *Titan* which is currently the world’s fastest supercomputer at 20 petaflops. The *Nautilus* machine has 4 TB of global shared memory visible to every processor on the system. *Titan* (and *Frost* too) has a distributed memory model with each node having 16 processors and 32 GB of RAM.

While in theory this is an embarrassingly parallel problem (parametric ensemble) and should be easy to parallelize, various complicating factors make this difficult to scale in practice. First, EnergyPlus was developed as a desktop application and was not supercomputer ready. In a typical execution trace for a single simulation, a sub-directory and a large number of files (12+ files amounting to 100+MB) are created. Second, constant moving and soft-linking of the files are done as the simulation workflow executes. Third, an annual simulation with 15-minute output of 82 channels is 35MB in size and currently needs to be stored on disk for later analysis. In other words, the entire process is particularly I/O intensive, which complicates the scalability of parallel execution on supercomputers. We attempt to mitigate these issues in many ways.

A branch of the source code for the engine has been made open-source; however, writing a wrapper for over 600,000 lines of code to streamline I/O for use on supercomputers is outside the scope of this work. We treat E+ as a black-box and use it to simulate our 8 million runs. In addition, the workflow depends on a number of additional executables, the source-code of which is not available.

The current job submission scripts were designed to pick up a set of input files and execute them in parallel on the number of cores requested and moving the output to a directory when complete. Initial experiments on the supercomputing systems delivered very poor scaling performance which were expectedly traced to the bandwidth and Lustre file-system saturation with the frequent number of large I/O requests. Most supercomputing systems use a parallel file-system named Lustre. Software communicates with the file-system through the metadata server which instructs the hardware to access the right set of disks. The file-system is typically located in its own server rack, separate from the computation nodes and connected via a high-speed, high-bandwidth communication network. The biggest bottleneck was determined to be the communication with the Lustre metadata server and the storage targets.

In order to alleviate the filesystem bottleneck, we made use of the memory-based virtual file-system which gave us more than two orders of magnitude improvement over using the Lustre filesystem. In addition, we block-partitioned and streamlined our input and output mechanisms. To outline the steps performed:

1) EnergyPlus comes with a large number of supporting executable programs and associated files. A typical E+ simulation is essentially a workflow where multiple executables are invoked with each producing temporary files ingested by subsequent programs. We minimized the engine's folder structure to include only the binaries and libraries required for our simulations, modified the execution scripts to use relative paths, and compressed the minimized file structure to make it ready to be loaded into the virtual filesytem.
2) In an effort to reduce the number of input files fetched, we performed a pre-processing step in which we grouped the inputs into blocks of 64 simulations each and packed them into compressed tarballs. This reduces the number of files fetched by a factor of 64 and reduces size by ~60%.
3) For *Nautilus*, individual jobs can be placed on individual processors using the 'dplace' command. A heavily modified job submission script allows us to request a specific number of cores and provide a count of the number of batches to run. For example, a request for 256 cores with 90 batches would start out by picking out $256/64 = 4$ blocks of compressed input files and the simulation engine, and then parallelly extract them to the virtual file-system. Each core then executes a simulation (using an explicit 'dplace' command which runs a job on a core). After completion, the data is moved to the physical file-system and the next batch of 4 compressed files is loaded. This is repeated 90 times.
4) For *Frost* and *Titan*, the virtual file system (*tmpfs*) is shared-memory visible within a node. Additionally, the systems do not use 'dplace' for direct placement of individual jobs on a processor. We wrote a message passing interface (MPI) program that would make each node load the engine and a block of 64 runs into its shared-memory. Since each node has 16 processors and there are 64 files in a compressed block of inputs, the node makes four iterations to run all 64 simulations.
5) Once a block of simulations is complete, the output files are added to a tarball and moved to disk. This is typically about 1.5 – 2.4 GB in size depending on the type of simulation. This also reduces the number of I/O interactions with the Lustre filesystem by a factor of 64. The virtual file system is cleaned and prepared for the next block to run. We experimented with compression of the output in memory and its transfer to disk but found that the idling of other processors was unacceptable while a subset of the processors performed the compression.

We have been able to complete a batch of 1 million E+ simulations for the warehouse building using *Nautilus* and *Frost* in under 2 weeks of continual execution (along with other users running other jobs on the systems). The theoretical runtime using average job execution metrics was estimated at about 7.6 days for the batch of 1 million simulations. We were also able to complete the 1 million Standalone Retail simulations in ~2 weeks.

Using *Titan*, we have been able to achieve excellent scalability completing ~250,000 EnergyPlus simulations using 65,536 processors in under 45 minutes, and expect our code to scale to be able to use all 299,008 cores.

## IV. Big data challenges

The output from these simulations produce a file between 35 MB to 70 MB each constituting anywhere from 80 to a 180 output variables at 15 minute intervals for a whole year. With 8 million simulations underway, a number of challenges emerge.

### *A. Data storage*

We are looking at approximately 270 TB of raw data when all simulations are complete. We have estimated that this can be compressed down to about 70 TB, which is still a large amount of data. This is the size of simulation data prior to any operations or storage performed as part of the analysis processes. There are certain logical partitions in the data such as type of building simulation, its vintage, location, and also the parameter sampling and grouping strategy which helps us in breaking down the data management space. While many database-related technologies have been tested and explored, effectively storing this data for quick retrieval and analysis remains a challenge.

We have explored a number of databases including MySQL, noSQL/key-value pair, columnar, and time-series database formats for simulation/sensor data and currently implement a hybrid solution with a part of the summarized data entered in a database and readily accessible for querying and analysis while and the raw data being fetched on demand. This data is currently provided with no guarantees since the entire data queryable with an assured turnaround time (a solution similar to a hadoop stack) for queries is currently infeasible.

We currently have an 80 TB Synology disk station which primarily serves as a data repository with fast access to individual files and are in the process of acquiring additional hardware for some of the analysis needs. Although additional hardware is being procured, it is geared towards adding compute capabilities for analysis but is not geared towards assured turn-around times for queries.

### B. Data transfer

There is a learning process in planning for such large data efforts. One of the things we learned is the cost of transferring data from the supercomputing infrastructure to our data repository. Although we have a 1 Gbps Ethernet backbone, initial data transfer speeds using ftp between computers that housed the output data and the data repository ranged from 7 MB/s to 12 MB/s indicating that it would take over three months to transfer all data for just the warehouse simulations. This was a multi-hop transfer because of various firewall rules within the laboratory. We overcame the firewall rules and were able to sustain 40 MB/s peer-to-peer transfer rates.

Most large supercomputing facilities have dedicated data transfer mechanisms such as *GridFTP* or *BBCP*, which are tools that optimize and parallelize data transfer. We are currently using GridFTP and are moving data at consistent speeds of around 120 MB/s. This works out to about 4 days for transferring the output of 1 million warehouse simulations which is a huge improvement over the initial estimate of over three months.

### C. Data analysis

Many domain experts and students are interested in various ways to analyze this valuable dataset. Moving the data for post-processing calculations is very expensive. The trend these days when working with big-data is to move the computation to the data. Map-reduce and distributed computing methods apply the same philosophy. When dealing with simulation data, especially as large and individually discrete as our simulations are, an added step of post-processing to calculate summarizing statistics is very beneficial. We have architecture our code to not only output data at 15 minute timestep intervals but also calculate hourly, daily, and monthly averages.

We are also working on engineering in-line processing of the data across simulations to calculate different kinds of metrics including those that can be used for sensitivity analysis. Typically, a batch of simulations is run and calculation of different metrics across simulations while the data is still in memory will help us save on expensive IO operations. In addition, techniques to coalesce calculated statistics across multiple batches to reflect the state of the entire class of simulations is also underway.

## V. Evolving system architecture

Figure 3 illustrates the software architecture of the system. Various parts of the system architecture are still evolving since many of the supercomputing and big-data challenges require us to revisit standard approaches to data management. We are working towards a scalable software framework that allows one to design massive building energy simulation experiments and allows different software agents to access the data and perform analysis. In addition to calculating summary statistics in-stream, we are also adding the capability of running machine learning agents on the data as they are generated.

The final product of the effort is envisioned to be a variant on software as a service. For Autotune's non-HPC side, users of the system will be able to visit a web-site and submit an autotuning job. Since the DOE reference buildings are used as a starting point for many users, we expect to provide these as templates. Techniques have been identified that will help provide customization features that will help users tweak these to more closely represent their buildings. The system will perform all the necessary computation and machine learning and come back with an answer that best fits the objective of the tuning. For Autotune's HPC side, advanced optimization techniques for mining large data will be ready to analyze new problems across domains. Execution traces and provenance of models and parameters previously tuned will be provided to analyze and identify the critical components in typical tuning processes. The Autotune process by itself is domain agnostic. The technology can be easily customized to tune smaller models and configured to run on single desktops, network of workstations, shared-memory supercomputers, distributed-memory supercomputers, as well as deployed on the cloud.

In addition to facilitating research activities and Autotune user access, the AutotuneDB approach of providing the entire simulation data for full public access is anticipated to be increasingly utilized by web services, researchers, companies/entrepreneurs, and home-owners to mine for their specific question of interest. The silent availability of this database has already triggered several interested parties (primarily involving researchers and future doctoral candidates) and dove-tails with the growing trend of increased transparency in government programs, as evidenced by popular White House initiatives such as GreenButton (for utility data) and many other data-sharing initiatives related to building performance (energy reporting now required in some cities) and high performance building databases.

## VI. Conclusion

The successful completion of the presented Autotune effort underway at the Oak Ridge National Laboratory will go a long way in alleviating the tedious task of tuning a building energy model to sensor data. The employment of machine learning agents performs a multi-objective optimization (see [8]) of the input parameters to provide a solution that best matches the input sensor data. The refinement and dimension reduction of the input parameter space to 156 important ones identified by the experts helps to reduce the computational space. Further, various methods to scientifically sample the input parameter space helps to reduce the computational space.

One of the questions asked is how the Autotune workflow allows faulty operation detection in its workflow. There is a large

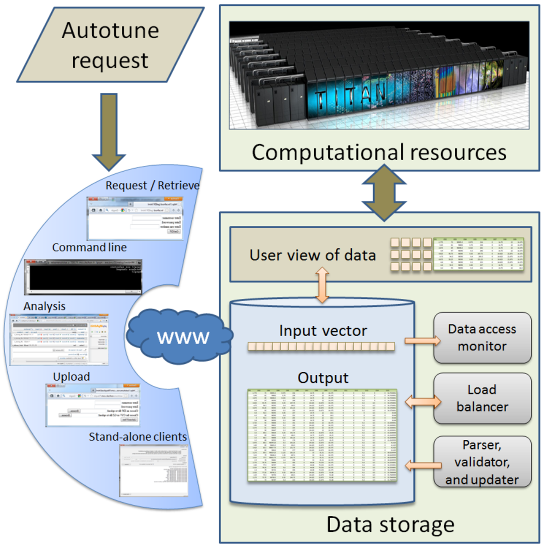


Fig. 3. Architecture of the Autotune software-as-a-service product.

body of work for Fault Detection and Diagnostics (FDD) with fault-injection and detection of equipment and whole-building systems currently being conducted that is outside the scope of this work. However, Autotune's latest unpublished work has been in automatic calibration of equipment schedules (in addition to properties) which should capture (and thus facilitate correcting) faulty/offline equipment, half-open dampers, and other faulty operations provided there is relevant sensor data on which to calibrate. Therefore, the normal Autotune workflow is expected (though hasn't been fully verified in ongoing work) to capture faulty and fixed operations throughout the span of time for which relevant sensor data is available. Since any time period can be handled, two models could be calibrated through two Autotune workflows, one for "faulty operation" and one for "proper operation" to allow energy-saving (or other) comparisons between them.

The main objective of the paper is to highlight the technical challenges faced in simulating a very large set of simulations using an engine that has been developed for desktop applications, and in the process, generating a large amount of data. We expect that lessons learned and software developed will be useful for researchers who intend to run large ensembles and perform data mining of very large data. use a similar engine and plan to run a large number of parametric simulations. We also hope that some of the challenges faced and lessons learned in analyzing, moving, and managing large data across hardware platforms will provide beneficial insight and help to researchers in planning such endeavours.

Lastly, we also expect that the open availability of the parametric simulation datasets for the standard DOE reference buildings will directly benefit the building sciences community.

## Acknowledgement

This work was funded by field work proposal CEBT105 under the Department of Energy Building Technology Activity Number BT0201000. We would like to thank Amir Roth for his support and review of this project. We would like to thank our collaborators which include Mr. Richard Edwards and Dr. Lynne Parker from The University of Tennessee, Dr. Aaron Garrett from Jacksonville State University, and Mr. Buzz Karpay from Karpay Associates. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Our work has been enabled and supported by data analysis and visualization experts, with special thanks to Pragnesh Patel, at the NSF funded RDAV (Remote Data Analysis and Visualization) Center of the University of Tennessee, Knoxville (NSF grant no. ARRA-NSF-OCI-0906324 and NSF-OCI-1136246).

Oak Ridge National Laboratory is managed by UTBattelle, LLC, for the U.S. Dept. of Energy under contract DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under Contract Number DEAC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

## References

[1] U.S. Dept. of Energy, *Building Energy Data Book*. D&R International, Ltd., 2010. [Online]. Available: http://buildingsdatabook.eren.doe.gov/docs%5CDataBooks%5C2008_BEDB_Updated.pdf

[2] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg *et al.*, "US Department of Energy Commercial Reference Building Models of the National Building Stock," 2011.

[3] R. Briggs, R. Lucas, and Z. Taylor, "Climate classification for building energy codes and standards: Part 1–development process," *ASHRAE Transactions*, vol. 109, no. 1, pp. 109–121, 2003.

[4] ——, "Climate classification for building energy codes and standards: Part 2–zone definitions, maps, and comparisons," *ASHRAE Transactions*, vol. 109, no. 1, pp. 122–130, 2003.

[5] IECC 2009 and ASHRAE 90.1-2007, *Energy Code Climate Zones*. [Online]. Available: http://resourcecenter.pnl.gov/cocoon/morf/ResourceCenter/article/1420

[6] K. Field, M. Deru, and D. Studer, "Using DOE Commercial Reference Buildings for Simulation Studies," 2010.

[7] J. R. New, J. Sanyal, M. S. Bhandari, and S. S. Shrestha, "Autotune e+ building energy models," *5th National SimBuild of IBPSA-USA*, 2012.

[8] R. E. Edwards, J. New, and L. E. Parker, "Predicting future hourly residential electrical consumption: A machine learning case study," *Energy and Buildings*, 2012.

# A Hierarchical Networked Micro-Simulator to Study Grid-Integration of Renewables and Electric Vehicles

**Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp**
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24061, USA
E-mail: {soumyo,shukla, jsthorp}@vt.edu

*Abstract*— **The effect of integrating intermittent renewable generation such as wind and solar, as well as plug-in electric vehicles (PEVs) on a grid is an important area of study. Renewable generation depends on weather. Energy consumption, storage, and emergency usage of battery-stored power in PEVs are dependent on the spread of such vehicles in a geographical area, commute patterns, and hours of long-term parking. These are stochastic in nature. We have developed a hierarchical networked micro-simulation environment to characterize their effect on the grid's load-carrying capacity, reliability of unit commitment and planning, and boundaries of grid safety, etc. We have used this micro-simulation environment for a number of studies based on 4-year real data from New York City's weather and load profiles, projected PEV population, and current commute profiles. In this paper, we describe our micro-simulator's architecture, and its ability to scale various abstraction levels depending on the accuracy needed, study objective, and computational time and resources.**

***Keywords— grid integration; load-carrying capacity; micro-simulator; plug-in electric vehicles; renewable energy***

## I. Introduction

Micro-simulators are often used in econometric and micro-economic studies where behaviors of individuals (e.g., persons, house-holds, industrial units etc.) are modeled, and their interactions are captured in the model so that during simulation various quantitative measures of the aggregate behavior of a community can be studied. Often stochastic modeling is used since the individual's behaviors could have uncertainty, and the macro measures that are developed from such simulations are often characterized with probabilities.

Our study started with predicting the reliability of renewable generation in a specific power grid, given their dependence on stochastic processes such as weather [1]. Since weather modeling and simulation is extremely compute intensive, we instead used the predicted weather as an input parameter of our study, and considered the past weather data against real weather data for the same time periods, and computed the reliability of weather prediction data to characterize the reliability of planned load servicing ability of a grid that has a certain penetration level of renewable energy such as wind and solar energy. However, in the initial studies we assumed that the renewable energy is not stored for future usage as the battery technology is one of the major bottlenecks for such storage-based integration of renewables. In our later studies [2-5], we introduced storage in the form of batteries in plug-in-electric vehicles (PEVs) which can be charged during their resting periods (during night, office hour parking, etc.), and can be relied on during peak hours to release their excess energy to grid via vehicle-to-grid (V2G) operation to mitigate the possible mismatch of load and generation in the event of weather mis-prediction or weather being non-conducive to renewable generation as per planning.

While our first study could be done with a combination of analysis of past weather data and prediction accuracy computed by comparing realized vs. predicted weather parameters, real weather data, and predicted renewable penetration, using statistical methods, the introduction of PEVs brought upon the challenge of modeling classes of vehicles, their commute patterns, parking distribution, etc. We therefore extended our statistical analysis tool to a hierarchical networked micro-simulation environment which can be further extended for many different studies of the effect of renewables on power planning for a specific grid.

Our micro-simulator is based on actor based modeling (a la Ptolemy II [6]) where one can model individual generators (both fossil fuel and renewable) fed with real weather data, and other parameters, and stochastic parameters for varying their behaviors, individual PEVs, their commute, and parking behaviors, their energy consumption, storage capabilities, and their ability to supply back stored energy to the grid. However, we decided that modeling millions of vehicles, or hundreds of renewable generators in future micro-grids may not be feasible from computational resource perspective, and hence we designed the micro-simulator in such a way that one can build individual actors that represent aggregate behaviors of a class of generators, or a class of PEVs having similar behaviors etc. This allows us to study in a computationally feasible way a lot of macro-level parameters such as reliability of renewables as a function of weather present in the geographical region of the grid under study, load carrying capacity , limits of safe operation of a given grid as renewable and PEV penetrations are increased to very high levels, etc. We have had access to the hourly load and solar irradiance data for New York City over 2007-2010 to feed to our simulator and study figures relevant to New York City.

In [2-5] we have studied large scale PEV penetration and their impact on the grid's reliability and load carrying capacity, impact of increasing installed solar and wind generation capacity, combined effect of intermittent weather-dependent renewable energy sources and fleets of PEVs on a grid's reliability, imbalance reserve requirements and safe operations, etc . using this infrastructure. In this paper, however, we only describe our micro-simulation environment, and its capabilities.

## II. Related Work

Micro-simulators are a very important tool in the arsenal of economists to study economic phenomena, health policy makers to study the spread of diseases, traffic planners to study traffic, and effect of various traffic related road improvements, etc. There are a number of commercial and research tools available for micro-simulation, some of which are domain specific. For example, Pensim2 [7] is a pension modeling and simulation tool used in UK, EUROMOD [8-9] is an econometric simulation tool for 15 European countries, PECAS [10] is for urban planning in the US, SimTraffic [11] is for micro-simulation of traffic patterns, CISNET [12-13] is for health science micro-simulation used by US National Cancer Institute, etc. Many of these micro-simulators use scripting languages such as Python to make it easier to use by non-computer scientists (mostly social and economic scientists are users of these tools). Our micro-simulation tool is written in C++, and based on object oriented concepts that allows us to create a hierarchy of individual simulators, which can vary from expressing behaviors of individual entities (such as single PEV, or single generator) or aggregate behaviors of a class of entities (one class for a set of PEVs with the same electric parameters or same commute patterns), or aggregate of all entities of a certain type (aggregate behavior of all PEVs).

As can be easily conceived, such variation in level of detail of the behaviors (full individual behavior to aggregate behaviors) would lead to different accuracy levels of the macro-level parameters computed. Also, some studies require only aggregate behaviors of classes of entities, rather than individuals because the study being carried out may not require individual level behavior simulation or there may not be enough available data to model behaviors of each individual. The stochastic attributes of the behaviors often account for the uncertainty or behaviors as well as the lack of detailed behavior.

Notwithstanding the fact that more accurate the simulation model is, better the study results, it is often imperative to have a quick and approximate simulation for policy makers to make decisions, and as long as the increase in accuracy does not negate the findings of less accurate models (which can happen in certain cases), it is important to be able to scale various abstraction levels depending on the availability of time and other computational resources, data availability, and the objectives of the study. We have not found any of the above mentioned tools to be directly usable for our purpose. We could have used some of the general modeling and simulation frameworks such as Mathematica [14] or MATLAB [15] or even Ptolemy II [6], but the problem is that in the first two we have to create some design patterns to fit our need of hierarchical abstraction levels, and in Ptolemy II we would have to possibly invent a new domain or use one of the already existing domains – without putting any domain specific constraints. Instead we chose to develop our own micro-simulation framework.

## III. Architecture of the Hierarchical Networked Simulator

We develop an approach of utilizing a flexible and hierarchical network of micro-simulators and demonstrate its applicability and efficacy in a range of applications in the areas of grid integration of renewable energy, load carrying capacity impact of PEVs, determining optimal capacity of renewable generators, and identifying boundaries of safe operations of a power system in the presence of a high penetration of renewable energy sources as well as PEVs. Each node in our simulation network is either a conductor node (C), a simulator node (S) or an algorithm node (A). Each simulator node is a plug-in micro-simulator that can model a variety of systems and processes in an energy system, weather parameters, commute behavior, etc. It generally takes data sets and modeling parameters as inputs. Each algorithm node implements a specific algorithm, e.g. V2G optimization algorithm. Each conductor node has the supervisory responsibility of coordinating across simulator and algorithm nodes to simulate an end-to-end scenario, or determine points of optimality in a problem space, or find boundaries of the viable operating space. A conductor node can be composed by combining other conductor nodes.

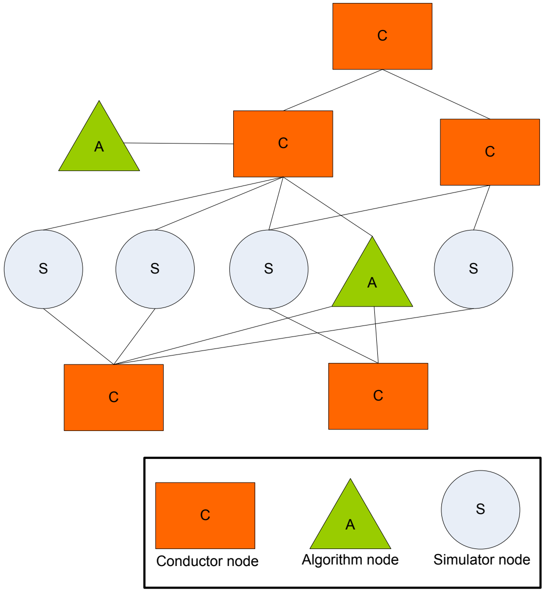


Figure 1: Simulator network with three different types of nodes

A simulator network can be represented as N where N = (V, E). V represents the set of nodes and E represents the set of edges.

V = {C1, C2, …, Cn, A1, A2, …,Ap, S1, S2, …,Sq} where n=number of conductor nodes, p=number of algorithm nodes, and q=number of simulator nodes.
E = {($v_i$, $v_j$) | $v_i$ is the i$^{th}$ node and $v_j$ is the j$^{th}$ node in V; $v_i$ E {C1, C2, …,Cn}, $v_j$ E V}

Each simulator node is typically implemented as a class. Multiple objects of the class are instantiated at run time depending on how many individual entities of the class need to be simulated. Level of abstraction of the simulation can be varied by controlling the number of objects of a class with higher number of objects representing higher level of granularity.

Algorithm nodes are implemented as a function that can be called by a conductor node to control the behavior of simulator objects. Algorithm nodes are typically stateless. Depending on the state of individual simulator objects or a group of simulator objects a conductor node may invoke an algorithm node and pass it the necessary context to determine any control action necessary.

Conductor nodes represent the coordination logic typically implemented as a function. A conductor function may launch one or more simulator objects, call one or more algorithm functions from time to time depending on simulator states to determine control actions, and perform computations on simulator object variables to produce overall outputs from the simulation exercise. A conductor function can call other conductor functions if the conductor node is composed by combining other conductor nodes.

## IV. Case Studies

Our hierarchical networked simulator approach can be best described with a few of the concrete applications where the simulator has been successfully employed.

### *A. Analyzing the load-carrying-capacity of PEVs and impact on solar generators*

The goals of this simulation exercise [2] are to:

1. Determine failure behavior of existing fossil-fuel generators in the grid
2. Simulate the behavior of existing fossil-fuel generators in the grid
3. Simulate the aggregate load in the grid
4. Simulate the commuting behavior of PEVs in the region
5. Employ a vehicle-to-grid (V2G) optimization algorithm that controls the charging (grid-to-vehicle, i.e. G2V) and discharging (V2G) of the PEVs
6. Simulate the power output of solar PV generators based on relevant weather parameters
7. Determine load carrying capacity - measured by effective load-carrying capacity (ELCC) - contributed by PEVs
8. Determine load carrying capacity contributed by solar generators
9. Determine combined ELCC contributed by PEVs and solar generators when both are present

We have the following relevant variables:

**L** - pattern of aggregate load on the grid over a time period T
**s** - aggregate peak supply capacity
**r** – model-based hourly failure rate of existing individual generating units
**lole -** reliability of the grid, represented by the loss of load expectation (LOLE)
**B** - effective operating range of the state of charge of a PEV's battery
**c –** average length of commute in miles
**m -** mileage of the PEV in miles per Wh
**P -** charging/discharging rate of the PEV battery
**I** - after-hours V2G participation rate of PEVs
**R** - pattern of solar radiation in the region under study over time period T

Note that the variables in capital letters represent vectors and the ones in lower case letters represent scalar values.

Figure 2 shows the simulation network with the following conductor, algorithm and simulator nodes.

S1 (fossil-fuel generator): simulates the power output including failures of existing fossil-fuel generators in the system; inputs: **s, r**
S2 (system load): simulates aggregate load in the grid; inputs: **L**
S3 (PEV): simulates PEVs including their driving patterns, battery state of charge, times when they are plugged in to the grid, etc.; inputs: **B, c, m, P, I**
S4 (solar PV generator): simulates the power output of solar PV generators; inputs: **R**
A1 (V2G algorithm): implements an algorithm that controls when PEVs are charged (G2V) or discharged (V2G)
C1 (determine generator failure rate): uses S1 and S2 to determine **r** that equates loss of load to **lole**
C2 (determine PEV ELCC): uses S1, S2, S3 and A1 to simulate the charging/discharging of PEVs under the V2G algorithm, power supply from existing generators, and system load to compute ELCC contribution of PEVs
C3 (determine solar PV ELCC): uses S1, S2 and S4 to simulate combined supply from existing (fossil-fuel) generators and solar PVs and aggregate system load to compute ELCC contribution of solar generators
C4 (determine combined ELCC of PEVs and solar PV): this is composed by combining the conductors C2 and C3; it uses C1 and C2 to compute the combined ELCC of PEVs and solar generators

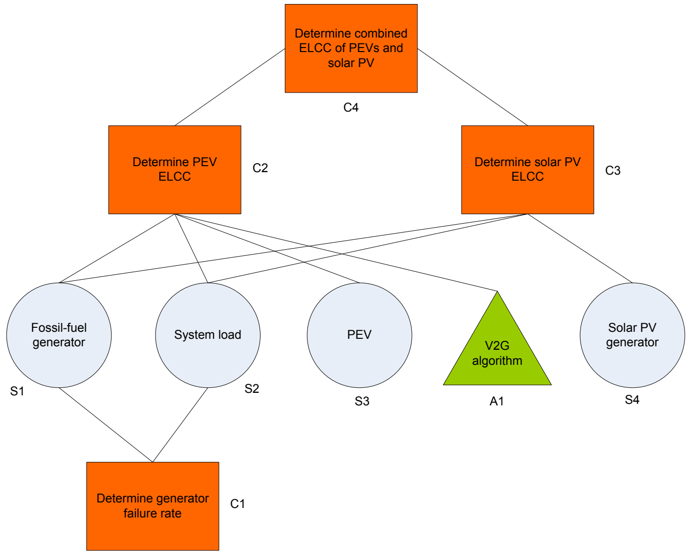


Figure 2: Simulator network for analyzing ELCC of PEVs and solar PVs

$\mathbf{f_1}$, $\mathbf{f_2}$, $\mathbf{g_1}$, $\mathbf{g_2}$ in the equations below represent the functions being simulated.

The simulation network is executed in phases. In Phase 1 of the simulation, conductor C1 is run to determine **r** such that we obtain the target **lole** given **L** and **s**.

$$\mathbf{lole = f_1(L,s,r)} \qquad \mathbf{(1)}$$

In Phase 2 of the simulation, conductor C2 is run. For each run i, it introduces $\mathbf{n_i}$ number of PEVs participating in the V2G program, and subtracts generating capacity $\mathbf{ep_i}$ from the existing capacity **s**. The goal is to come up with an $\mathbf{ep_i}$ such that the grid's LOLE remains unchanged at **lole**.

$$\mathbf{lole = f_2(L,\ s - ep_i,\ r,\ n_i,\ B,\ c,\ m,\ P,\ I)} \qquad \mathbf{(2)}$$

So $\mathbf{ep_i}$ represents the ELCC of a fleet of PEVs of size $\mathbf{n_i}$ (for given values of **B, c, m, P** and **I**).

In Phase 3, conductor C3 is run. For each run j, it introduces total solar photovoltaic rated capacity $\mathbf{v_j}$ into the grid and subtracts generating capacity $\mathbf{es_j}$ from existing capacity **s**. Note that at this phase we assume there is no V2G program available in the grid. The goal is to come up with an $\mathbf{es_j}$ such that the grid's LOLE remains unchanged at **lole**.

$$\mathbf{lole = g_1(L,\ s - es_j,\ r,\ v_j,\ R)} \qquad \mathbf{(3)}$$

So $\mathbf{es_j}$ represents the ELCC of a pool of solar generators with rated capacity $\mathbf{v_j}$ without any V2G program in the grid.

Finally, in Phase 4, conductor C4 is run. C4 is composed by combining nodes C2 and C3. C4 runs each iteration j in C3 in conjunction with **n** number of PEVs participating in the V2G program in C2. Generating capacity $\mathbf{esp_j}$ is subtracted from existing capacity **s**. So, C4 effectively implements a co-simulation by simulating PEVs (C2) and solar generators (C3) concurrently. Again, the goal is to come up with an $\mathbf{esp_j}$ for each $\mathbf{v_j}$ such that the grid's LOLE remains unchanged at **lole**.

$$\mathbf{lole = g_2(L,\ s - esp_j,\ r,\ v_j,\ R,\ n,\ B,\ c,\ m,\ P,\ I)} \qquad \mathbf{(4)}$$

So $\mathbf{esp_j}$ represents the ELCC of a pool of solar generators with rated capacity $\mathbf{v_j}$ with a V2G program in the grid that includes **n** PEVs.

Figure 3 shows the sample results from execution of C4 using actual load, supply and solar irradiance data from New York City over 2008-2009, about 54,000 PEVs, battery and mileage specifications of Chevy Volt, and average commute pattern of the region. It shows simulated load carrying capacity (ELCC) of solar generators in the presence of PEVs as solar capacity is increased from none to 5,000 MW.

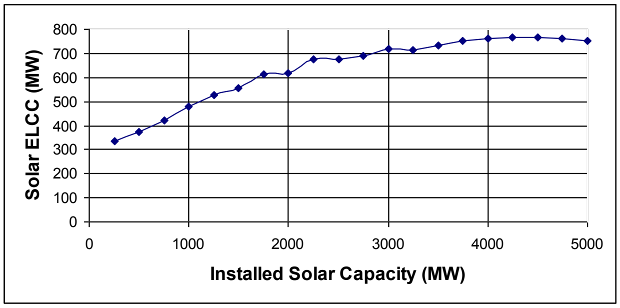


Figure 3: ELCC of solar generation with PEVs

*B. Determining boundaries of safe operation of the grid with PEVs*

The goal of this simulation exercise [3] is to determine the grid's reliability (as measured by LOLE) at various levels of ELCC expected from PEV V2G operations as the number of PEVs increases. Given a target reliability (typically 1 day of loss of load in 10 years), the exercise provides the boundaries of safe operation of the grid.

The simulation is implemented by reconfiguring the simulator network from the earlier example and adding a new conductor node C5 that leverages S1, S2, S3 and A1 to determine the grid's LOLE for a given expected V2G ELCC and a given number of PEVs. The network is outlined in Figure 4.

The simulation is executed in 3 phases as outlined in Figure 5. In Phase 1 of the simulation, conductor C1 is run to determine plant failure rate **r** to match target **lole** given **L** and **s** per (1).

In Phase 2 of the simulation, conductor C5 is run. For each run i, it introduces $\mathbf{n_i}$ number of PEVs participating in the V2G program, and subtracts generating capacity $\mathbf{ep_{i,}}$ which represents the expected ELCC from PEV V2G, from the existing capacity **s**. The goal is to determine the grid's LOLE **(lole)** per (2).

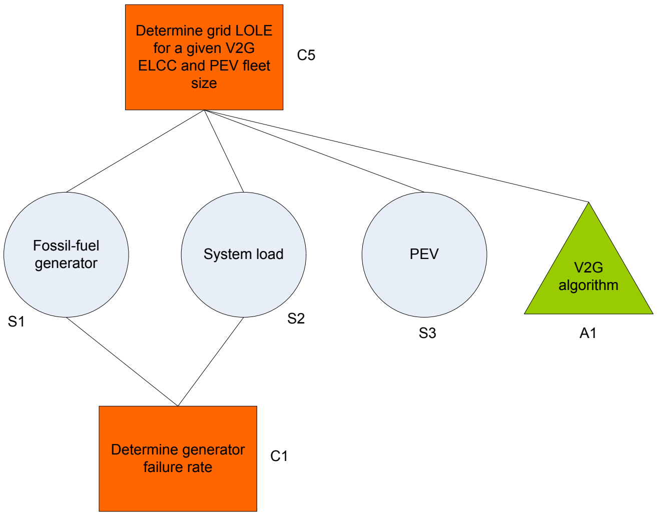

Determine grid LOLE for a given V2G ELCC and PEV fleet size
C5
Fossil-fuel generator
S1
System load
S2
PEV
S3
V2G algorithm
A1
Determine generator failure rate
C1

Figure 4: Simulator network for analyzing grid LOLE for various expected V2G ELCC and PEV fleet sizes

In Phase 3, C5 is run in a stress mode with the PEV penetration and expected V2G ELCC increased to very high levels to identify points where grid's reliability starts falling short of target.

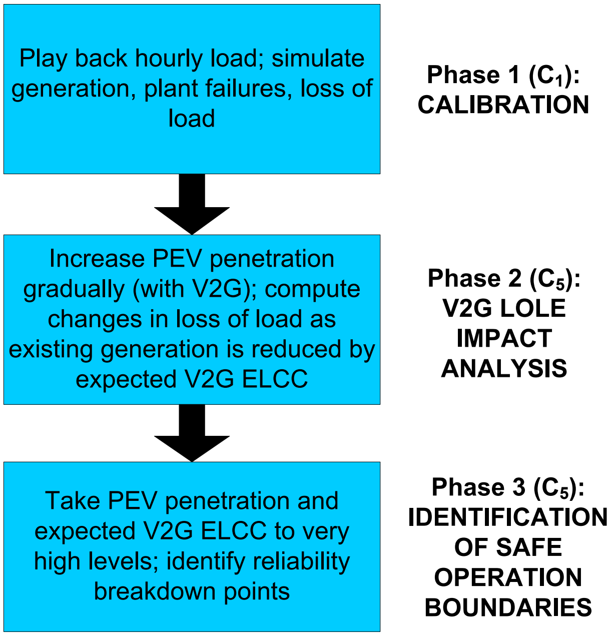

Play back hourly load; simulate generation, plant failures, loss of load
Phase 1 ($C_1$): CALIBRATION
Increase PEV penetration gradually (with V2G); compute changes in loss of load as existing generation is reduced by expected V2G ELCC
Phase 2 ($C_5$): V2G LOLE IMPACT ANALYSIS
Take PEV penetration and expected V2G ELCC to very high levels; identify reliability breakdown points
Phase 3 ($C_5$): IDENTIFICATION OF SAFE OPERATION BOUNDARIES

Figure 5: Three phases of the simulation to analyze PEV ELCC

The sample results of the simulation exercise are shown in a surface plot in Figure 6. It shows the output generated by C5 at various values of expected V2G ELCC and PEV fleet size including the region where the grid is no longer able to maintain its target reliability (1 day of LOLE in 10 years).

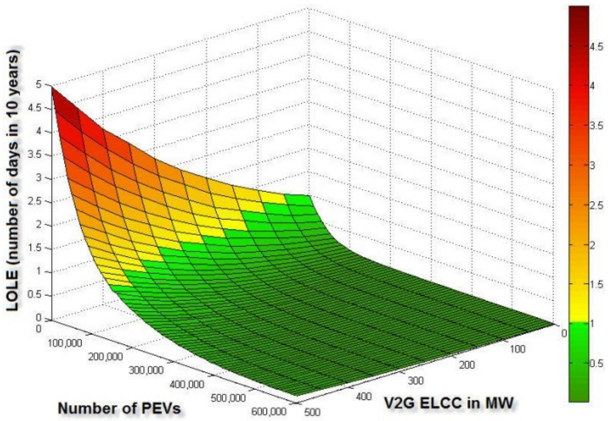


Figure 6: Sample output of the simulation showing grid reliability (LOLE) for different levels of PEV and expected V2G ELCC and, reliability breakdown points

### *C. Computing optimal solar penetration in the presence of PEVs*

The goal of this simulation exercise [4] is to compute the optimal installed solar generation capacity in a grid given a certain number of PEVs present so that the grid's overall ELCC is maximized. This simulation is implemented by the network in Figure 7 that is obtained by reconfiguring the simulator network from Figure 2. Conductor node C4 is replaced by a new conductor node C6 that runs co-simulation (C2 and C3) of PEVs and solar generators to determine combined ELCC as well as determines marginal gain in ELCC from small increases in solar capacity for a given PEV fleet size. It leverages these marginal gain figures to compute optimal solar capacity for the grid.

ELCC contributed by a combination of PEVs and solar generators is computed by function $\mathbf{g_3}$.

$$\mathbf{elcc = g_3(L, s, r, R, n, B, c, m, P, I)} \qquad \mathbf{(5)}$$

Optimal solar capacity $\mathbf{s^*}$ for a given $\mathbf{n=\lambda}$ is computed as

$$\mathbf{s^* = \min(s) \text{ s.t. } \frac{\delta(elcc)}{\delta(s)}\bigg|_{n=\lambda} \le \varepsilon \text{ for a given } L, r, R, B, c, m, P, I} \qquad \mathbf{(6)}$$

Node C6 implements (5) and (6). Sample output from the co-simulation executed by C6 is shown in Table I that shows the optimal combinations of installed solar capacity and PEV fleets where the combined ELCC of PEVs and solar generators is maximized as installed solar capacity and the number of PEVs are both increased in conjunction.

## V. Conclusion and Future Work

In this paper, we discuss how we designed a networked micro-simulator from scratch to study the effect of integrating renewable generators, micro-grids, and PEVs into the power grid in terms of the reliability of the grid, as well as increasing its load carrying capability. The modeling paradigm used here is flexible in the level of details one can model – depending on the need for accuracy and expediency in determining the effects. If detail household level models were created, one would model New York City's household power consumption, consumption patterns, stochastic models of consumption behaviors, individual PEVs, their commute behaviors etc., as well as the generators, and the weather patterns. Such micro-level modeling and simulation would take substantial amount of computing time, but will yield much better accuracy of the results. However, for policy decision makers, often such micro-level details are not essential, and thus, one can aggregate the entire segment of a population in a single agent in the model, and thereby gain computational efficiency at the expense of accuracy. We have not yet done an accuracy/expediency trade-off study for our micro-simulator – but have the ability to do so, as models of various levels of abstraction can be co-simulated in our framework, and experimental data can be obtained to do such a study. Our next course of action is to study such trade-offs in our framework.

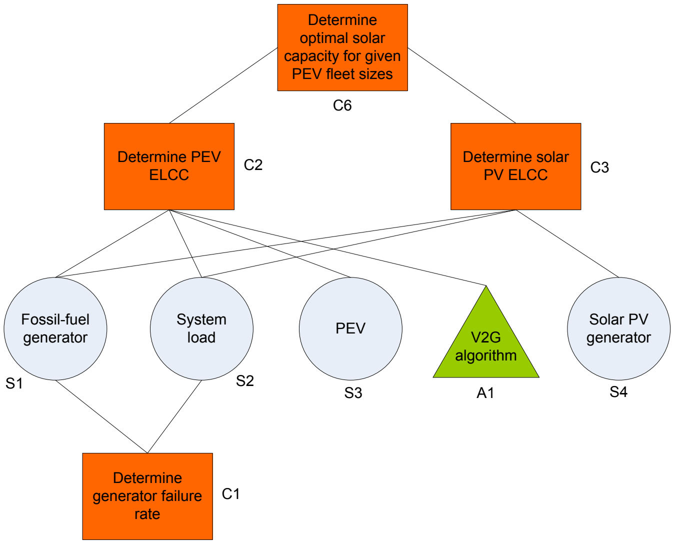


Figure 7: Simulator network for determining optimal solar capacity in the presence of PEVs via co-simulation

TABLE I: OPTIMAL SOLAR-PEV COMBINATIONS DETERMINED BY CONDUCTOR C6

| Number of PEVs | Installed Solar Capacity | ELCC in MW |
|---:|---:|---:|
| - | 2,500 | 630 |
| 50,000 | 2,500 | 780 |
| 100,000 | 2,500 | 880 |
| 150,000 | 2,500 | 980 |
| 200,000 | 2,750 | 1,025 |
| 250,000 | 3,000 | 1,080 |
| 300,000 | 3,000 | 1,080 |
| 350,000 | 3,000 | 1,080 |
| 400,000 | 3,000 | 1,130 |
| 450,000 | 3,000 | 1,130 |
| 500,000 | 3,000 | 1,130 |
| 550,000 | 3,000 | 1,130 |
| 600,000 | 3,000 | 1,130 |
| 650,000 | 3,000 | 1,130 |
| 700,000 | 3,000 | 1,130 |
| 750,000 | 3,000 | 1,130 |
| 800,000 | 3,000 | 1,130 |
| 850,000 | 3,000 | 1,130 |
| 900,000 | 3,000 | 1,130 |
| 950,000 | 3,000 | 1,130 |
| 1,000,000 | 3,000 | 1,130 |

However, using our micro-simulator at various levels of aggregation for different involved entities, we have already found some interesting results – which are often part of policy maker folk knowledge, but our simulators can yield with reasonable accuracy including implications of various policy decisions, incentives to adopt renewables, and effect of the intermittent resources on the overall reliability of a grid.

## REFERENCES

[1] Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp, "Optimizing Grid Contribution and Economic Returns from Solar Generators by Managing the Output Uncertainty Risk", IEEE PowerTech 2011, Trondheim, Norway

[2] Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp, "A framework for analyzing load-carrying-capacity of plug-in electric vehicles and impact on solar generators", 2011 IASTED European Conference on Power and Energy Systems, Crete, Greece

[3] Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp, "A Detailed Analysis of the Effective-Load-Carrying-Capacity Behavior of Plug-in Electric Vehicles in the Power Grid", IEEE PES Conference on Innovative Smart Grid Technologies 2012, Washington, D.C.

[4] Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp, "Computing Optimal Solar Penetration in the Presence of Plug-in Electric Vehicles", IEEE EnergyTech 2012, Cleveland, OH

[5] Soumyo V. Chakraborty, Sandeep K. Shukla, James Thorp, "System Imbalance Minimizing Renewable Generation Portfolio Selection in the Presence of Plug-in Electric Vehicles", IEEE PES General Meeting 2012, San Diego, CA

[6] Ptolemy II, http://ptolemy.eecs.berkeley.edu/

[7] Cathal O'Donoghue & Howard Redway & John Lennon, 2010. "Simulating Migration In The Pensim2 Dynamic Microsimulation Model," International Journal of Microsimulation, International Microsimulation Association, vol. 3(2), pages 65-79.

[8] EUROMOD, Website: https://www.iser.essex.ac.uk/euromod

[9] Annual Monitoring Report 2009, Social Situation Observatory, European Commission

[10] PECAS, http://hbaspecto.com/styled/index.html

[11]SIMTRAFFIC, http://trafficwareinc.com/transportation/product/simtraffic-8-0

[12] Kurian AW, Munoz DF, Rust P, Schackmann EA, Smith M, Clarke L, Mills MA, Plevritis SK. Online Tool to Guide Decisions for BRCA1/2 Mutation Carriers. *J Clin Oncol* 2012 Feb 10;30(5):497-506.

[13] Mandelblatt JS, Cronin KA, Berry DA, Chang Y, de Koning HJ, Lee SJ, Plevritis SK, Schechter CB, Stout NK, van Ravesteyn NT, Zelen M, Feuer EJ. Modeling the impact of population screening on breast cancer mortality in the United States. *Breast* 2011 Oct;20 Suppl 3:S75-81

[14] Mathematica, http://www.wolfram.com/

[15] MATLAB, http://www.mathworks.com/products/matlab

# On the Capacity value of Renewable Energy Sources in the presence of Energy Storage and Ramping Constraints

Alberto J. Lamadrid
Department of Economics
Lehigh University
Bethlehem, PA, 18015
Email: ajlamadrid@lehigh.edu

Tim Mount and Ray Zimmerman
Department of Applied Economics and Management
and School of Electrical and Computer Engineering
Cornell University
Ithaca, NY, 14853

***Abstract*—The objective of this paper is to analyze the value that renewable energy sources (RES) have in providing capacity, and specifically study the interactions that take place in the presence of energy storage systems (ESS) and ramping constraints. To examine this question, we use a new analytical framework that optimizes over different high and low probability scenarios using a stochastic, security constrained Optimal Power Flow (S-SC-OPF). We are interested in the effect of adding a significant amount of RES and analyze the individual generator response and the consequences for overall system metrics. Past studies have shown that, while higher RES penetrations are usually associated to lower system costs, including the provision of ancillary services, the most common direct collateral consequence is the increase in the total generating capacity needed to reliably operate the system [1]. Our model determines the amount of reserves as an endogenous variable, given a set of credible contingencies and a characterization of the uncertainty coming from the renewable energy sources. The main advantage of this model is the explicit inclusion of the cost for the capacity required, both for contingency reserve and for ramping transitions between periods of analysis, as well as a valuation of the wear-and-tear incurred by the generators in these transitions.**

**Our approach simulates several periods applying a S-SC-OPF that minimizes the total system costs including the procurement of energy and ancillary services. We applied this model to a reduction of the Northeastern Power Coordinating Council (NPCC) and calibrate the demand to a day similar to the loading conditions in a hot summer. The system results show reduction of system costs adding wind and storage into the system while helping reduce capacity.**

## I. Introduction

The last decade has witnessed a large scale integration effort of Renewable Energy Sources (RES) into the power system. This development has prompted the development of a voluminous literature addressing the advantages and challenges that RES growth pose to the way the power system is operated, in the regulatory framework of maintaining reliability of service. The secure and reliable operation of the power system implies its ability to withstand both discrete and continuous disturbances. Once RES become a more prevalent source in the amounts of energy procured to the power system, the continuous disturbances related to these stochastic sources will require the increased commitment of reserves. The reserves required range from contingency reserve, with the ability to respond in very short timeframes [2], to load following reserve, needed to follow the changing demand over a day. This increase in the cost of reserves, a capacity charge, usually counterbalances the savings coming from displacement of expensive generation sources, an energy saving. In fact, evidence coming from places with high wind penetrations like Ireland and Germany show that the costs associated to follow both demand and the continuous RES disturbances have increased, especially related to maintenance and the possibility of failures[3], [4].

The primary objective of this paper is to evaluate whether the RES capacity installed in a network can provide some reserve capacity, and its interaction with Energy Storage Systems (ESS), in the presence of constraints that limit the technical capability to ramp. RES typically displaces the most expensive sources of generation in an un-congested network. However, transmission constraints in certain areas can in fact electrically isolate these resources. This *de facto* islanding limits the ability to cover the possible demand (and the situations in which the system is disturbed) in places that are characterized by high economic activity (usually urban demand centers). Additionally, the locational marginal prices (LMP's) can be depressed due to the transmission congestion, leading to lower revenues for wind generators (i.e., a "financial adequacy" problem for RES). This contrasts with low demand periods, in which energy can freely flow in the system and in many instances RES will be the marginal units.

ESS are generally expensive sources of energy, and the exclusive use for utility-scale services makes its capital cost in general non-economical endeavor. Nevertheless, some demonstration projects are underway [5], and there are intent to continue these projects. A different kind of storage resource is actually not storage *per se* but rather a change in the demand pattern over a day. This involves the inter temporal trade-off of consumption across time, taking into account the dynamic aspects that this decision implies (e.g. changes in prices). This kind of application is more likely to be used for provision of thermal demand services. The services rendered can range from cooling during summer periods, by producing ice at low demand periods and melting it at peak demand periods, to heating using heat pumps and ceramic elements. The main theoretical benefits of deploying deferrable demand targeting the temperature-sensitive demand are twofold: it prevents the congestion buildup of the transmission system at peak times, and the capital cost of implementation is shared between the main purpose (thermo services) and the secondary purpose (decrease in electric demand).

The implementation of deferrable demand for thermo services has an additional benefit in terms of the operation of the system: the system operator (SO) can involve more directly resources that, while in the realm of the distribution network, are predictable and can provide for a better operation of the bulk system. The task for the SO is to maximize the total social welfare[1], which includes the minimization of the cost of energy and reserves, as well as the maximization of the consumer surplus in the form of minimal interruptions and load not served (LNS). Historically, the management given to the system assumed completely exogenous demand, and sources of generation with generally large economies of scale, catered at covering the peak demand of the system. This peak demand has outpaced the growth of demand, and therefore the system is left with spare capacity that is only used a number of hours a year. Nevertheless, the pace of demand seems to be growing faster than the peak demand according to the latest Reliability assessments [7]. This is in part driven by increased Demand Side Management (DSM), totaling around 80GW in ten years for the full North American System.

The paper is organized as follows. Section II briefly summarizes the motivation and literature relevant to this work. Section III presents a general description of a Security Constrained Optimal Power Flow (SC-OPF) followed in Section IV by a description of its specific features of our proposal, such as storage capacity, the representation of deferrable demand and the variability of the potential wind generation. The results and the description of the case studies are presented in Section V. The paper ends with the conclusions in Section VI that include our recommendations for further research directions and guidelines for appraising the capacity value of RES.

## II. Literature Review

The motivation for this work comes from the concerns to properly assess the contribution that RES makes into the power system. This is a need that has been recognized in the area, leading to the creation of task forces in charge, with inputs from a variety of stakeholders [8]. Overall, depending on the topology of the network, it has been found that diversification on the supply of RES can lead to better capacity contributions and lower reserve requirements to reliably operate [9]. The main question that researchers are trying to identify is what are the important features that need to be incorporated in the models used for the operation of the system. While there is consensus on the importance of including security constraints in the dispatch [10], in reality system operators (SO's) are faced with the challenge of obtaining models that are implementable in the time scales required. This constraint has led them to use heuristics and hard constraints based on the experience operating the system [11]. To better incorporate the security constraints necessary to reliably operate the system, a stochastic program allows to co-optimize the energy and the ancillary services needed [12]. In the context of higher penetration of renewables, there have been a number of proposals that suggest different ways to make the problem more manageable [13], [2]. These approaches leverage on the past literature analyzing the combined Unit Commitment and OPF frameworks suggested [14], [15], [16], [17]. However, the UC problem is not common to all markets, and the final market design can still benefit from understanding whether the self commitment of generators can lead to optimal market solutions[18] [19]. Irrespective of the final market structure, there is a general agreement regarding the effect of renewables on the operation of the system [20], [21], [22]. The approach considered in this paper is germane to [12]'s framework, with robust determination of the reserves for contingencies, load following and ramping for multiple periods. We call this framework the 'Super-OPF', and it can be described as a stochastic multi-period security constrained AC OPF with co-optimizing endogenous reserves to provide ramping to mitigate wind variability and cover a set of credible contingencies. This model is an extension of the model proposed in [23], accounting for several periods in which the demand could be transferred.

The case study presented in this article uses an advanced, multi-period version of the Super-OPF, developed at Cornell (the second generation SuperOPF). The main challenge for the effective usage of the model is the proper calibration of the input data. This is discussed in the following sections.

## III. Formulation of the Analytical Model

A new second-generation 'SuperOPF', is used for this analysis. The objective function of the new SuperOPF is to maximize the expected sum of producer and consumer surplus over a defined time horizon (e.g. 24 hours) for a set of credible contingencies with low individual probability, and including uncertainty about the forecasts of availability in the RES. The model also allows for storage and deferrable demand. Rather than using the standard criterion of minimizing cost subject to covering physical contingencies, the model allows shedding load at a high Value of Lost Load (VOLL) if it is economically efficient to do so. This formulation determines the optimal dispatch of a set of previously committed generating units subject to their physical characteristics (e.g., rated capacity, cost and ramping capabilities) and the network's topology (e.g. transmission line constraints). The model solves the expected cost for a number of high probability cases for stochastic RES generation ("intact" scenarios), as well as for a set of credible contingencies that occur relatively infrequently.

[1] The requirement to minimize the interruption, or Load Not Served (LNS), for customers [6] means that, if valuing demand at the Value of Lost Load (VOLL) in the objective function, the objective function is maximizing consumer surplus.

The expected cost is minimized over the intact scenarios and the contingencies using probabilities that reflect the relative likelihood of the different states of the system occurring. This formulation has the advantage of determining endogenously the amounts of different ancillary services (e.g., contingency reserve and ramping reserve to mitigate wind variability) needed to meet the load profiles and maintain the reliability of the delivery system. The optimal dispatch is determined as a day-ahead settlement, incorporating the best available information that the Independent System Operator (ISO) has at that time.

A simplified formulation of the objective function for the problem is shown in (1) and the notation is defined in Table I.

$$\begin{aligned}\min_{G_{itsk}, R_{itsk}, \mathrm{LNS}_{jtsk}} & \sum_{t\in\mathscr{T}} \sum_{s\in\mathscr{S}^t} \sum_{k\in\mathscr{K}} \pi_{tsk} \Bigg\{ \sum_{i\in\mathscr{I}} \Big[ C_{G_i}(G_{itsk}) + \\ & \mathrm{Inc}^{+}_{its}(G_{itsk} - G_{itc})^{+} + \mathrm{Dec}^{-}_{its}(G_{itc} - G_{itsk})^{+} \Big] \\ & \sum_{j\in\mathscr{J}} \mathrm{VOLL}_j \mathrm{LNS}(G_{tsk}, R_{tsk})_{jtsk} \Bigg\} + \\ & \sum_{t\in\mathscr{T}} \rho_t \sum_{i\in\mathscr{I}} [C^{+}_{R_{it}}(R^{+}_{it}) + C^{-}_{R_{it}}(R^{-}_{it}) + C^{+}_{L_{it}}(L^{+}_{it}) + \\ & C^{-}_{L_{it}}(L^{-}_{it})] + \sum_{t\in\mathscr{T}} \rho_t \sum_{s_2\in\mathscr{S}^t} \sum_{s_1\in\mathscr{S}^{t-1}} \sum_{i\in\mathscr{I}^{ts_2 0}} \\ & \big[ \mathrm{Rp}^{+}_{it}(G_{its_2} - G_{its_1})^{+} + \mathrm{Rp}^{-}_{it}(G_{its_2} - G_{its_1})^{+} \big] \end{aligned} \tag{1}$$

Subject to meeting demand and all of the nonlinear AC constraints of the network.

The nodal levels of demand are fixed blocks for each time period and are modeled as negative injections with associated negative costs (the VOLL per load at the substation level). Since this specification allows for LNS in some states of the delivery system, valued at VOLL, minimizing the expected cost, including load shedding as a cost, corresponds to maximizing the expected sum of consumer and producer surplus.

The most important features of the SuperOPF for this analysis are that 1) the stochastic characteristics of potential wind generation at multiple sites can be represented in different ways, 2) the amount of conventional generating capacity, including reserves, needed to maintain Operating Reliability is determined endogenously, and it depends on how the stochastic characteristics of potential wind generation are represented, and 3) the additional ramping costs caused by the inherent variability of wind generation can be incorporated into the objective function.

## IV. Model Specification

The calibration of input data was done using publicly available sources, and it encompasses the modification of the test network and the modeling of RES generation, deferrable demand and utility-scale Energy Storage Systems (ESS) co-located at the wind sites.

TABLE I
Definition of Variables, simplified Formulation

| | |
|---|---|
| $\mathscr{T}$ | Set of time periods considered, $n_t$ elements indexed by $t$. |
| $\mathscr{S}^t$ | Set of scenarios in the system in period $t$, $n_s$ elements indexed by $s$. |
| $\mathscr{K}$ | Set of contingencies in the system, $n_c$ elements indexed by $k$. |
| $\mathscr{I}$ | Set of generators in the system, $n_g$ elements indexed by $i$. |
| $\mathscr{J}$ | Set of loads in the system, $n_l$ elements indexed by $j$. |
| $\pi_{tsk}$ | Probability of contingency $k$ occurring, in scenario $s$, period $t$. |
| $\rho_t$ | Probability of reaching period $t$. |
| $G_{itsk}$ | Quantity of apparent power generated (MVA). |
| $G_{itc}$ | Optimal contracted apparent power generated (MVA). |
| $C_G(\cdot)$ | Cost of generating $(\cdot)$ MVA of apparent power. |
| $\mathrm{Inc}^{+}_{its}(\cdot)^{+}$ | Cost of increasing generation from contracted amount. |
| $\mathrm{Dec}^{-}_{it}(\cdot)^{+}$ | Cost of decreasing generation from contracted amount. |
| $\mathrm{VOLL}_j$ | Value of Lost Load, (\$). |
| $\mathrm{LNS}(\cdot)_{jtsk}$ | Load Not Served (MWh). |
| $R^{+}_{it} < \mathrm{Ramp}_i$ | $(max(G_{itsk}) - G_{itc})^{+}$, up reserves quantity (MW) in period $t$. |
| $C^{+}_{R}(\cdot)$ | Cost of providing $(\cdot)$ MW of upward reserves. |
| $R^{-}_{it} < \mathrm{Ramp}_i$ | $(G_{itc} - min(G_{itsk}))^{+}$, down reserves quantity (MW). |
| $C_R(\cdot)$ | Cost of providing $(\cdot)$ MW of downward reserves. |
| $L^{+}_{it} < \mathrm{Ramp}_i$ | $(max(G_{i,t+1,s}) - min(G_{its}))^{+}$, load follow up (MW) $t$ to $t+1$. |
| $C^{+}_{L}(\cdot)$ | Cost of providing $(\cdot)$ MW of load follow up. |
| $L^{-}_{it} < \mathrm{Ramp}_i$ | $(max(G_{its}) - min(G_{i,t+1,s}))^{+}$, load follow down (MW). |
| $C_L(\cdot)$ | Cost of providing $(\cdot)$ MW of load follow down. |
| $\mathrm{Rp}^{+}_{it}(\cdot)^{+}$ | Cost of increasing generation from previous time period. |
| $\mathrm{Rp}^{-}_{it}(\cdot)^{+}$ | Cost of decreasing generation from previous time period. |

### A. The Test Network

Figure 1 shows the one-line diagram of the network used in the case study. This is a New York and New England centric reduction of the Northeast Power Coordinating Council (NPCC) network [24], that has been modified to include very detailed information of the generating units at each bus, obtained from the PowerWorld Corporation. We call this case the North Eastern Test Network (NET Net).

The total load of the system is around 138 GW, and the generation capacity available is 143 GW [24]. For the simulation, one day in a high demand period was calibrated (following historical load information from August 2008), distinguishing the profiles between urban and rural nodes. The peak system load occurs at 3PM, caused mainly by the high demand at urban nodes. Table II has a summary of the generation capacities and loads for each Regional Transmission Organization (RTO) considered. The average fuel costs vary by location, with the highest coal and oil costs

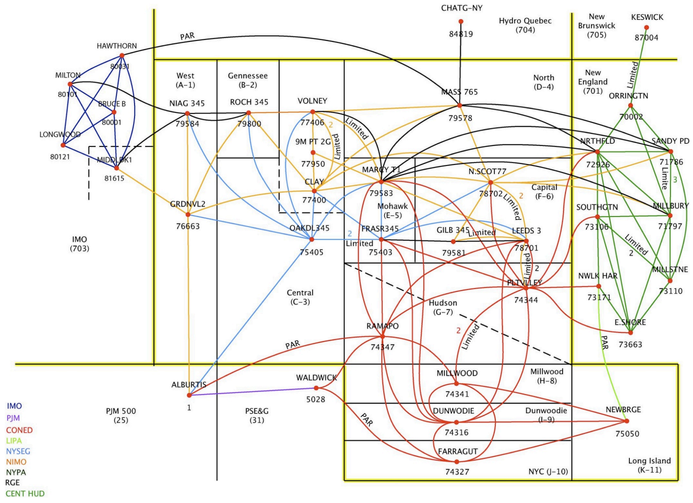


Fig. 1. A One-Line-Diagram of the 36-Bus Test Network.

in New England, and the highest natural gas costs in New York, PJM and Ontario.

TABLE II
SUMMARY OF GENERATION CAPACITY AND LOAD

| | Capacity per Fuel Type (MW) | | | | | Total Cap. | Load |
|---|---|---|---|---|---|---|---|
| Location (RTO) | coal | ng | oil | hydro | nuclear | | |
| isone | 1,840 | 9,219 | 4,327 | 1,878 | 5,698 | 22,962 | 23,847 |
| marit. | 2,424 | 1,072 | 22 | 641 | 641 | 4,800 | 3,546 |
| nyiso | 4,557 | 18,185 | 5,265 | 7,345 | 4,714 | 40,066 | 38,274 |
| ont. | 5,287 | 3,594 | 0 | 779 | 12,249 | 21,910 | 21,158 |
| pjm | 14,453 | 14,611 | 8,915 | 2,604 | 12,500 | 53,083 | 51,588 |
| quebec | 0 | 0 | 0 | 800 | 0 | 800 | 0 |
| Total | 28,562 | 46,681 | 18,530 | 14,048 | 35,802 | 143,707 | 138,412 |
| Rp.C.[b] | 30 | 10 | 10 | 60 | 60 | | |

[a] Values shown are taken as peak values.
[b] Ramping costs ($t/MW).

The cost of ramping services is assigned by fuel type using quadratic cost functions. This is consistent with our previous work (see [25]). The values set were relatively high for baseload units and lower for peaking units, signaling the different generators' willingness to be moved from their current operating point. This modeling approach implicitly presumes homogenous conditions for all generators of a given fuel type.

The loads available in the system were classified as either rural or urban, each one with a different profile over the day, with more pronounced peaks for the urban loads compared to the rural loads. The changes observed over one day were derived from 2008 historical data for the study area, allowing for different changes in hour to hour demand according to the location of the load. The Value of Lost Load (VOLL) is also dependent on location, with a value of $10,000/MWh for urban areas and $5,000/MWh for rural areas.

### B. Specifications for Renewables Energy Sources

Due to the location of the network, we focused on the adoption of wind as the stochastic generation source. This study analyzes a case with a wind penetration close to 20% of the total system load. The specification of the wind resource is divided into two main tasks: assigning locations and sizes of the wind resources on the test network, and characterizing the variability of these wind resources.

The locations of the wind farms are obtained from the National Renewable Energy Laboratory (NREL) Eastern Wind and Transmission Study (EWITS) data [21]). The matching of the data between the NREL database and the available buses in the reduced NPCC network is done using a principal component analysis (PCA). This matching leads to nine locations in New York and seven sites in New England that correspond to specific buses on the network[2].

The spatial and temporal variability of the wind resources are captured using a clustering analysis with a *k-means* methodology for scenario reduction [26]. The determination of the clusters is done using the hourly wind speeds for different locations from the EWITS data. The wind speeds are then converted to the potential wind generation using a multi-turbine modeling approach [27]. The information was grouped by hours, and therefore all decisions are in hourly steps. In case the data had finer resolution, an arithmetic mean was used. The data for three years (2008-2010) is then grouped to select similar days in terms of the variability observed, which are then reduced to four scenarios. The clustering in the four scenarios for a 24 hour spell allows to calculate the transition probability matrices from one scenario to another scenario in the next hour. The overall objective is to model the variability of wind realistically in a way that captures geographic averaging and is consistent with the EWITS data from NREL. In this setup, there is a 'high wind' scenario, a 'low wind' scenario and two intermediate scenarios, with different profiles for each wind site.

### C. Specifications for Deferrable Demand

The concept of using deferrable demand as a mechanism to improve the management of the system management to work done in the late 80's [28], [29], laying the foundation for the modeling used in this paper. The specification of deferrable demand assumes that the timing of the purchase of electricity for specified percentages of the total hourly demand can be decoupled from the timing of the energy services delivered. Examples include charging the batteries in electric vehicles and thermal storage for space conditioning (e.g. traditional central Air Conditioning (AC) systems can be augmented with ice batteries). In effect, this differentiates the demand in two hourly profiles: Conventional demand must be supplied in real time from the grid. Although deferrable demand must also be supplied in real time, the sources of supply can come from thermo storage and/or the grid.

In the case study, five urban buses are selected. The total amount of deferrable demand (as a percentage of the total demand) set at a bus are location specific. Deferrable demand

[2] The location of the wind farms are in the following buses: Orrington, Sandy Pond, Millbury, Northfield, Southington, Millstone, Norwalk harbor, Millwod, Newbridge, 9Mile Point, Leeds, Massena, Gilboa, Marcy, Niagara and Rochester.

accounts for 7% of the total demand for the New York City buses and for 6.25% at the Buffalo bus. For the Millbury bus and the Sandy Pond bus in New England, the values were set to 2% and almost 5% of the total daily demand, respectively. These values correspond to the average values estimated econometrically from historical patterns of demand in the different regions for the years 2007 to 2010 [30].

### D. Specifications for Utility-Scale Storage

The Energy Storage Systems (ESS) collocated at the wind sites are modeled as special generators that can both inject or draw power from the system. The rate of charging and discharging is independently set, to approach the physical behavior of different ESS technologies. The energy available in any ESS can be used to provide energy in the different wind scenarios and to help support the grid in contingencies. The optimal use of storage is dependent in part on the final value assigned to the stored energy. If it is valued at zero, then stored energy is always used in contingencies and in the last hour of the planning horizon. There is, however, an opportunity cost for discharging the ESS that provides a high threshold for discharging. If the nodal price in the terminal state is very low, for example, it would in reality be optimum to not discharge the ESS and wait until a later period when the price is higher than the high threshold. A similar argument can be made for charging the battery, and a low threshold provides the opportunity cost for charging. It is optimum to charge the ESS if the price is below the low threshold. If the price is between the two thresholds, the optimum is to do nothing and save the stored energy for use later.

In the empirical analysis, the ESS are located at the same buses as the wind farms and the total power capacity of the ESS is the same as the total deferrable demand energy. This specification makes it easier to make comparisons between cases with ESS and deferrable demand. The maximum hourly power available per ESS is set to be one sixth of the energy capacity (i.e. it would take six hours to completely deplete a fully charged ESS, if the discharging efficiency is set to 100%).

To determine the threshold price of the stored energy for discharging, the initial pattern of dispatch for generators and the initial amounts of stored energy, an iterative process is implemented in which the daily dispatch is simulated several times, using the same input specifications, until the differences in the threshold price and initial conditions are stable and below a tolerance. These initial conditions can be considered, therefore, as a steady state solution for a series of identical days.

## V. Results of the Case Study

The results in this section summarize the capacity value of wind for four different cases. The total cost of serving a given demand profile for a 24-hour period is provided as a reference. The injections and exports from outside of the New York and New England region (NYNE) are fixed, to focus on this territory. For this reason, the results include information only for NYNE, and the locations of wind farms and storage are all in this region. The analysis assumes that the wholesale market is deregulated and run by an Independent System Operator (ISO).

We focus on the cost of service as opposed to the customer payments because of the associated decrease in the energy prices when renewable energy sources are available. We have argued in earlier research that this emphasis ignores the financial adequacy issue for conventional generators [31]. Since the offers submitted by renewable sources are effectively zero, average nodal prices are generally lower. Therefore, these new renewable sources displace fossil fuels and the conventional generators receive less net revenue to cover their capital expenses. To rectify this situation and still maintain system reliability, generators are further compensated in capacity markets that help to provide the "missing money". To avoid the distortions from evaluating a policy based solely on the wholesale payments from customers, the different cases are evaluated using measures that reflect the total system costs. The main interest of the analysis is the management of stochastic wind generation and the provision of load following reserves. For this reason the time steps are hours, therefore abstracting from the provision of dynamic services in real time that require rapid changes in the dispatch patterns to balance demand and supply in response to the variability of both the intermittent renewable sources and demand.

### A. The Capacity Value of Wind

The analysis of the effects of adding wind and the different types of storage are illustrated using the following four cases:

1) Case 1: No Wind: Initial system
2) Case 2: Case 1 + 29GW of Wind Capacity at 16 locations.
3) Case 3: Case 2 + 5.5GW (power capacity) of Deferrable Demand (DD) at five load centers .
4) Case 4: Case 2 + 5.5GW (power capacity) of Energy Storage Systems (ESS) collocated at the 16 wind farms.

This simulation starts at midnight and finishes 11PM. Although the initial starting hour does affect the results with stochastic inputs, an analysis of this topic will be subject of another paper.

Figure 2 shows the amount of capacity needed from both conventional generators and the different forms of storage/deferrable demand in the system. The obvious implication from this result is that wind in fact reduces the amount of capacity needed at all hours of the day. In the calculation of this capacity value, a combination of low and high probability scenarios is considered, meaning that whenever a conventional generator is used to either cover an outage (low probability event), or to counteract the variability of wind. The spatial distribution of the resources shows a higher capacity value in places electrically close to the main demand centers. This phenomenon occurs in buses close to the demand centers with high wind availabilities: Bus 36 in Rochester, and 35, Niagara, both in New York State.

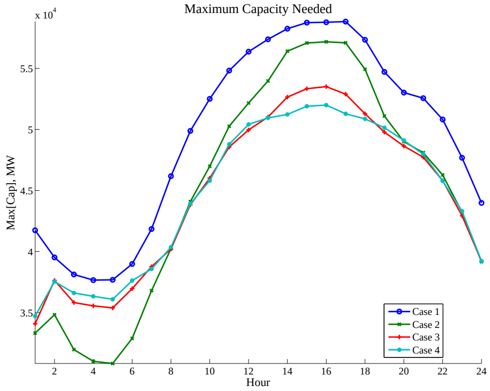


Fig. 2. Maximum Capacity Needed From Conventional and ESS Sources

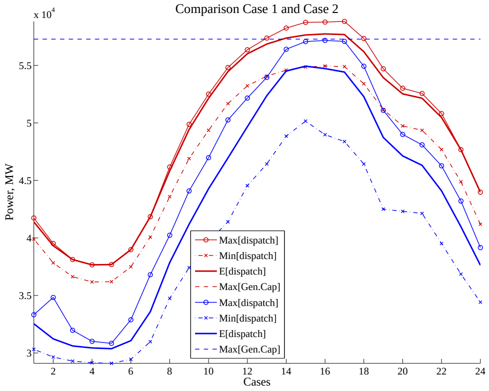


Fig. 3. Capacity Needed From Conventional and ESS Sources

Table III summarizes the main results for the four cases. The direction and magnitude of the changes in operating costs (row 1) are consistent with the expectations, with decreases observed with the addition of wind, the demand response management and the use of utility-scale storage.

TABLE III
SUMMARY OF THE DAILY RESULTS FOR THE FOUR CASES

| | Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|---|
| **1. e[Operating Costs] (k$/day)** | 51,582 | 42,777 | 41,654 | 41,487 |
| **2. e[Ramping Costs] (k$/day)** | 464 | 1,519 | 1,215 | 1,252 |
| **3. M[GenCap] (MW)** | 58,828 | 57,282 | 53,743 | 52,311 |
| **4. M[GenCapE] (MW)** | 58,828 | 57,282 | 53,743 | 57,654 |
| **5. e[Wind Energy] (MWh)** | 719 | 148,410 | 159,913 | 162,565 |
| **6. Total Surplus (k)** | 8,955,181 | 8,967,077 | 8,968,631 | 8,969,184 |

The expected Ramping Costs (row 2) are larger in Case 2 than in the storage cases (Cases 3 and 4), showing the value of storage for mitigating the variability of the potential wind generation.

The value of capacity for wind can be observed from rows 3 and 4, measuring the maximum generation capacity from conventional sources, and including the storage capacity in case 4. This capacity is the minimum necessary value rehired by the system operator to be $n-1$ secure. Therefore, decreases in the value of this capacity signal the ability of the stochastic resource to cover the different operating regimes that the system may face

To better understand the value of the wind resources, Figure 3 compares the hour-to-hour capacity of Case 1 (red lines) and Case 2 (blue lines) with the uncertainty bands added. Not only the maximum but also the expected and minimum follow a similar pattern to the one observed in Figure 2

### B. Computational requirements and Methods

Our method is implemented using MATPOWER's extensible architecture [32]. The solvers used include CPLEX, Gurobi and Mosek for the DC versions of the problem, and a Primal-Dual Interior Point Method (PDIPM) implemented in C, with a mex interface [33]. A typical NET Net case with 36 buses, 81 conventional generators, 16 wind generators, 121 transmission lines, 4 scenarios, 2 contingencies and one base case, and 24 time periods leads to a quadratic problem (QP) with 167,308 variables and 364,816 constraints. We did our simulations on a Mac Pro running OS-X Lion server with 12 cores, and allocated 10 threads to the job. The average setup time for the problem was 27.86 seconds. The average solving time is 405.63 seconds. We have been testing larger systems including a modified IEEE 118 bus system and a 279 bus system. The mean solving time for the 118 bus system is 6.8534e+03 seconds. We are now considering some methods for scaling up the problem, as this is an area that requires further work.

## VI. CONCLUSIONS

The gains form adoption of renewables have been put in doubt when taking into account the multiple costs that need to be internalized by the system operator and the different agents operating in the system. An area of importance for all participants relates to the necessary capacity required to operate reliably. Our case study shows that, at least for the topology observed and with the historical patterns of demand observed, there is in fact some support provided by wind generators for reliability purposes. But to really be able to affectively compensate the different generators in the system, a layer of information with updated situational awareness is necessary, collecting the measurements necessary to price the new costs incurred and internalize the ramping of conventional demand. This collection of ramping information and the proper appraisal of the economic effects of this activity is an area that regulators should encourage, and the different stakeholders, including utilities should help to support.

## ACKNOWLEDGMENT

This research was supported by the US Department of Energy through the Consortium for Electric Reliability Technology Solutions (CERTS) and the Office of Electricity Delivery and Energy Reliability under the project "The Future Grid to Enable Sustainable Energy Systems", and by the Power Systems Engineering Research Center (PSERC). The authors are responsible for all conclusions presented in the paper and

the views expressed have not been endorsed by the sponsoring agencies.

## REFERENCES

[1] M. Ilic, L. Xie, and J.-Y. Joo, "Efficient coordination of wind power and price-responsive demand part i: Theoretical foundations," *Power Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 1875 –1884, nov. 2011.

[2] A. Papavasiliou, S. Oren, and R. O'Neill, "Reserve requirements for wind power integration: A scenario-based stochastic programming framework," *Power Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 2197 –2206, nov. 2011.

[3] N. Troy, D. Flynn, M. Milligan, and M. O'Malley, "Unit commitment with dynamic cycling costs," *Power Systems, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[4] S. Spiecker and C. Weber, "Integration of fluctuating renewable energy; a german case study," in *Power and Energy Society General Meeting, 2011 IEEE*, july 2011, pp. 1 –10.

[5] J. Kumagai, "A Battery as Big as the Grid," *Spectrum*, vol. 49, no. 1, pp. 45–46, Jan. 2012. [Online]. Available: http://spectrum.ieee.org/energy/the-smarter-grid/a-battery-as-big-as-the-grid#.UGXbK7x2ISk.citeulike

[6] NERC, *Reliability Standards for the Bulk Electric Systems of North America*, NERC, Ed. 3353 Peachtree Road NE Suite 600, North Tower, Atlanta, GA 30326: North American Electric Reliability Corporation, 2013. [Online]. Available: http://www.nerc.com/docs/standards/rs/Reliability_Standards_Complete_Set.pdf

[7] ——, *Long-Term Reliability Assessment*, NERC, Ed. 116-390 Village Road, Princeton, NJ, 08540: North American Electric Reliability Corporation, 2012. [Online]. Available: http://www.nerc.com/files/2012_LTRA_FINAL.pdf

[8] A. Keane, M. Milligan, C. J. Dent, B. Hasche, C. D'Annunzio, K. Dragoon, H. Holttinen, N. Samaan, L. Soder, and M. O'Malley, "Capacity value of wind power," *Power Systems, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2010.

[9] D. Halamay, T. Brekken, A. Simmons, and S. McArthur, "Reserve requirement impacts of large-scale integration of wind, solar, and ocean wave power generation," *Sustainable Energy, IEEE Transactions on*, vol. 2, no. 3, pp. 321 –328, july 2011.

[10] J. Condren, T. Gedra, and P. Damrongkulkamjorn, "Optimal power flow with expected security costs," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 541–547, May 2006.

[11] E. Ela, M. Milligan, and B. Kirby, "Operating reserves and variable generation," National Renewable Energy Laboratory, Tech. Rep., 2011. [Online]. Available: http://www.nrel.gov/docs/fy11osti/51978.pdf

[12] J. Chen, T. D. Mount, J. S. Thorp, and R. J. Thomas, "Location-based scheduling and pricing for energy and reserves: a responsive reserve market proposal," *Decis. Support Syst.*, vol. 40, no. 3-4, pp. 563–577, 2005.

[13] P. Meibom, R. Barth, B. Hasche, H. Brand, C. Weber, and M. O'Malley, "Stochastic optimization model to study the operational impacts of high wind penetrations in ireland," *Power Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –12, 2010.

[14] R. Baldick, "The generalized unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 465 –475, feb 1995.

[15] P. Carpentier, G. Gohen, J.-C. Culioli, and A. Renaud, "Stochastic optimization of unit commitment: a new decomposition framework," *Power Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 1067–1073, May 1996.

[16] C. Murillo-Sanchez and R. Thomas, "Thermal unit commitment including optimal ac power flow constraints," in *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, vol. 3, 1998, pp. 81–88 vol.3.

[17] F. Bouffard, F. Galiana, and A. Conejo, "Market-clearing with stochastic security-part i: formulation," *Power Systems, IEEE Transactions on*, vol. 20, no. 4, pp. 1818–1826, Nov. 2005.

[18] H. Outhred, "A review of electricity industry restructuring in australia," *Electric Power Systems Research*, vol. 44, no. 1, pp. 15 – 25, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/B6V30-3TW320D-3/2/a8ca131ef5cd3be4694b3d40d706c402

[19] G. Shrestha, K. Song, and L. Goel, "Strategic self-dispatch considering ramping costs in deregulated power markets," in *Power Engineering Society General Meeting, 2004. IEEE*, 2004, p. 1146 Vol.1.

[20] R. Doherty, H. Outhred, and M. O'Malley, "Establishing the role that wind generation may have in future generation portfolios," *Power Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 1415 –1422, 2006.

[21] NREL, "Eastern wind integration and transmission study: Executive summary and project overview," EnerNex Corporation, The National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado 80401, Tech. Rep., January 2010.

[22] GEEnergy, "Western wind and solar integration study," The National Renewable Energy Laboratory, Tech. Rep., May 2010.

[23] R. Thomas, C. Murillo-Sanchez, and R. Zimmerman, "An advanced security constrained opf that produces correct market-based pricing," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, July 2008, pp. 1–6.

[24] E. Allen, J. Lang, and M. Ilic, "A combined equivalenced-electric, economic, and market representation of the northeastern power coordinating council u.s. electric power system," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 896–907, Aug. 2008.

[25] T. Mount and A. J. Lamadrid, "Are existing ancillary service markets adequate with high penetrations of variable generation?" in *PES General Meetings*, jul. 2010, pp. 1 –9.

[26] J. W. Guojun Gan, Chaoqun Ma, *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, 2007.

[27] P. Norgaard and H. Hottlinen, "A multi-turbine power curve approach," in *Nordic Wind Power Conference*, 2004, pp. 1–5.

[28] F. Schweppe, B. Daryanian, and R. Tabors, "Algorithms for a spot price responding residential load controller," *Power Systems, IEEE Transactions on*, vol. 4, no. 2, pp. 507 –516, may 1989.

[29] C. Gellings and W. Smith, "Integrating demand-side management into utility planning," *Proceedings of the IEEE*, vol. 77, no. 6, pp. 908 –918, jun 1989.

[30] J. Y. Mo, "Economic analyses of plug-in hybrid electric vehicles, carbon markets and temperature-sensitive loads," Ph.D. dissertation, Cornell University, 2011.

[31] T. Mount, S. Maneevitjit, A. Lamadrid, B. Thomas, and R. Zimmerman, "The hidden system costs of wind generation in a deregulated electricity market," *The Energy Journal*, vol. 33, no. 1, pp. 161–186, 2012.

[32] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12 –19, 2011.

[33] H. Wang, C. Murillo-Sanchez, R. Zimmerman, and R. Thomas, "On computational issues of market-based optimal power flow," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1185 –1193, 2007.

# A Homogeneous Group Bargaining Algorithm in a Smart Grid

Mengmeng Yu
Department of Electronic Systems Engineering
Hanyang University
Ansan-Si, Korea
ymm929@gmail.com

Seung Ho Hong
Department of Electronic Systems Engineering
Hanyang University
Ansan-Si, Korea
shhong@hanyang.ac.kr

Min Wei
Key Laboratory of Industrial Internet of Things
Chongqing University of posts and Telecommunications
Chongqing, China
weimin@cqupt.edu.cn

Aidong Xu
Industrial Control Network and Systems Laboratory
Shenyang Institute of Automation
Shenyang, China
xad@sia.cn

***Abstract*—In this paper, we consider a smart grid incorporating numerous devices. In order to manage the energy distribution, we propose an algorithm that employs the group bargaining concept of game theory to efficiently allocate electricity resources among these devices. First, we divide the devices into non-overlapping groups, and allow devices to bargain both within and across groups. Second, we regard each group as homogeneous, so that the bargaining authority can be delegated to a representative device. Thus, the bargaining problem involves only the representatives, each of which bargains with other representatives on behalf of the group it belongs to. Third, we compare various bargaining results via a utility function that indicates the satisfaction level of a representative in terms of the bargaining result. Finally, the algorithm determines the optimal resource distribution pattern by maximizing the aggregated utilities of all representatives. Simulation results show that the proposed algorithm can reduce the complexity of energy distribution by distinguishing between inter- and intra-group bargaining processes, and can efficiently allocate resources to various devices according to their actual requirements.**

***Keywords—smart grid; group bargaining; homogeneous; representative; utility***

## I. Introduction

Electric power systems around the world are being upgraded with a combination of communication, automation, and metering infrastructures, commonly known as a smart grid [1]. Smart grid technology enables demand-side response systems to operate with pinpoint accuracy. At the heart of the smart grid concept is active customer participation to help reduce costs, and at the same time increase the efficiency of the system. Also, real-time pricing information is available to customers to help them reduce or shift the load during peak hours. Smart buildings are practical enablers of the smart grid concept.

Thus far, most existing studies have utilized time scheduling to shift loads to off-peak hours [2]. In our previous work, we developed a system in which different kinds of devices bargain with each other, assigning devices to three main classes based on their priorities and requirements [3]. Non-shiftable devices require an uninterrupted power supply while operating, regardless of energy cost. Examples of non-shiftable loads include office PCs and building alarm systems. Controllable devices, such as lighting and HVAC units, can increase or decrease their demand according to energy costs, by dimming, thermostatic control, and/or fan-speed variation. On the other hand, shiftable loads can schedule their operations during off-peak hours, when energy costs are lower, so that not only are peaks avoided, but also monthly bills are reduced. Examples of shiftable loads include energy storage systems, plug-in hybrid electric vehicles (PHEVs), and water heaters.

Some recent research provides new insight into the resource allocation problem, which employs the idea of Nash bargaining from game theoretic methods. [4] surveyed the papers for game theoretic approach for smart grid. The original bargaining model only considers two players, however, a smart grid system usually comprises a wide variety of devices, and it is difficult to include all of them in the resource bargaining process. In the economic domain, different parties sometimes have to share public facilities, and some of the parties may be groups of individuals, such as labor unions, companies, or entire countries. In some of the existing studies, groups have been treated as single agents, and a theoretical foundation has been laid for the practice of treating groups of individuals as single bargainers. In particular, [5] introduced a bargaining model in which groups of individuals bargain with each other. Based on this model, a Nash bargaining solution was obtained across groups, with the aim of increasing group benefits by forming a union. In the present paper, we adopt this basic group bargaining idea to develop an algorithm for efficiently distributing electricity resources among devices in a smart grid.

This paper is organized as follows. Section II describes the system model and the formulation of the group bargaining game. Simulation results are presented in Section III. Conclusions and future research plans are summarized in Section IV.

This work is partly supported by the GRRC program of Gyeonggi province [GRRC Hanyang 2012-B01, Integrating User Facilities with Smart Grid using USN]; China-Korea International S&T Cooperation Program No.2012DFG12240; the Project of High-end Foreign Experts of State Administration of Foreign Experts Affairs, the P.R. of China (GX20110491004) and the Program of Chinese Academy of Sciences Visiting Professorship for Senior International Scientists.

## II. System Model

### A. A group Bargaining Game based on Representatives

We adopt the group bargaining concept to efficiently distribute electricity resources among the various devices in a smart grid. We assume that these devices "actively" bargain with each other for electricity resources. Since there are always a great many devices in a facility (including building, home and industry), it is not realistic to regard each device as a single entity and include it in the bargaining process. Here, we introduce a group bargaining model, in which devices are divided into non-overlapping groups. In each group, the bargaining authority is delegated to a special device called the representative, which bargains with other representatives on behalf of the group it belongs to. The final bargaining result is obtained by solving a maximization problem, which provides the optimal resource distribution. We describe the formulation of the group bargaining game in *1)* to *3)*.

#### 1) Characterize group preferences

In order to model a group as a single bargaining entity, we must specify its preferences. In real life, bargaining authority is usually delegated to a designated group member called the representative, whose preferences become the group's preferences. The representative then bargains with other representatives on behalf of his own group. In this study, we adopt this idea and select one device as the representative for each group.

#### 2) Homogeneous property

We assume that the bargaining problem is symmetric for all devices in the same group. In other words, all devices in a group have identical bargaining characteristics, and we say that such a group is homogeneous [5]. Combining this property with 1), it is assumed that once bargaining authority has been delegated to a representative, during the subsequent bargaining process, all devices in the group will receive the same resource proportion as the representative. A detailed explanation will be given in subsection *C*.

#### 3) Formulation of the group bargaining game

Consider a smart grid system represented by a set *E* containing *n* devices, and let *E*={{1},…,{*n*}}. In accordance with conventional game theory, we regard each "device" as a "player" in what follows. A utility vector is used to represent the *n* players' utilities, and such a vector is regarded as an element of the *n*-dimensional Euclidean utility space $R^E$ , which is indexed by the *n* players.

By applying some grouping technique based on the theory of coalition formation [6], we can divide *E* into *k* non-overlapping groups, represented by a partition $G = \{G_1, ...G_k\}$ of the set *E*. In this paper, we will not specify how the players are to be grouped, since this is a complex problem in itself, and will be investigated in our future work. Here we simply assume that the players are divided into groups with a group structure *G*, and its feasible utility set *F* is a subset of $R^E$ . Then the bargaining problem can be represented as a tuple (*G*, *F*), consisting of a group structure and a feasible utility set.

Let *j* denote the identifier (ID) of any group ($1 \le j \le k$), and let $i_j$ denote a player belonging to group *j*. Since we assume that each group is homogeneous, any group member can be chosen to act as the representative. Let $i_j^*$ denote the representative for group *j*. Then the original group bargaining game can be transformed into a sub-game between the representatives of the different groups [5].

During the bargaining process, each representative player will try to reserve a resource amount denoted by $r(i_j^*)$ . We let *x* denote the vector composed of the sequence of resources reserved by the representatives, so that $x = \left(r(i_1^*), \ldots, r(i_j^*), \ldots, r(i_k^*)\right)$ . Then *x* is regarded as an agreement for the bargaining game. Here, the agreement concept is derived from cooperative game theory, where players try to collaborate to reach an agreement on how to share public goods, and if the negotiation process fails, disagreement is the result [7]. In our case, the players will share common resources, and a resource division is regarded as an agreement containing a set of resources. Based on the definition of agreement *x*, we denote $i_j^*(x)$ to be the resource amount received by $i_j^*$ instead of $r(i_j^*)$ in the following parts. Let $u(i_j^*(x))$ be the utility of $i_j^*$ , where *u* denotes the utility function, which determines how much payoff $i_j^*$ can obtain during the bargaining process. A detailed description of $u(i_j^*(x))$ will be given in subsection *B*.

Unlike the existing bargaining model [5], here we exclude the concept of a disagreement point. As in our case, we always enforce the baseline that non-shiftable players receive the required resources, and controllable players receive at least the minimum required resources, regardless of peak hours. Obviously, this baseline is also an acceptable result in our model.

A solution of the proposed group bargaining game can be obtained by solving the maximization problem [8]:

$$Max_{u \in F} \sum_{j=1}^{k} u(i_j^*(x)) \qquad (1)$$

the solution being the optimal agreement that maximizes the sum of the utilities of the group representatives..

### B. Defining the Utility Function

In order to model the responses of different representatives to real-time electricity prices, we adopt the utility function concept from microeconomics [3]. Until now, how to define a suitable utility function for demand-response applications has remained an open question [9]. In this research, we assume that each representative will try to adjust its bargaining resource to maximize its satisfaction level, so that an optimal agreement *x* can be reached by maximizing (1). We employ the following quadratic utility function to characterize the satisfaction level of a representative during the bargaining process:

$$u(i_j^*(x)) = N_{G_j}\left(\frac{P_j}{\lambda}\cdot i_j^*(x) - \frac{\alpha}{\beta}(i_j^*(x))^2\right) \qquad (2)$$

$N_{G_j}$ is the number of players in group *j*. $P_j$ is the priority of group *j*, there are *k* groups altogether, and we assume that the sum of all the group priorities is constrained to equal one ($\sum_{j=1}^{k} p_j = 1$). $\alpha$ is the real-time electricity price received from the utility company [10]. $\lambda$ and $\beta$ are parameters defined by the user, determined by specific demands or different times of day, and also help to quantify the utility value.

The utility function given by (2) can indicate the satisfaction level of any representative in terms of the agreement *x*. Moreover, this utility function is concave, and thus as long as the feasible agreements are finite, we can obtain the optimal result by using (1) to check all agreements. The process will be discussed in detail in subsection *C*. The physical meaning of (2) will be discussed in Section III.

### *C. Group Bargaining Algorithm*

Group bargaining can be regarded as happening on two levels: inter-group bargaining and intra-group bargaining [7]. The former occurs between groups, while the latter occurs between the players inside a group. In this paper, we discuss these two bargaining processes separately.

We assume that the system is equipped with an energy service interface (ESI), which serves as the contact point between the utility company and the customer side [11]. The algorithm is to be installed at the ESI, which employs a centralized method of managing the energy distribution for the whole system.

#### *1) Input to the Algorithm*

- Unit price data: The hourly unit price data are received from the utility company through the ESI. An example of a utility that currently provides this type of service is Ameren Illinois Power [10].
- Available budget: The system designer specifies an hourly budget, which facilitates the computation of the total available resources for distribution.
- Group priority: The priority of each group depends upon the group's characteristics and importance. The sum of all the group priorities should be constrained to equal one.
- Device information: The device information includes the class of the device and its required amount of energy during a specific hour. For a non-shiftable device, the required energy could be a fixed value. On the other hand, the required energy for a controllable device can be specified as a range of values, varying from a minimum to a maximum. A shiftable device also has a required energy, which cannot be guaranteed when the energy cost is high. We assume that this required energy can also be specified by a range of values, varying from zero to a nominal resource amount.

#### *2) Output of the Algorithm*

The output of the proposed algorithm is the agreement that maximizes the sum of the utilities of all group representatives, and indicates the optimal resource distribution pattern.

#### *3) Main Steps*

Although player requirements might actually vary from one hour to the next, here we assume that the requirements of each player are the same for each of the 24 hours. Thus, we describe our algorithm on an hourly basis, but not for any specific hour. In the simulation, we will demonstrate a 24-hour resource distribution pattern. The main steps of the proposed algorithm are described below, and a flow chart that summarizes each step is shown in Fig. 1.

**Step 1:** Receive the hourly price and calculate the available resources

The algorithm starts by receiving the electricity price from the utility company, together with the pre-specified hourly budget, and calculates the available resource $S_{ava}$ via the equation

$$S_{ava} = \mathrm{Hourly\,budget}(\$)/\mathrm{Hourly\,Unit\,Rate}(\$/kWh) \qquad (3)$$

**Step 2:** Divide the devices into groups

As mentioned in subsection *A*, we are not yet specifying how the players are to be grouped. Here we simply divide players into groups according to device class (non-shiftable, controllable, or shiftable) and the number of players in each class.

**Step 3**: Select a representative from each group

Since all groups are homogeneous, any player in a group can act as the representative to bargain with other representatives.

**Step 4:** Check all possible agreements *x* based on step size (inter-group bargaining)

Since non-shiftable players have strict resource requirements, they must be satisfied before we consider the other two classes of players, and are excluded from the bargaining process. Thus, only controllable and shiftable representatives will bargain with each other. Note that the required resource can vary from a minimum to a maximum value for a controllable player, and from zero to a nominal resource for a shiftable player. As a result, each agreement *x* is a vector composed of different reserved resources for controllable and shiftable representatives.

During the bargaining process, each representative tries to increase its reserved resource (from the minimum to the maximum required resource, or from zero to the nominal resource) based on the step size $\Delta$. There will be different combinations of values corresponding to different agreements. For each agreement *x*, we calculate the sum of the utilities, as specified in (1), and record the value. If the new sum is greater than the previously recorded sum, the algorithm will go to step 5; otherwise, it will go back to step 4 and check other agreements.

The step size is calculated from the required resource range $[C_{i_j^*}^{\min}, C_{i_j^*}^{\max}]$ of a controllable representative and the nominal resource $C_{i_j^*}^{nom}$ of a shiftable representative, and is also related to the total number of groups *k*. In order to balance the computational time and efficiency, we use half of *k* as the denominator for calculating $\Delta$:

$$\Delta = \min[(C_{i_j^*}^{\max} - C_{i_j^*}^{\min})\Big/ k/2, C_{i_j^*}^{nom}\Big/ k/2] \qquad (4)$$

**Step 5:** Calculate the reserved resources for non-representatives from the representative's proportion (intra-group bargaining)

In this research, the homogeneous property is taken to mean that non-representative players in a group have the same resource proportion as the representative. Take any controllable representative $i_j^*$ as an example. Recall that $r(i_j^*)$ denotes the reserved resource of $i_j^*$, and let $\theta$ be its resource proportion, which is calculated from the required resource range $[C_{i_j^*}^{\min}, C_{i_j^*}^{\max}]$:

$$\theta = (r(i_j^*) - C_{i_j^*}^{\min})\Big/(C_{i_j^*}^{\max} - C_{i_j^*}^{\min}) \qquad (5)$$

Suppose that $i_j$ is one of the non-representative players in the same group as $i_j^*$, and that its required resource range is $[C_{i_j^*}^{\min}, C_{i_j^*}^{\max}]$. Using this value of $\theta$, we can then calculate the reserved resource for $i_j$ from (6).

$$r(i_j) = \theta(C_{i_j}^{\max} - C_{i_j}^{\min}) + C_{i_j}^{\min} \qquad (6)$$

Thus, the reserved resources can be calculated for all non-representatives. This simple procedure is regarded as the intra-group bargaining process in the present research.

**Step 6:** Calculate the reserved resource total $S_{tot}$

For any agreement *x*, after calculating the reserved resources for all non-representative players, we check whether or not the reserved resource total $S_{tot}$ is within the available resource limit. $S_{tot}$ can be calculated from

$$S_{tot} = \sum_{j=1}^{k}\Big(\sum_{\substack{i_j=1 \\ i_j \neq i_j^*}}^{N_{G_j}} r(i_j) + r(i_j^*)\Big) \qquad (7)$$

Note that if $S_{tot}$ exceeds $S_{ava}$, we should discard agreement *x*; otherwise, we temporarily record agreement *x* as the best agreement and go back to step 4 to check other agreements.

There is a special case in which the available resources fail to guarantee that non-shiftable devices will receive the required resources, and controllable devices will receive the minimum required resources. Under these circumstances, we will increase the budget and go back to step 1 to restart the algorithm [4].

**Step 7:** Determine the final optimal agreement *x*

Since each representative has a limited required resource range, and the resource reservation process of step 4 is based on a step size, so that the total number of possible agreements is finite, and we can ultimately arrive at a solution to the proposed bargaining game. This solution will be the optimal agreement obtained by maximizing the sum of the utilities of the group representatives based on (1).

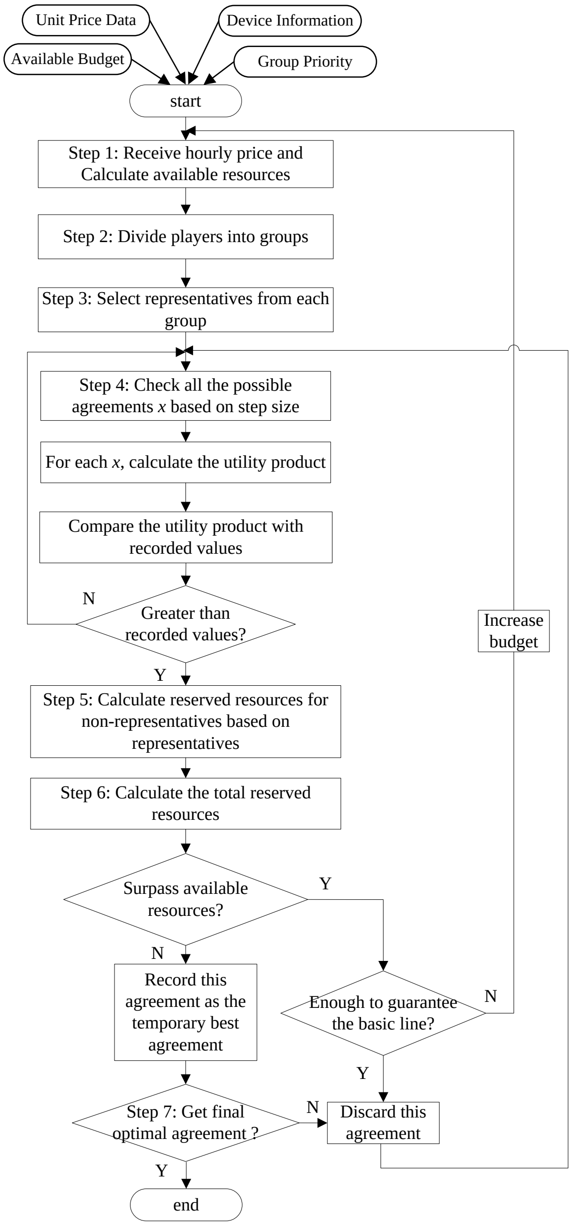

Unit Price Data
Device Information
Available Budget
Group Priority
start
Step 1: Receive hourly price and Calculate available resources
Step 2: Divide players into groups
Step 3: Select representatives from each group
Step 4: Check all the possible agreements x based on step size
For each x, calculate the utility product
Compare the utility product with recorded values
N
Greater than recorded values?
Y
Increase budget
Step 5: Calculate reserved resources for non-representatives based on representatives
Step 6: Calculate the total reserved resources
Surpass available resources?
Y
N
Record this agreement as the temporary best agreement
Enough to guarantee the basic line?
N
Y
Step 7: Get final optimal agreement ?
N
Discard this agreement
Y
end

Fig. 1. Flow chart for the proposed group bargaining algorithm

TABLE I. GROUP INFORMATION

| Controllable | | | Shiftable | | |
|---|---|---|---|---|---|
| Group ID | Devices | Required resource (Wh) | Group ID | Devices | Nominal resource (Wh) |
| Group 1 | **Cj*_1** | 50~150 | Group 1 | **Sj*_1** | 0~100 |
| | Cj_1_1 | 75~125 | | | |
| | Cj_1_2 | 55~145 | | Sj_1_1 | |
| | Cj_1_3 | 65~130 | | Sj_1_2 | |
| Group 2 | **Cj*_2** | 65~125 | Group 2 | **Sj*_2** | |
| | Cj_2_1 | 75~125 | | | |
| | Cj_2_2 | 60~140 | | Sj_2_1 | |
| | Cj_2_3 | 50 ~120 | | Sj_2_2 | |
| Group 3 | **Cj*_3** | 70~130 | Group 3 | **Sj*_3** | |
| | Cj_3_1 | 85~115 | | Sj_3_1 | |
| | Cj_3_2 | 60~140 | | Sj_3_2 | |
| Group 4 | **Cj*_4** | 80~120 | Group 4 | **Sj*_4** | |
| | Cj_4_1 | 55~145 | | Sj_4_1 | |
| | Cj_4_2 | 75~125 | | Sj_4_2 | |

# III. SIMULATION RESULTS

In this section, we present simulation results and assess the performance of our proposed group bargaining algorithm. We first describe the simulation scenario, then explain the key steps, and finally discuss the results of the simulation.

## A. *Scenario*

We chose Matlab as the simulation tool and chose twenty-six devices commonly associated with facility (home, building or industry) including all three device classes (non-shiftable, controllable, and shiftable). Only two non-shiftable devices were chosen, since they are not involved in the bargaining process. We selected fourteen controllable devices and divided them into four groups, two with four devices each, and the other two with three devices each. Similarly, twelve shiftable devices were chosen for the simulation, and divided into four groups of three devices each. To facilitate understanding, each device was assigned the same required resource for each of the 24 hours, although it may actually vary from hour to hour. Table I provides information on the eight groups of controllable and shiftable devices. Because of the homogeneous property, each device had the same bargaining ability within the group, and we simply selected the first device in each group as the representative. The representatives are indicated by boldface type in Table I, and the remaining entries are non-representative devices.

The pricing data was provided by the Ameren Illinois Power Company [10]. As Fig. 2 indicates, low prices, high prices, and peak prices for different hours were included.

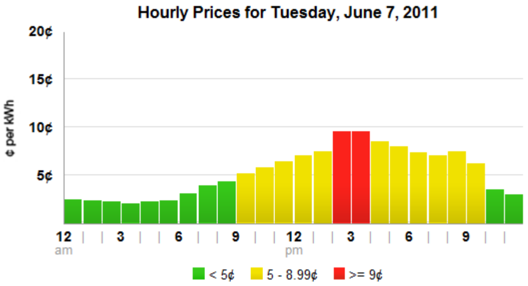


Fig. 2. Hourly price data from Ameren Illinois Power

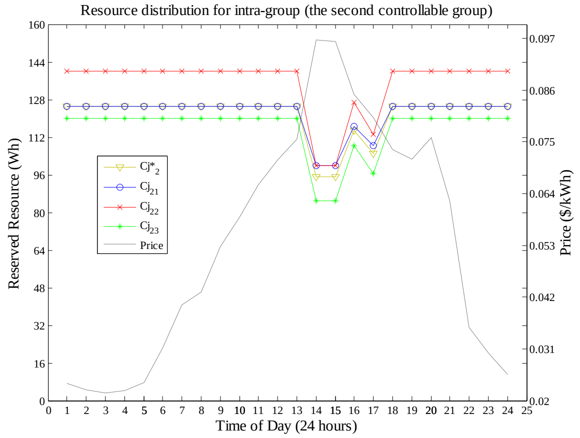


Fig. 3. Intra-group resource distribution

## B. *Simulation Steps*

**Step 1 to 3:** We assumed an average hourly budget of $0.15. After receiving the unit price data from the utility company, we calculated the available resources for each hour. Since we are not yet specifying how devices are to be grouped, here we simply grouped the devices as listed in Table I.

**Step 4:** In this step, we began to compare different agreements by calculating the sum of the utilities. As described above, each agreement is composed of a sequence of resources reserved by the representatives. Before comparing the agreements, we first calculated step size $\Delta$ using (4). From Table I, the representative Cj*_4 of the fourth controllable group was selected, and $\Delta$ was then calculated as $\Delta = (120 - 80)/8/2 = 10$, based on the gap between the minimum and maximum required resources of Cj*_4.

**Step 5:** In this step, if some agreement *x* resulted in a greater sum of utilities, we calculated the reserved resources for the non-representatives. Take the second controllable group as an example. At 5 p.m., device 7 received a resource of 105 watts, and the three non-representative devices received 108.3,

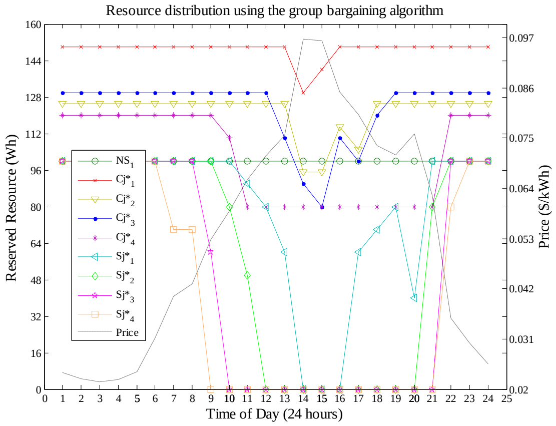


Fig. 4. Resource distribution via group bargaining

113.3 and 96.6 watts, respectively, based on the proportion $\theta$, which was calculated as $\theta = (105-65)/(125-65) = 2/3$. Fig. 3 shows the intra-group resource distribution for the second controllable group.

**Step 6:** In this step, we checked whether or not the reserved resource total, calculated from the agreement of step 5, exceeded the available resource limit. Take 9 a.m. as an example. For one agreement, the reserved resource total was 284.33 watts, which surpassed the available resource limit of 284.04 watts. Thus, this agreement should have been discarded.

**Step 7:** In this step, we determined the final optimal agreement from among all the possible agreements for each hour based on (1). The resource distribution results for 24 hours are shown in Fig. 4 for one non-shiftable device, the four controllable representatives (Cj*_1 to Cj*_4), and the four shiftable representatives (Sj*_1 to Sj*_4).

### *C. Discussion of the Results*

Using the proposed group bargaining algorithm, we efficiently distributed the available resources within and across groups. As Fig. 3 and Fig. 4 indicate, the resource distribution generally operated inversely to the price-changing process, and properly avoided peak-hour consumption.

According to Fig. 4, the representative's reserved resource always decreased later in a higher priority group than in a lower priority group. In real life, people usually prefer to satisfy higher priority devices first, since they tend to play more significant roles in daily lives. This is reflected in (2). Since the utility function is concave and the price is fixed for a specific hour, it is easy to conclude that the more resource a higher priority (greater value) representative reserves, the greater its utility will be. And as we are aiming to maximize the sum utility based on (1), hence, the utility function can guarantee that higher priority representatives (or groups) will acquire more resources than lower priority ones during the bargaining process.

During peak hours (2 p.m. and 3 p.m.), Cj*_1 and Cj*_2 were not reduced to the minimum required resources (as Cj*_3 and Cj*_4 were). Associating with real life, we assume that controllable devices such as air conditioners might not be adjusted to the minimum level during summer. To some degree, the results shown in Fig. 4 closely correspond to real life, and other devices of this type could be assigned to the same group.

At 5 p.m., there was a fluctuation: the resources of Cj*_2 and Cj*_3 decreased a little, while the first shiftable representative began to receive some amount of resource. Due to the proportional intra-group distribution, other devices in this group also received some resources. In daily life, 5 p.m. is usually the beginning of off-duty time, and devices such as PHEVs could be assigned to this group to facilitate people's lives.

Last but not least, through the proposed inter- and intra-group bargaining scheme, our algorithm reduced the complexity of distributing resources to devices by adopting group bargaining idea, and provided some insights into the energy management problem.

## IV. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a group bargaining algorithm to efficiently distribute electricity resources among numerous devices in a smart grid. In the group bargaining model, devices are divided into non-overlapping groups. Because of the homogeneous property, the bargaining authority for each group is delegated to a special device called the representative, which bargains with other representatives on behalf of the group to which it belongs. From among all the possible agreements under given conditions, the optimal resource distribution is obtained by maximizing the sum of the utilities of the representatives. The ideas developed in this paper can be extended in several directions. A sophisticated exploration scheme must be developed to reduce the computational burden. The inter-group bargaining concept could be applied to the intra-group resource distribution process, where each group might be regarded as heterogeneous; for example, each group might contain devices with different priorities. More parameters might be included in the utility function to reveal a variety of daily requirements, such as fairness, cost, discomfort level, and so on. Also, energy storage system could be included to help reducing peak hours based on its flexible operation.

## REFERENCES

[1] C. W. Gellings, The Smart Grid: Enabling Energy Efficiency and Demand Response: Lilburn, GA: Fairmont Press, 2009.

[2] P. Samadi, A. Mohsenian-Rad, R. Schober, V. W. S. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 415-420.

[3] A. U. Haq, S. H. Hong and M. M. Yu, "Efficient DR algorithm for heterogeneous devices in smart grid: A game theoretic approach," IEEE Transactions on Smart Grid, submitted in Energy.

[4] W. Saad, H. Zhu, H. V. Poor, and T. Basar, "Game-Theoretic Methods for the Smart Grid: An Overview of Microgrid Systems, Demand-Side Management, and Smart Grid Communications," Signal Processing Magazine, IEEE, vol. 29, pp. 86-105, 2012..

[5] S. Chae and P. Heidhues, "A group bargaining solution," Mathematical Social Sciences, vol. 48, pp. 37-53, 2004.

[6] K. Apt and A. Witzel, "A generic approach to coalition formation," in Proc. of the Int. Workshop on Computational Social Choice (COMSOC), Dec. 2006.

[7] N. Quérou, "Group bargaining and conflict," in Fondazione Eni Enrico Mattei Working Papers, 2010, pp. 513.

[8] W. Saad, H. Zhu, and H. V. Poor, " Coalitional game theory for cooperative micro-grid distribution networks," in IEEE International Conference on Communications Workshops (ICC), 2011, pp. 1–5.

[9] F. Zhong, "Distributed demand response and user adaptation in smart grids," in IFIP/IEEE International Symposium on Integrated Network Management (IM), 2011, pp. 726–729.

[10] Ameren Illinois Power. Power Smart Pricing. Available: http://www.powersmartpricing.org/about-hourly-prices

[11] "Facility Smart Grid Information Model," in Advisory Public Review Draft, ed: ASHRAE, 2012.

# Model-Based Evaluation of GPS Spoofing Attacks on Power Grid Sensors

Ilge Akkaya, Edward A. Lee, Patricia Derler
University of California at Berkeley
{ilgea, eal, pderler}@eecs.berkeley.edu

*Abstract*—**Emerging cyber-physical system (CPS) applications require reliable time synchronization to enable distributed control and sensing applications. However, time reference signals are vulnerable to attacks that could remain undetected for a long time. Sensor-rich distributed CPS such as the "smart grid" highly rely on GPS and similar time references for sub-station clock synchronization. The vulnerability of time synchronization protocols to spoofing attacks is a potential risk factor that may lead to falsified sensor readings and, at a larger scale, may become hazardous for system safety. This paper describes a simulation-based assessment of the effect of time accuracy on time-centric power system applications. In particular, the vulnerability of power grid sensors to erroneous time references and the potential risks of time-base spoofing on power grid health are studied, using the Ptolemy modeling and simulation tool. Both the false alarm and the missed generation scenarios are considered, where the GPS spoofer may lead the substation to declare an erroneous out-of-phase situation, or the substation may be disabled to detect anomalies that are present in the incoming phase data.**

## I. Introduction

Cyber-physical systems are becoming increasingly complex due to the growing number of components in the next generation distributed CPS and the climb in the sensor data rates for precise control and monitoring applications.

In many CPS applications, due to increased sampling rates at sensors and the need to aggregate data from multiple nodes that are possibly operating at distant locations, the accuracy of component clocks has become a point of concern. Time synchronization protocols such as PTP [1] are being widely deployed for precise synchronization of substation clocks to a master time reference. However, the vulnerability of systems against time synchronization attacks is still a concern in many systems including Unmanned Air Vehicles (UAV) [2] Audio-Video Bridging [3], automotive and power grid applications.

The power grid is a large scale CPS, which relies on GPS time synchronization for time-alignment of spatially distributed sensor data. It is predicted that over 100 million sensors and meters will be present in the future power grid [4]. Installation of high-throughput precise-time phasor measurement units (PMUs), also known as *synchrophasors*, into the grid infrastructure has enabled the acquisition of time-synchronized measurements of state variables at electrical nodes in the transmission network. Requirements on time-synchronization and clock precision at local substations has subsequently become a point of attention, since the benefit of the time-stamped data for real-time control and detection purposes is directly determined by the integrity of the measurements.

The wide accessibility and high precision of GPS signals have promoted GPS time synchronization as a trusted wireless clock synchronization technique for synchronized sensor devices. However, civilian GPS signals are susceptible to spoofing, putting numerous safety-critical sensors at risk of producing unreliable data, while remaining undetected by the target platform over long periods of time [5].

In the case of large distributed CPS, pre-deployment modeling and simulation is a desirable method for assessing protocols and infrastructures. Time synchronization is a well-fit simulation problem, since testing against time-base spoofing attacks in practice require deployment of complex equipment and more importantly, it is an extremely time-consuming process [6]. Spoofing attacks usually require time commitments in the order of millions of seconds, causing pre-deployment spoofing tests to become extremely difficult, if not impossible.

In this paper, we present a simulation-based assessment of the effect of time precision on PMU data streams and evaluate potential risks of erroneous time references on power grid health. As a case study, we quantify the vulnerability of PMU readings under two GPS spoofing scenarios, which may either trigger false generator trips or conceal existing phase angle deviations in the power grid to cause potential grid instability.

## II. Related Work

Recent research has shown that GPS receivers in many sensor devices, including synchrophasors are vulnerable to GPS time-base spoofing attacks [2]. Experiments have indicated that following a 10-15 minute take over period required for the PMU receiver to be completely captured by the spoofer, it is possible to drift the time reference of the PMU local clock in the order of tens of microseconds in several minutes, causing the phasor measurements to become entirely unreliable.

Some countermeasures against GPS spoofing have been proposed and experimented [7], [8]. However, systematic clock manipulations that are in comparable rates to the local clock jitter are drastically difficult to detect. Simulation-based assessment of such scenarios is therefore essential for pre-deployment evaluation of potential security risks.

## III. Modeling Time in Cyber-Physical Systems

Time is an ambiguous notion for a cyber-physical system. CPS consist of distributed physical and computational components that have hardware and software clocks that could either be synchronized to a master clock or be running in standalone mode. This variation of components combined with the physical clock imperfections such as clock drift induced by temperature and vibration gives rise to the need of more detailed clock implementations in CPS simulation.

Ptolemy is a modeling and simulation tool that is widely used for heterogeneous system design [9]. Ptolemy provides support for modeling different notions of time at different components of a composite model. Every level of hierarchy in a Ptolemy model has a `director` that governs the execution semantics of the sub-model according to a model of computation. Each director has a *local clock* that keeps track of the model time within the sub-model. Local time is related to the time of the enclosing model (environment time) via a parameter called `clockRate`. The global time resolution of the model is also a user-defined parameter, which should be adjusted according to the order of magnitudes of the clock parameters.

A clock rate of `1.0` at the sub-model indicates that model time advances in exactly the same rate within the sub-model as in the enclosing director. In many CPS applications, this will not be the case. Platform clocks in general have offsets from the global time reference (UTC, GPS) and have crystal imperfections that cause the local clock to drift. This effect can be encapsulated within the multiform time enabled by the Ptolemy model.

In a CPS model, the top level time is referred to as *oracle time*. This can be considered a global time reference for the model and is not an actual physical quantity.

### A. Clock Imperfections

Although the `clockRate` parameter in Ptolemy models provides a means to model different clock rates, clock imperfections still need to be modeled to account for random jitter in the oscillator.

In Ptolemy, Discrete-Event(DE) model of computation is the most natural model to represent the ”cyber” behavior in a CPS, due to the discrete nature of computation and communications. A sub-system with DE semantics has a local clock that runs at a relative speed to its environment. Using a discrete clock within this sub-model will directly enable simulating ticks produced by an imperfect oscillator.

To introduce random oscillator deviations from the clock rate, we use an additional synchronization block. One example of such component is given in Figure 4, called `Noise/Drift Generator`. The `Noise/Drift Generator` component has the following parameters

1) `syncPeriod`: The synchronization period to the master clock
2) `freeRunDrift`: The part-per-million (PPM) clock drift, denoting the drift rate of the oscillator in free run mode
3) `oneSigmaNoise`: The one sigma positive or negative time excursion from the mean clock rate
4) `impairmentPeriod`: Period of the sawtooth clock deviation.

### B. Modeling Time Synchronization

Components of a distributed system usually share a common reference of time through synchronization protocols. In the most general case of physical systems, the reference time can be UTC time. PTP [1] and NTP [10] are common time synchronization protocols used for synchronization of distributed sub-station clocks. It is also common practice to use the direct GPS signal to discipline a local oscillator.

GPS clock synchronization is a common method of time synchronization for synchrophasors. Most commercially available PMUs contain a GPS disciplined oscillator (GPSDO) for maintaining the local clock at the PMU substation. A 1 pulse-per-second (PPS) broadcast GPS clock signal is used to discipline the local oscillator.

In Figure 4, the `Noise/Drift Generator` component performs clock synchronization by comparing the local time to the received master clock time and adjusting the local clock rate accordingly. This base mechanism may be configured to implement any given well-defined clock synchronization protocol. In the following sections, this component will be used to synchronize sub-station times to the GPS clock reference.

## IV. Time Synchronization in the Power Grid

The ”smart grid” is incorporating sensor devices capable of providing precise-time high quality measurements of the grid variables. These sensors depend on precise time measurements to be able to produce reliable data.

One prominent example of a sensor that requires extremely precise time alignment is a phasor measurement unit (PMU). PMUs are sensors that perform synchronized real-time measurements of the grid state (voltage, current, phase). The local clock of the PMU must be synchronized to a global time reference with relatively low deviation rates. Considering that a $1\mu s$ deviation in the local clock causes a $0.021°$ phase detection error, it is important to maintain local clock rate synchronized to the global time reference, on average. Since all generators in a grid segment must operate ”in phase” within a fraction of $1°$, it becomes essential to ensure that the local clock of the PMU satisfies the precision requirements of power grid applications.

### A. Time Synchronization for Signal Processing Accuracy

The PMU includes a signal processing block for the synchrophasor estimation of the phase of the power signal. The input signal (voltage or current) is sampled simultaneously with two local quadrature oscillators that have a $90°$ phase shift, constituting the local phasor reference. Complex multiplication of the input phasor with the local reference followed by low pass filtering yields the phase angle of the input signal Consider a reference input voltage signal at a reference node in the power grid, given by

$$v(t) = cos(2\pi f_n t + \theta(t)) \qquad (1)$$

where $f_n$ is the nominal grid frequency that is 60 Hz for the US grid, and $\theta(t)$ is the phase at time t. The magnitude is assumed to be unity for simplicity. Time in the physical system, denoted by $t$, advances at a rate denoted by a reference time, i.e. GPS. Consider the sampling of $v(t)$ by a PMU substation. A sample of the signal at GPS time $t_0$ will have the value $v(t_0) = cos(2\pi f_n t_0 + \theta(t_0))$. The following derivations consider the amplitude detection is performed with negligible error during sampling.

If the local oscillator of the PMU is not perfectly synchronized to the reference physical clock, which is the common case, there will be an offset at the platform time of the PMU at the time of sampling. We denote the platform time as $\tau$. Assume that at the time of sampling, platform time is related to the GPS time by the equation $\tau = t_0 + \epsilon(t_0)$, where $\epsilon(t_0) < 0$ for a platform time that lags behind GPS time. Note that the offset in the platform time will cause the local phasor reference to be produced at a different GPS time and in turn, the GPS time representation of the signal will have the form

$$v_{PMU}(t) = cos(2\pi f_n \tau) + jsin(2\pi f_n \tau) \qquad (2)$$

The process of phase angle detection as outlined above, requires complex phasor multiplication of the local phasor reference $V_{PMU}$ with the signal $v(t)$, represented as

$$V_0 = e^{i\theta(t_0)}$$

in phasor notation at the time of sampling, with a reference angular frequency $\omega = 2\pi f_n$ and GPS time reference t. The PMU phasor using the same reference will then be

$$V_{PMU} = e^{i2\pi f_n \epsilon(t_0)}$$

where the error in platform time reference appears as a phase term in the PMU phasor. Complex multiplication followed by low-pass filtering of the high-frequency terms will yield the phase estimate at $t_0$ to be

$$\tilde{\theta}(t_0) = \theta(t_0) - 2\pi f_n \epsilon(t_0) \qquad (3)$$

The absolute phase angle error is given by $|\hat{\theta}(t_0)| = |\tilde{\theta}(t_0) - \theta(t_0)| = 2\pi f_n \epsilon(t_0)$. This error is purely induced by the local clock deviation. If synchronization error is random around the nominal value, there is no systematic deviation in the phasor measurements. However, a systematic deviation of time from the GPS clock, i.e. by a constant factor, will cause the phase readings to significantly deviate from the actual value, relative to GPS clock.

### *B. Modeling Synchrophasor Measurement Accuracy*

The IEEE Standard for Synchrophasor Measurements for Power Systems states that the synchrophasor measurements must be accurate within 1% of the ”ideal” phase [11]. The deviation from the reference is measured by a quantity called *total vector error* (TVE) that aggregates errors that could happen in *amplitude* and in *phase* during phasor estimation.

Assuming a perfect amplitude estimate, a phase error of 0.57° will cause a 1% TVE, which corresponds to an error of approximately 26.39 $\mu s$ in the time reference [11]. While this upper bound implies certain requirements on the local accuracy of the PMUs, imperfect clock references can lead to the violation of the standard [12].

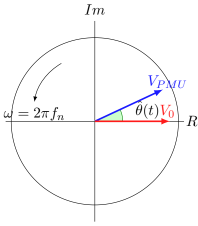


Fig. 1: Phasor representation of phase detection process

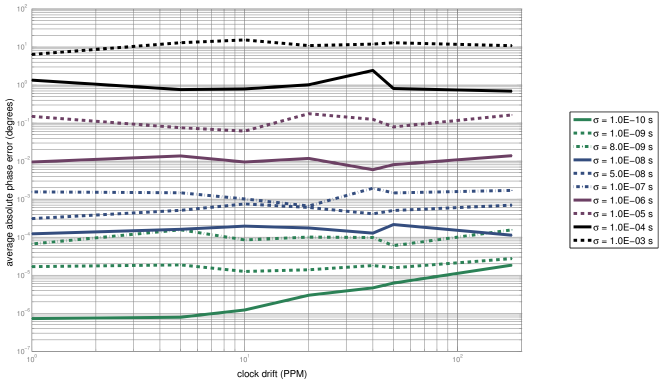


Fig. 2: Effect of Clock Precision on the Phase Estimation Accuracy

Before we assess the effect of GPS time-base spoofing on PMU operation, we initially study the stability of local PMU clock and its effect on phase detection performance. We consider a model-based approach to simulate the average absolute phase error caused by oscillator imperfections. A synchrophasor model given in Figure 4 is used to model the behavior of a time-synchronized Phasor Measurement Unit. By setting `free run clock drift` and `one-sigma noise` on the clock jitter, it is possible to simulate physical oscillator characteristics in a consistent way.

Figure 2 shows the correspondence between one sigma clock noise and phasor measurement accuracy. The tests are carried out using a 1 PPS GPS pulse for time synchronization. The phase angle error is calculated by averaging the phase difference measured by a synchrophasor with an imperfect oscillator and a reference synchrophasor set to have zero clock drift and noise, over 2 seconds, at a sampling rate of 900 Hz.

Analysis of Figure 2 reveals that the maximum allowable phase error of 0.57° is only achievable with clock jitter within tens of microseconds, under normal operating conditions. The two test cases with one-sigma noise of 0.1 ms and 1 ms were shown to have an average phase error of more than 1°, which violates the IEEE Standard. Commercially available PMUs are currently adopting more precise GPS clock receivers to overcome this challenge [13].

## V. Assessment of the Effect of GPS Spoofing on Synchrophasor Measurements

GPS spoofing is a security threat that has been successful in causing erroneous data references in several commercial GPS based systems [14], [15]. Increasing dependence on GPS synchronization has caused the power grid to become a vulnerable candidate for such spoofing attacks. PMUs with individual GPS receivers that receive direct civilian GPS signals to discipline their local clock are a natural point of exposure to time-base spoofing attacks.

In a simple scenario, at a reference bus in the power grid, we consider two co-located Phasor Measurement Units (PMUs) measuring the power signals at the same electrical node on the power system. [12] has shown that a GPS spoofer located at some proximity of the GPS receiver of the PMU can successfully take over the GPS receiver and carry off the time reference at the substation by an arbitrary amount in time to cause false power flow and phase angle readings at the node.

We will additionally explore a more sophisticated scenario, where the spoofer is also assumed to be able to access the phase measurements from the unaffected PMU at the same node. This scenario is particularly realistic for PMUs that send data wirelessly to a data concentrator for further data aggregation and analysis. In this case, it is also possible to induce a non-constant time deviation to drive the phase measurement in an arbitrary manner from the actual value, with a purpose to cause missed detections of grid disturbances. This scenario is potentially much more hazardous for the power grid health, since it is likely to delay or completely disable the detection of certain anomalies.

A model-based approach for studying possible GPS spoofing attacks and their effect on system stability has numerous advantages. In a simulation environment, it is convenient to specify desired clock characteristics for systems and experiment seamlessly with interchangeable models of sub-stations.

### A. Inducing False Alarms in the Power Grid

We use the top level Discrete Event model presented in Figure 3 for investigating a spoofing attack to induce false alarms due to erroneous time reference at the local substation. The top level model has a local clock that advances at a rate of 1.0, in the global time reference for the entire model, called *oracle time*. The model includes a `GPS Transmitter`, that is assumed to have a clock that advances synchronously with oracle time. There are two `Synchrophasor` components with identical local clock characteristics (clock jitter and clock drift). The `GPS Synchrophasor` is assumed to have a local clock synchronized to the 1 PPS GPS signal. The `Spoofed Synchrophasor`, however, models a synchrophasor captured by the spoofer unit `GPS Spoofer`, which first manipulates the actual GPS time reference and delivers an erroneous master clock reference to the synchrophasor.

The details of the `Synchrophasor` implementation are given in Figure 4. The functional flow of this component consists of time synchronization to the master clock reference followed by the signal processing unit to estimate phase angle, which is outlined in Section IV-A.

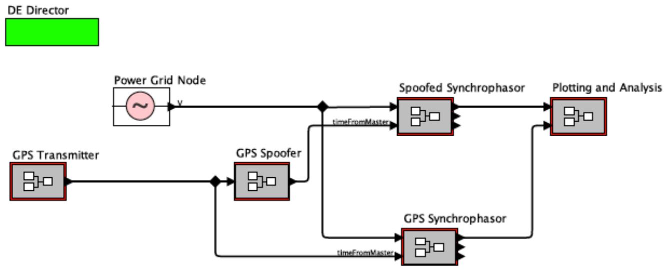
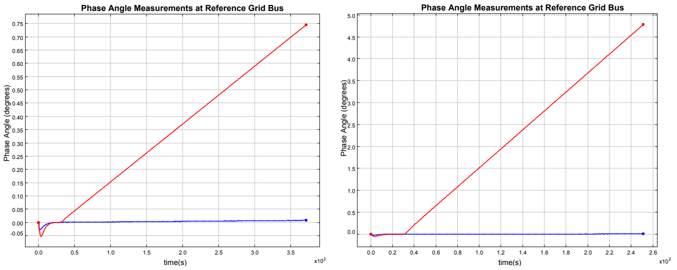


Fig. 3: Top Level Ptolemy Model for the GPS spoofing setup



(a) $\epsilon(t) = 0.1\mu s$ (b) $\epsilon(t) = 1\mu s$

Fig. 5: Phase Angle Measurements from two PMUs located at the same node of the power grid. [red: PMU synchronized to spoofer clock, blue: PMU with perfect time reference

The spoofer is assumed to have captured the GPS receiver of the compromised PMU at the beginning of simulation. After time synchronization to the master clock ( from `GPS Spoofer`) has been established, the spoofer deviates its reference clock rate from the GPS reference at a rate of $\epsilon(t)$ per second. For $\epsilon(t) = 0.1\mu s$, this corresponds to a deviation of 0.1 PPM.

Figure 5 demonstrates the phase measurements from the `Spoofed Synchrophasor` plotted in oracle time, for two clock deviation rates. For a deviation rate of $0.1\mu s/s$, the C37.118.1-2011 standard is violated at t = 290.7 s, that is, approximately 260 s since the beginning of the spoofing attack. For a more aggressive attack with $\epsilon(t) = 1\mu s/s$, the standard violation occurs shortly before the 60 s mark, approximately 30 s after the beginning of attack.

### B. Time-Stamp Manipulation to Mitigate Existing Phase Faults in the Grid

With the existing model for the synchrophasor-GPS communications, we next investigate a more complicated spoofer model that has access to the readings from the non-spoofed `GPS Synchrophasor` readings with some feedback delay. The model for this test is given in FIgure 6. The modification in this model variant is that, the spoofer clock has access to the phasor readings from the non-spoofed synchrophasor at the same node, which is used to modify the time reference with the aid of a PID controller, to force the phase reading of the spoofed synchrophasor towards zero.

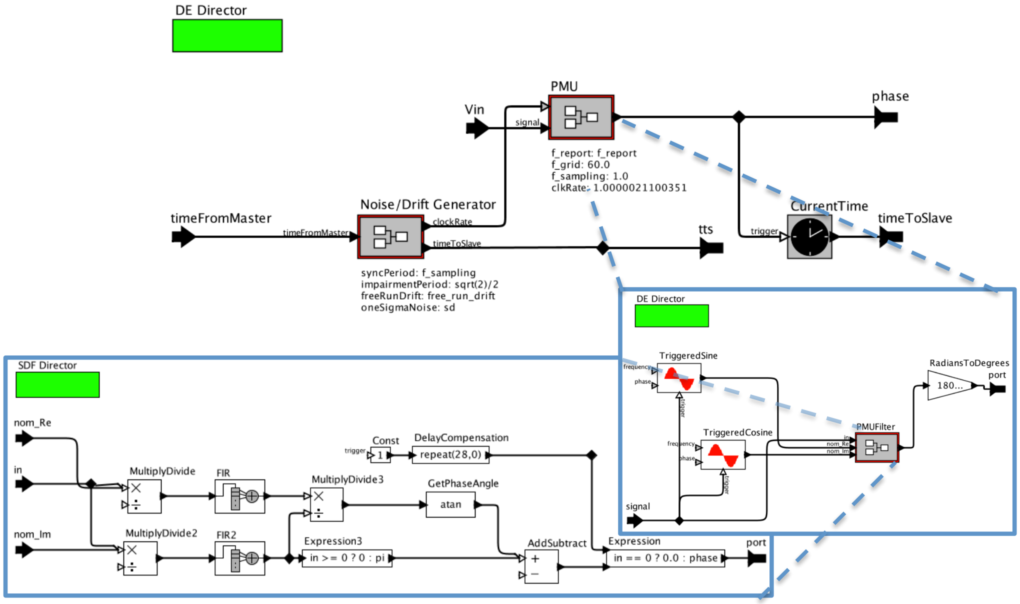


Fig. 4: The Synchrophasor Model

A worst-case feedback delay of 10 samples is modeled on the phase-feedback connection to the spoofer. At a sampling rate of 900 Hz, this roughly corresponds to a network delay of 11.1 ms.

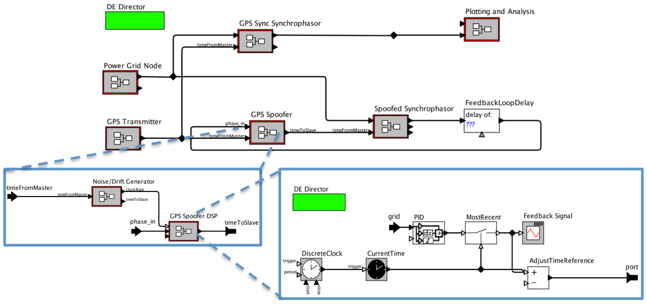


Fig. 6: Detailed Model for Phase-Feedback GPS Spoofing

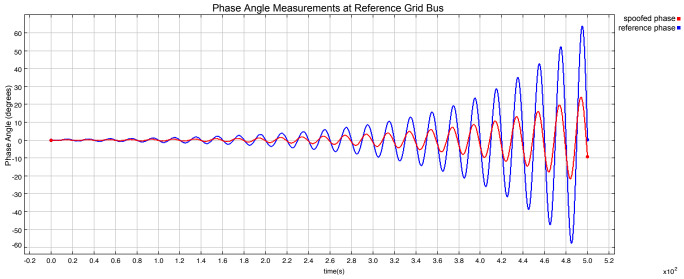


Fig. 7: Time-base spoofing attack for delaying out-of-phase tripping action for an unstable harmonic phase disturbance

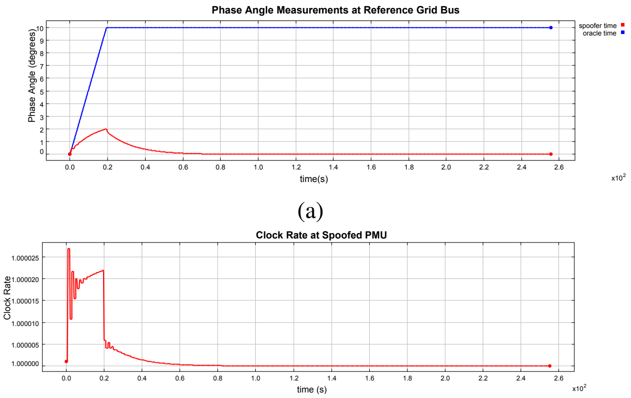
A commonly observed grid disturbance is a growing sinusoidal phase harmonic that may eventually cause the responsible generator to be tripped, to maintain grid stability. This kind of disturbance is usually detected within several seconds and remedial action is taken [16]. The spoofing model with feedback capability is used to simulate one such disturbance in the grid. Figure 7 shows the simulated waveforms of the measured phase angle under such condition. The blue waveform denotes the phasor estimates performed by the healthy synchrophasor and the red waveform corresponds to the phase estimate of the spoofed platform. It can be seen that detection of the harmonic disturbance can be significantly delayed due to the time-base spoofing attack. In this scenario, it is not possible to completely cloak the presence of the disturbance, since we assume a 1PPS synchronization signal, as well as some feedback delay due to a realistic network infrastructure.

Table I presents a quantitative evaluation of the delay in detection caused by the spoofer at the protective relay. For 1° allowable phase margin relative to nominal grid phase angle (0°), the detection of the harmonic disturbance is delayed by approximately 90 seconds. During the delay, the phase angle at the node has exceeded the maximally allowed deviation by 2.3°.

| Phase Error Tolerance(°) | Nominal Tripping Time (s) | Delayed Tripping Time (s) |
|---|---|---|
| 0.20 | 11.29 | 12.80 |
| 0.50 | 14.11 | 104.07 |
| 1.00 | 84.05 | 173.60 |
| 2.00 | 153.95 | 244.00 |

TABLE I: Phasor Measurement Spoofing to Cause Delayed Detection of Out-Of-Phase Tripping for an Unstable Harmonic Phase Disturbance

Another disturbance pattern that may be observed in power grid nodes is a ramp deviation from the nominal grid phase an-



(a)

(b)

Fig. 8: Time-base spoofing attack for delaying out-of-phase tripping action for a ramp phase disturbance

gle that settles at a steady-state angle. Figure 8a demonstrates a ramp-phase deviation with a 10° steady-state value, and the corresponding spoofer action to suppress and then mask the disturbance pattern completely. The experiments reveal that in less than 220 s, the disturbance is entirely hidden by the spoofed time reference, making the disturbance virtually undetectable by the synchrophasor, with an overshoot of 2°. Once the time offset at the synchrophasor local clock has been established by the GPS spoofer, the steady-state clock rate deviation needed to conceal the phase disturbance becomes zero, as demonstrated in Figure 8b.

## VI. Conclusion

We presented a simulation based evaluation of potential implications of imperfect time references in cyber-physical systems. We studied GPS spoofing attacks and their possible effect on PMU data quality. It was shown that, under certain circumstances, GPS spoofing may lead to missed detections of phase disturbances in the grid and long-term coordinated attacks may even lead to severe consequences, such as cascading blackouts and damage to equipment.

Future work includes investigating alternative time synchronization techniques and modeling security counter-measures for grid components. Additionally, the model-based assessment technique will also be beneficial in modeling architectures that rely on time-synchronization between sensor platforms for systems in which not all sensor devices may have access to the global time reference. This scenario is particularly interesting to demonstrate error propagation and respective cascading faults in the grid.

## Acknowledgement

This work is supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

## References

[1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2002*, pp. i –144, 2002.

[2] D. Shepard, J. Bhatti, T. Humphreys, and A. Fansler, "Evaluation of Smart Grid and Civilian UAV Vulnerability to GPS Spoofing Attacks," 2012.

[3] D. Pannell, "Audio video bridging gen 2 assumptions," March 2012. [Online]. Available: http://www.ieee802.org/1/files/public/docs2012/avb-pannell-gen2-assumptions-1203-v9.pdf

[4] P. A. Craig and T. P. McKenna, "Technology security assessment for capabilities and applicability in energy sector industrial control systems," PNNL - 21313, Tech. Rep., 2012. [Online]. Available: http://www.mcafee.com/us/resources/reports/rp-energy-sector-industrial-control.pdf

[5] T. Humphreys, "Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil GPS spoofing," 2012.

[6] E. Lee, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*. IEEE, 2008, pp. 363–369.

[7] B. O'Hanlon, M. Psiaki, T. Humphreys, and J. Bhatti, "Real-time spoofing detection in a narrow-band civil gps receiver," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, 2001, pp. 2211–2220.

[8] J. Warner and R. Johnston, "Gps spoofing countermeasures," *Homeland Security Journal*, 2003.

[9] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong, "Taming heterogeneity - the ptolemy approach," in *Proceedings of the IEEE*, 2003, pp. 127–144.

[10] Network time protocol. [Online]. Available: http://www.ntp.org

[11] "IEEE standard for synchrophasor measurements for power systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1 –61, 28 2011.

[12] D. Shepard, T. Humphreys, and A. Fansler, "Evaluation of the vulnerability of phasor measurement units to gps spoofing attacks," *International Journal of Critical Infrastructure Protection*, 2012.

[13] Model 1133A Power Sentinel Phasor Measurement Unit. [Online]. Available: http://www.arbiter.com/catalog/product/model-1133a-power-sentinel.php#tabs-2

[14] J. Warner and R. Johnston, "A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing," *Journal of Security Administration*, vol. 25, no. 2, pp. 19–27, 2002.

[15] N. Tippenhauer, C. Pöpper, K. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 75–86.

[16] E. Schweitzer, D. Whitehead, A. Guzman, Y. Gong, and M. Donolo, "Advanced real-time synchrophasor applications," in *35th Annual Western Protective Relay Conference, Spokane, Washington, USA*, 2008.

# A Framework for Model-Based Design of Embedded Systems for Energy Management

Javier Moreno Molina, Xiao Pan, Christoph Grimm, Markus Damm
University of Kaiserslautern
Email: {moreno|pan|grimm|damm}@cs.uni-kl.de

***Abstract*—Model-Based Design of Cyber-Physical Energy Systems (CPES) is a challenge from a modeling and simulation point of view. Multi-domain and multi-scale modeling and simulation as well as high simulation performance are required in order to model distributed systems, appliances, embedded systems, electric components, and physical systems of different nature at different levels of abstraction.**

**In this paper we describe a framework, based on SystemC, for the model-based design of embedded HW/SW systems for distributed energy management applications in buildings and neighborhoods. These embedded systems are included in smart appliances, that are capable of gather information, control the appliance and communicate with the network. Communication (wireless, PLC) is modeled using TLM extensions, in order to achieve high simulation performance. On the other hand, physical domains are modeled using AMS extensions.**

**For demonstration, we model, simulate and evaluate the performance of an in- house energy management system.**

## I. Introduction

Energy has become a resource of particular significance. The transition from a centralized power distribution grid to an approach with distributed management and distributed feed-in by small renewable energy resources is addressed by the *Smart Grid* approach. In a Smart Grid, the distribution grid is enhanced with embedded systems that act as IT infrastructure for managing generation and feed-in. Such energy management can for example balance consumption of appliances or electric vehicles and volatile generation of photovoltaics.

*Smart appliances* are able to decide when to consume power in an autonomous way based on user's inputs and the state of the power grid. For this purpose, smart appliances need sensors to get the internal state, and knowledge of the state of the grid. A *gateway* networks smart appliances and runs methods for local energy management. Furthermore, it communicates with management of the (local or global) distribution grid. However, this sketched field is rather a research topic; practical experiences in larger scale are missing.

The development of embedded systems and methods for energy management demands realistic scenarios and accurate models. A model-based approach seems to be the most appropriate way, but not without problems. A good question is what 'realistic scenario' means, as there are no established Smart Grids that could be modeled as environment. Obviously, one can assume a particular approach of Smart Grid, and create a model of it; however, what needs to be modeled, what not? Practical experiences of applications are missing. Therefore, modeling and gathering practical experiences using prototypes must go hand in hand – but how?

### A. Objectives and Overview

The objective of this paper is to give answers to the questions raised above, based on experiences gained within the EC FP7 Project SmartCoDe. We give a new approach for model-based design of embedded systems and methods in cyber-physical energy systems. The idea is to use modeling/simulation where appropriate and feasible, and in parallel to apply a prototype-based approach where models are not available or not validated.

In Figure 1 we give an overview of the overall approach: design starts with specification of the functionality and an API to the platform. After this initial specification, methods built on top of the API are developed and evaluated within simulated scenarios, or within physical scenarios using a rapid prototype that implements the API. In parallel to method development, development of the HW/SW platform that implements the API while fulfilling constraints is done.

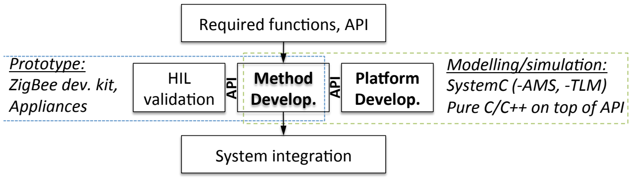


Fig. 1. Overview of approach.

Use of methods or software both in a prototype and in modeling/simulation is best supported by SystemC. Such a C-based approach in particular enables seamless transition from method- and software development to modeling and simulation. To support the above-described design-flow, SystemC has been extended to allow modeling/simulation of

- networking and communication at application layer,
- behavior of smart appliances and its power consumption.

We discuss related work in detail in section II . In section III we describe the methodology in detail. In section IV we give an outline of the simulation framework. Application of the methodology within the SmartCoDe project, experiences during application and performance benchmarks are given in section V.

## II. Related work

### A. Methodology

Approaches to combine prototyping and Model-Based Design are well-known, in particular in the automotive industry. Design teams in charge of different part of the system can obtain valuable information for their design through models.

On the other hand, "Hardware-in-the-loop" (HIL) simulations e.g. [1] allow the validation of methods and assumptions used for the design of embedded systems in a simulated physical environment, when validation in the real physical environment is unfeasible.

Combination of HIL and Model-Based Design approach makes modeling and simulation an extremely powerful and helpful tool in order to achieve better designs with shorter time-to-market and lower cost.

Typically, methods are first developed using model based approach and verified using HIL, and after that embedded platforms are developed using modeling/simulation. In previous work, model-based design is presented as a sequential process, that can in principle be done in parallel (e.g [2]); however, parallelization and its requirements are not discussed.

The methodology we propose in particular deals with *parallelization* of HIL/model validation, method design, and design of HW/SW platform. Note, that we do not outline every single step; instead we focus on the requirements for parallelization. Our approach is enabled and driven by application of platform-based design [3] that leads to layered implementation of SW applications, and well-defined API between layers. Such API in particular allows us to replace a hardware-prototype by a simulation model.

### B. Framework for model-based design

While the parallelization proposed is applicable to embedded systems in general, we focus on Embedded Systems in the Smart Grid. The reason to focus on a particular application domain is that for model-based design typically application-specific capabilities are needed. This includes appropriate models of computation and availability of modeling primitives. In the Smart Grid, the lack, the variability and the complexity of application scenarios make combination of Model-Based Design and HIL approaches an exceptional tool to assist in design and verification.

The requirements for the model based approach are the ability to simulate physical processes, user behavior and interactions with the system, and software and embedded systems at various levels of abstraction and layers of implementation.

A particular challenge is modeling communication among the spatially distributed subsystems, which has become more relevant in recent years [4], [5].

There are three different approaches to deal with these issues:

1) Creating a single framework in which multiple models of computation are supported. A particularly suited framework is Ptolemy II [6] that is the prototypical example of the single-framework approach. Evaluation of the use of Ptolemy II and Simulink/Simscape for CPES has been presented in [7]. Advantage of the single-framework approach is ease-of-use and efficiency. However, it does not directly support distributed and high-performant simulation of networks.
2) Simulator-coupling allows combination of different, very specialized simulators; for CPES in particular Modelica is useful. In [8], a Modelica based energy system model is evaluated, comprising temperature variation and several agents with influence on system behaviour. By coupling Modelica with a network simulator, as done in [9], most of the requirements of CPES are covered. However, a major drawback of the simulator-coupling approach is that it is quite difficult to use, as models have also to provide and to consider simulator coupling interfaces. Furthermore, performance and/or accuracy are limited due the need to synchronize different simulator kernels.
3) Agent-Based modeling/simulation well matches the distributed aspects and can well deal with heterogeneity (e.g. [10]). Unfortunately, such approaches are not yet common in the model-based design of embedded systems.

The framework presented in this paper is based on SystemC [11], and follows the single-framework approach. The main reason to select SystemC is that it is C-based. This allows us to easily switch between running C-based software on a prototype or as part of a simulated scenario. Furthermore, it is widely accepted for system-level design of embedded systems. However, it lacks ability to model wireless networks as it is required for model-based design in the given application domain. In previous work we have shown ability of SystemC to model and simulate energy management methods for fridges in a Smart Grid [12]. In [20] we give an overview of a platform designed applying this approach. In this paper we focus on the overall methodology, its application, and simulation performance in a more complex scenario.

## III. Methodology

### A. Design Flow in detail

The design methodology is shown in more detail by figure 2. Design starts with a simple functional model that focuses on giving an overview of the required functionality. The main objective of this first step is to structure functionality, and to define an API between a higher-level application software and an underlying platform. After API is defined, three activities are done in parallel:

1) Rapid development of a prototype board, and a software that concretely implements the API. After that, this prototype is used for hardware-in-the-loop validation of methods and algorithms.
2) Development of methods and algorithms on top of the specified API in C++.
3) Design of an embedded platform using a model-based approach based on the simulation framework described in section IV. In a first step, system-level partitioning

is done using experiences and data gained by method development. After that, an embedded platform is developed.

The design is finished by system integration, integration validation and deployment.

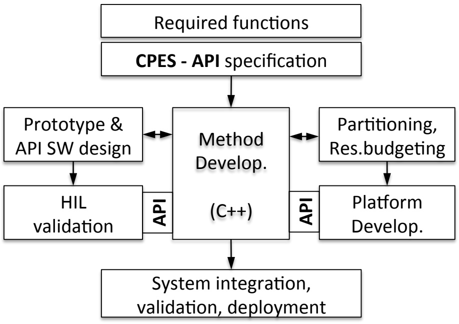


Fig. 2. Design methodology in more detail.

The methodology is supported by the modeling framework which provides means and primitives for simulation of scenarios in Smart-Grids (section IV), and by a mostly generic API between Embedded Systems in Smart Appliances, and methods for energy management realized on top of them.

### B. Generic API for CPES

In Cyber-Physical Systems, we can split the main functionality into two different categories: communication and environment interaction. In addition, there are typically two ways of interacting with the application: actively, as part of the application tasks, or reactively, as incoming events.

In the case of communication, almost all the required functionality is already implemented in protocol stacks. The application can therefore transparently communicate with other nodes. Hence, the minimum functionality is to send and receive messages. Sending messages is actively decided by the application itself. However, the application does not know when a message to be received will arrive. Message reception is usually passive from the application point of view. The received message arrives just as one kind of the possible incoming events.

In the case of the environment interaction we have again both options. Actuation is intrinsically active. Sensing can be reactive, if the sensor controller periodically receives a sensed value, or active, if the sensed quantity is polled by the application.

In order to quickly develop a portable application, all the cases previously described have to be provided as an application API. Therefore we can sum up the API in the following elements:

- Event data structure: Events of different types can be defined.
- Event queue: It stores all the incoming events.
- Application Control Loop: This control loop is in charge of handling the different events from the event queue. When event queue is empty, application can go to sleep. All reactive operations can be performed from this control loop, e.g. message reception.
- Sending Messages: A function to send messages has to be provided so that the application can communicate using the protocol stack.
- Actuation: Two functions have to be provided: a function to register the required actuators and a function to perform the required actuations on the attached appliance. This interface usually selects among several appliance's operation modes or states. Therefore, this function must just select one possible finite value. The application designer must be able later to restrict the number of possible states, to avoid forbidden actuations.
- Active Sensing: As with actuators, two functions have to be provided, one to register the active sensors, and another one to obtain the current sensed values.

However, semantics for M2M (machine-to-machine) communication have to be defined. A promising standard for this purpose are ZigBee application profiles, in this case, the Smart Energy profile. Within the SmartCoDe project we implemented our own research application profile.

## IV. SystemC-Based Framework

This section introduces the framework called SICYPHOS (Simulation of Cyberphysical Systems). The goal of this framework is to enable the use of the methodology described in Section III. The framework is written in C/C++ and uses SystemC for simulation, as well as SystemC's TLM and AMS extensions. The overall architecture of SICYPHOS can be seen in Figure 3.

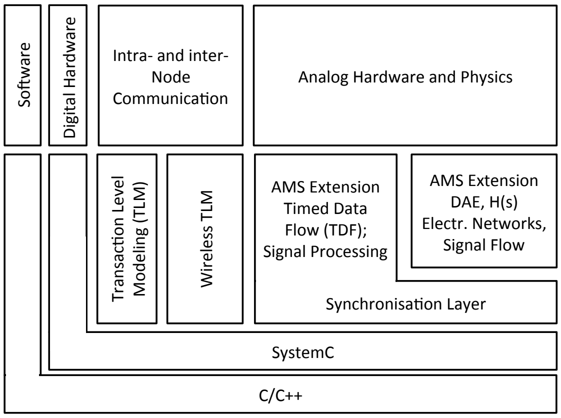

Software
Digital Hardware
Intra- and inter-Node Communication
Analog Hardware and Physics
Transaction Level Modeling (TLM)
Wireless TLM
AMS Extension Timed Data Flow (TDF); Signal Processing
AMS Extension DAE, H(s) Electr. Networks, Signal Flow
Synchronisation Layer
SystemC
C/C++

Fig. 3. Architecture of the SICYPHOS framework

### A. Modeling Networks

A key requirement is the support for modeling wired and wireless communication, as well as different network protocols and architectures. The problem of network and communication modeling is split into two main parts: modeling the channel, and a flexible model of the communication stack.

*1) Channel Modeling:* One of the main problems faced by every network simulator when addressing WSN is the scalability problem. The large number of nodes that are eligible to receive a transmission had a devastating effect in simulation performance [13]. In case of multi-hop wireless communication this becomes a severe constraint. In order to mitigate the simulation performance problem several strategies have been followed:

- Introduction of an intermediate object, which distributes the messages only to the nodes that are likely to receive it, by computing distance and attenuation depending on the given environment properties.
- An abstraction level in communication is introduced. Many existent simulators model every point-to-point communication in a different message data structure. In SICYPHOS, the communication data structure comprises a full end-to-end communication. This means that all messages with the same payload are represented by one single transaction. Hence, the need for allocating new messages while forwarding and broadcasting messages is reduced drastically and simulation performance is significantly improved [14]. However, managing all the point-to-point specific parameters is still necessary. This is achieved by maintaining, in every transaction, maps containing all point-to-point specific parameters.
- With the reduction on memory allocated objects caused by communication abstraction into transactions, there is a need for continuously creating new transactions, but without requiring many of them to be simultaneously in use. This scenario becomes very favorable to achieve even further efficiency by using a transaction pool pattern [15].

*2) Communication Stacks:* In CPES, different communication stacks can be used, either as alternatives or as complementary subnetworks for different parts of the application. Furthermore, there are neither ubiquitous nor mature standards, and therefore research frequently affects the protocol stacks and different applications demand different protocols. As a result, a flexible way to create customizable protocol stacks is a hard requirement for our framework.

The approach selected to provide this functionality has been to leverage the interoperability provided by TLM communication abstraction approach. As shown in Figure 4, every protocol is modeled as a TLM module with two initiator/target socket pairs, one for stack upwards communication (receiving) and another for stack downwards communication (sending). This way, each protocol is implemented independently and protocol stacks can be built just by binding the corresponding sockets.

Protocols specific data, which in real communication is stored in packet headers, can be efficiently stored in the TLM transaction with the use of TLM generic payload extensions.

Nevertheless, for heterogeneous networks, the strict layered communication model is typically enhanced with cross-layer optimization. For this purpose, the framework also provides a cross-layer optimization TLM generic payload extension.

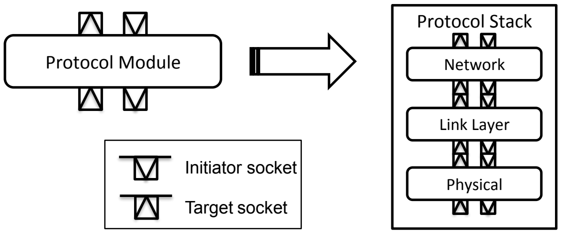


Fig. 4. TLM Protocol Stack

Application dependent extensions or simulation extensions can also be defined. An example of this is the extension to annotate power consumption, explained in Section IV-C.

### B. Modeling Appliances

Although appliances design is out of the scope of our model-based design approach, the behaviour of the different appliances has a significant impact in the Cyber-Physical System, and also has to be evaluated in order to validate the application. Consequently, functional models of CPES appliances are also required in order to realistically simulate CPES applications.

However, modeling appliances, might involve new simulation aspects to be explored. For instance, highly relevant appliances in CPES are those controlling heat, such as HVAC systems or fridges and freezers. These devices work typically by switching on or off in order to keep a measured temperature within some boundaries. Hence, whether these devices are switched on or off depends on the temperature variation. Modeling that variation involves physics modeling which is out of scope of most common network and system level simulators.

In order to incorporate physics simulation to our SystemC based framework, SystemC-AMS extensions have been used. These extensions include several Models of Computation (MoCs) to simultaneously simulate multi-disciplinary models [16]. Currently provided MoCs include Timed Data Flow (TDF), which provides structures for discrete time modeling, and Linear Signal Flow (LSF), which provides continuous time simulation infrastructure. Furthermore, the extensions also supply the interfaces between the different MoCs.

For the HVAC example, which is relevant for the CPES case, temperature variation can be efficiently modeled using TDF MoC. Room temperature models exist which estimate temperature variation by using an equivalent electrical circuit model, where temperature variation behaves like voltage variation through a capacitor. The losses due to non-ideal insulation behave like a thermal resistance [17] [8].

In our SystemC-AMS TDF approach, temperature variation is modeled as a low-pass filter, whose output can be easily computed using the Laplace Transfer Function (ltf) functionality included in TDF MoC.

### C. Modeling power consumption of embedded hardware

An important design constraint for embedded systems in smart appliances is standby power consumption. As the number of nodes grows, and might even embrace single light bulbs, standby power consumption becomes more and more significant. Since one of the goals of energy management is energy efficiency, achieving it on the embedded hardware itself is imperative. During design, impact of methods, communication protocols, and architecture-level partitioning has to be evaluated. Consequently, estimating the energy consumption is also a major requirement for the Model-Based approach.

*1) Power-State Machines:* The framework provides structures to create finite state machines to account power consumption. Every state is therefore associated to a power consumption pattern. State transitions can be automatically logged and energy consumption can be calculated then based on the spent time estimated at each state.

*2) Transaction Power Annotation:* In order to optimize power consumption through the network, awareness about the impact on other nodes energy consumption, of every communication, becomes a very powerful tool. This is provided by the framework by leveraging the transaction concept introduced in Section IV-A. The energy consumed on each node while handling a transaction can be annotated on it, obtaining not only information of energy consumption from the hardware point of view, but also from a communication perspective. This energy profiling technique permits optimization in network topology and protocol stack [18].

### *D. Modeling Application*

One of the major uses of this framework is the development and evaluation of applications. For this purpose, the framework offers a double-layered API.

*1) General Purpose API:* The first layer API is an abstraction layer for general purpose, offering basic functionality which is common to every CPS application, e.g. sending, receiving, sensing, actuating (see Section III-B).

*2) Application Profile API:* The top layer API is based on the application profile concept used in protocol stacks, e.g. ZigBee. This API has a narrower application scope and is built on top of the General Purpose API. In the case of CPES, the framework provides the SmartCoDe Application Profile API, developed during SmartCoDe project [12] [19]. Other application profiles, such as Home Automation or Smart Energy ZigBee profiles are planned to be provided.

With these APIs, developed applications make use of the same infrastructure implemented in the final systems. By using just the basic C language supported by the final embedded system, application code portability is achieved and hardware/software co-design improves. Actual portability has been validated during the SmartCoDe project, where the application developed during simulation was successfully ported to the NXP/Jennic Platform used for demonstration [12].

## V. Results and Lessons Learned

Within the SmartCoDe Project, we have designed an SoC that includes Wireless Communication based on ZigBee standard, power metering, power supply, and smart-card based security [20].

### *A. What in HIL, what in Simulation?*

Also as part of the SmartCoDe project, the model-based approach has been used for the following tasks:

- System level design issues such as partitioning, resource budgeting, etc.
- Development of methods for energy management, because in the model based approach it is much easier to set up different scenarios with a huge number of appliances.
- Evaluation of the power efficiency of protocols and network topology, in particular those based on IEEE 802.15.4.
- Development of embedded system hardware.

The hardware prototype has been used within a complex demonstrator site. There, it allowed us to

- Evaluate dependability of the wireless connections in real buildings.
- Evaluate impact and savings in a realistic scenario.
- Analyze interaction with human users.

Dependability of the wireless connections and interaction with human users had partially unexpected results: While wireless connections are mostly dependable, they are not if users for example put some iron parts close to the antennas. Furthermore, practical application showed a lack of robustness against shore black- or brown outs. In future work, the framework will be enhanced with accurate means for modeling obstacles, and more sophisticated means for modeling the user behavior.

### *B. Performance Benchmarks*

Apart from the framework applicability, a key issue is how it performs. In order to evaluate performance, virtual scenarios based on a fridge model validated in SmartCoDe Project [12], have been set up.

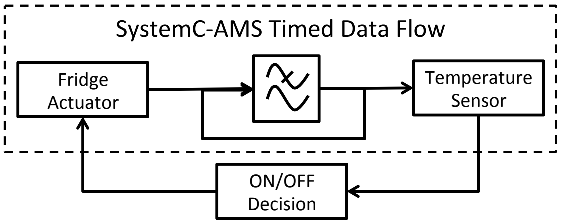


Fig. 5. Diagram of the SystemC-AMS fridge model

Comparisons with other simulators are intentionally avoided. There are very few simulators with the same scope, and even those, make emphasis in different aspects and use models with different detail levels. As a result, it is difficult to compare simulators without falling into unfair comparisons. However, we still think that this kind of results can be valuable to the experienced user and provide some notions about how the framework performs.

We focus on the simulation performance of the physical environment. In our scenario, this physical model is a temperature variation model within a fridge. Temperature variation has been modeled as an RC low-pass filter, which is an already validated approach [17] [8]. This low-pass filter, depicted in Figure 5 is a SystemC-AMS TDF module.

In Figure 6 the simulation performance of networks with different number of fridges is shown. The network consists of the specified number of fridges and an energy management unit which acts just as a sink node. The communication model includes the IEEE 802.15.4 Standard physical and MAC layer protocols, based on the Atmel AT86RF230 ZigBee transceiver electrical characteristics. No routing protocol is included in the model. The figure starts with an exponential growth but at some point it converges to almost linear progression. Of course this depends also on the network configuration and topology, but in this case communication has been isolated. Further information about communication and the wireless physical layer simulation performance can be found in [14].

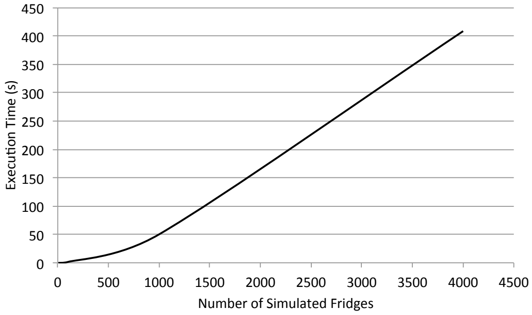


Fig. 6. Simulation performance for temperature simulation (different node number)

Table I shows that execution time grows linearly with simulated time. Of course, for long simulation times this is trivial, but we can also perceive this behavor in very short execution time simulations, which means that the overhead due to initialization (all SystemC modules instantiation and all TLM generic payload instantiation and population) has no significant impact in execution time.

TABLE I
SIMULATION PERFORMANCE FOR DIFFERENT TEMPERATURE SIMULATION TIME PERIODS

| Simulated Time | Execution Time |
|---|---|
| 1h | 1.024s |
| 3h | 2.992s |
| 12h | 11.365s |
| 24h | 22.949 s |
| 1 week | 158.518 s |

## VI. Conclusion

In this paper we have presented a simulation framework for a Model-Based Design methodology of embedded systems for energy management applications. Optimization of this kind of systems requires evaluating cross-layer and cross domain aspects and interactions.

The introduced SystemC based framework permits modeling all the related aspects by leveraging SystemC extensions like Transaction-Level Modeling and Analog Mixed-Signal extensions. The result is an efficient simulation framework, capable of simulating networks of over a thousand nodes, each one with its own different physical environment model.

## References

[1] M. Short and M. Pont, "Hardware in the loop simulation of embedded automotive control system," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, sept. 2005, pp. 426 – 431.

[2] J. Jensen, D. Chang, and E. Lee, "A model-based design methodology for cyber-physical systems," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, july 2011, pp. 1666 –1671.

[3] A. Sangiovanni-Vincentelli and G. Martin, "Platform-based design and software design methodology for embedded systems," *Design & Test of Computers, IEEE*, vol. 18, no. 6, pp. 23–33, 2001.

[4] M. Erol-Kantarci and H. Mouftah, "Wireless sensor networks for smart grid applications," in *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, april 2011, pp. 1 –6.

[5] A. Llaria, O. Curea, J. Jimenez, J. Martin, and A. Zuloaga, "Wireless communication system for microgrids management in islanding," in *Power Electronics and Applications (EPE 2011), Proceedings of the 2011-14th European Conference on*, 30 2011-sept. 1 2011, pp. 1 –10.

[6] "Ptolemy II," UC Berkeley EECS Dept. [Online]. Available: http://ptolemy.berkeley.edu/ptolemyII/

[7] E. Widl, P. Palensky, and A. Elsheikh, "Evaluation of two approaches for simulating cyber-physical energy systems," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, oct. 2012, pp. 3582 –3587.

[8] A. Elsheikh, E. Widl, and P. Palensky, "Simulating complex energy systems with modelica: A primary evaluation," in *Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on*, june 2012, pp. 1 –6.

[9] A. T. Al-Hammouri, "A comprehensive co-simulation platform for cyber-physical systems," *Computer Communications*, vol. 36, no. 1, pp. 8–19, 2012.

[10] J. Lin, S. Sedigh, and A. Miller, "Modeling cyber-physical systems with semantic agents," in *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*, july 2010, pp. 13 –18.

[11] *IEEE Standard SystemC Language Reference Manual*, IEEE Std., Rev. 2011, 2006.

[12] J. Moreno, M. Damm, J. Haase, C. Grimm, and E. Holleis, "Unified and comprehensive electronic system level, network and physics simulation for wirelessly networked cyber physical systems," in *Specification and Design Languages (FDL), 2012 Forum on*. IEEE, 2012, pp. 68–74.

[13] J. Haase, J. Moreno, and D. Dietrich, "Power-aware system design of wireless sensor networks: Power estimation and power profiling strategies," *Industrial Informatics, IEEE Transactions on*, vol. 99, no. 99, pp. 1–1, 2011.

[14] M. Damm, J. Moreno, J. Haase, and C. Grimm, "Using transaction level modeling techniques for wireless sensor network simulation," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, Mar. 2010, pp. 1047 –1052.

[15] M. Kircher and P. Jain, "Pooling," in *Proceedings of the 2002 European Conference on Pattern Languages of Programs*, 2002.

[16] *SystemC AMS Extensions 2.0 Language Reference Manual*, Accellera Systems Initiative Draft, March 2012.

[17] F. Kupzog and C. Roesener, "A closer look on load management," in *Industrial Informatics, 2007 5th IEEE International Conference on*, vol. 2, june 2007, pp. 1151 –1156.

[18] J. Moreno, J. Wenninger, J. Haase, and C. Grimm, "Energy profiling technique for network-level energy optimization," in *AFRICON, 2011*. IEEE, 2011, pp. 1–6.

[19] FP7, "Smartcode project," https://www.fp7-smartcode.eu/.

[20] S. Mahlknecht, M. Damm, and C. Grimm, "A smartcard based approach for a secure energy management node architecture," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*. IEEE, 2010, pp. 769–773.

# Security in cyber-physical energy systems

Usman A. Khan and Aleksandar M. Staković
Department of Electrical and Computer Engineering, Tufts University
161 College Ave., Medford, MA 02155, USA
Email: {khan,astankov}@ece.tufts.edu

***Abstract*— In this paper, we consider cyber-security in the context of distributed estimation in Cyber-Physical Energy Systems (CPES). Distributed estimation is where the state of the energy system is to be estimated via a collection of geographically dispersed sensors. In order to implement the estimator, the sensors, in addition to sensing, implement a simple data fusion protocol that relies on inter-sensor communication. We study estimation of energy systems when there is an adversarial attack on the sensed and communicated information. We propose a novel methodology to address the detection of such attacks, and further incorporate appropriate remedial actions in the estimator. Our methodology is based on the notions of local consistency and nodal consistency and is further reinforced by exploiting the underlying physical-layer in the energy-system description.**

## I. Introduction

This paper outlines a paradigm that aims to develop a new class of secure networked estimation and control procedures for energy systems (power systems, electric drives and power electronics) with the aim of providing the analytical foundations for Cyber-Physical Energy Systems (CPES). The procedure will combine descriptions of the energy layer and of the information layer into one coordinated structure. The aim is to explicitly maintain the identity of each layer so that practical problems in control, estimation, and fault accommodation can be addressed in a realistic and productive way. At present, the two layers are not fully interconnected and the information layer is not fully networked.

We focus on cyber-physical energy systems (CPES), where networking of the cyber (information) layer is needed to enable new functionalities [1], [2]. Key aspects of the cyber layer can be tabulated as follows [3]: (i) Controllers are commonly implemented on computers, offering additional (and presently underutilized) functionalities; (ii) Networking of the information layer is pervasive for computational and communication efficiency reasons; (iii) Availability of commodity information technology (IT) solutions, offering low price and open design, but cyber-security is a concern; and (iv) Large and highly skilled IT workforce is available globally.

However, the integration of a cyber layer with the energy layer is not straightforward. *Firstly*, each layer is operated in a way that is only vaguely aware of the other (but tightly coupled) layer. For example, sensors and actuators are not sufficiently coupled in the energy layer– control is too local or even detrimental to the system as a whole, and the standard software designs include timing information only with great difficulty. *Secondly*, emerging need to communicate with a multitude of spatially distributed sources and loads overwhelms existing communication and control design procedures. *Thirdly*, cyber-crime is a downside [4], and it should be dealt with at all steps of system design and operation. We argue later that energy portions of CPES descriptions can be used for model-based detection of cyber intrusions.

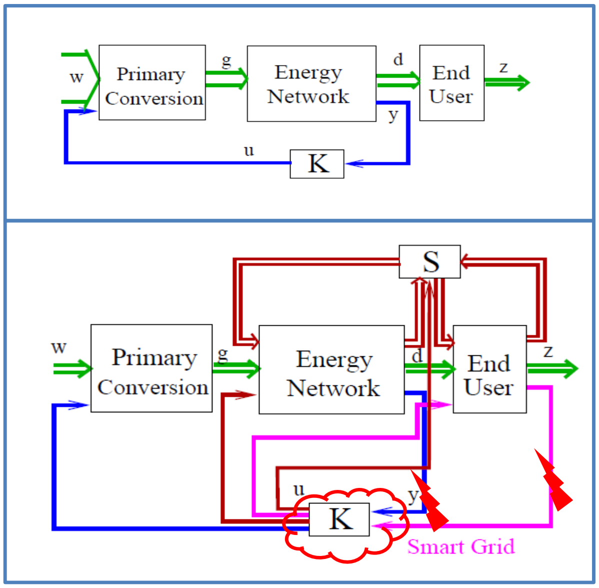


Fig. 1. A schematic of existing (top) and future (bottom) electric energy systems.

**Cyber-security aspects:** In terms of communication modalities, the future cyber-layer will include a variety of options, from Power Line Carriers (PLC) to Ethernet, ZigBee, and Bluetooth. While the promise of the improved information flow is clear, it also brings a host of potential problems, largely centering around cyber-network reliability and security. While the reliability problems are expected to be dealt with in the natural evolution of underlying technologies, the cyber security problem needs to be addressed in the very foundations of the CPES concept. The list of potential adversaries varies from poorly trained personnel and hackers, to competitors and disgruntled employees, and, to criminals and terrorists. Broadly speaking, cyber-attacks can be component-based (like Stuxnet that focused on a single product line of a single manufacturer) or protocol-based (e.g., tailored to different SCADA systems). Security features that are particular to CPES include numerous legacy components, geographically dispersed field locations and "security through obscurity" as the organizational features of utilities are largely opaque to the outsiders.

We envision cyber-physical energy systems to follow Fig. 1–bottom, in contrast to the existing infrastructure of Fig. 1–top. In the bottom architecture, loads are brought "inside" the system by suitable information flows, and the distributed multi-level control combines local and contextual (global) control. Overall efficiency improvements would reduce the energy input $\mathbf{w}$ while encouraging transition to sustainable sources. Please note that "**S**" can denote both electric storage and a scheduled exchange with other energy systems (gas, heat) through "energy hubs". The present Smart Grid initiative largely focuses on the highlighted feedback path. Traditional systems theory analyses, although relevant, is highly impractical mainly due to the sheer size of the CPES architecture. These challenges are foreseen to be overcome by distributed solutions but the focus has been predominantly towards (i) algebraic principles, e.g., rank tests for observability/controllability, and (ii) asymptotic performance guarantees, e.g., consensus-based approaches. Furthermore, the cyber layer (over which the distributed solutions are formulated) adds to the design challenges as the traditional protocols must be adjusted for the underlying communication network and its imperfections.

In the remainder of the paper, we describe the proposed cyber threat models and secure inference protocols in Section II. Section III demonstrates the proof of concept on a simple yet illustrative scenario and finally, Section IV concludes the paper.

## II. Networked Sensing, Dynamic Estimation, and Security

Distributed estimation would enable key advances in Smart Grid such as retail markets and calculation of real-time distribution Locational Marginal Prices (LMPs) [5], as well as provide inputs for multi-objective optimization. Distributed estimation in Cyber-Physical Energy Systems (CPES) depends heavily on the reliability of the data. This is because CPES have been envisioned to have an intricate cyber layer over which the data is exchanged between essential system components for the purposes of control, estimation, and other systemic analyses. The wide-scale operation and socio-economic impacts of CPES demands such a cyber layer to be secure and robust to communication and sensing threats. As a starting point for the formulation of secure protocols, we assume the cyber threats have the following properties, see Fig. 2:

(i) **Compromised communication**: An adversary gets control of one or more of the outgoing links at node $i$ and sends meaningless information to one or more neighbors of node $i$.

(ii) **Compromised sensors**: An adversary gets control of the sensors at node $i$ and sends meaningless sensing information to node $i$.

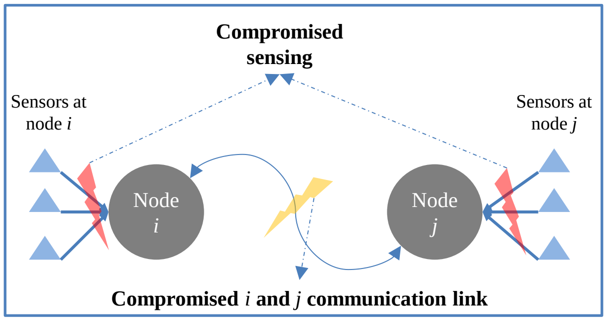


Fig. 2. Security concerns at distributed nodes.

This classification is further appended with the following two assumptions: (a-i) The number of compromised nodes (in either sensing or communication sense) in any neighborhood is much less than the number of non-compromised nodes; and (a-ii) The adversary is not an oracle in the context of the underlying system. In other words, the adversary does not have the complete physical and/or cyber knowledge of the underlying dynamics. Note that (a-i) is widely used on the adversaries, e.g., consider the $F$-local and $F$-global standard Byzantine adversary models [6]–[9]. Whereas, the assumption (a-ii) is also natural as an adversary may obtain the seasonal variations, historical data, and other high-level system descriptors, but does not know the system transients and current operating points etc.

The above classification entails a large set of practical threats that can be targeted towards a CPES. Our philosophy towards designing secure protocols in CPES is to exploit the underlying physical models that are "in some sense" common across different CPES modules and further provide a means to verify the information exchanged over the cyber layer. We can broadly based the proposed solution on the following ideas.

(a) **Nodal consistency**: Any node $i$ with an information (data)set $\mathcal{I}_i$ may declare its own dataset to be *trusted* if the evolution of this dataset is *statistically consistent* over time.

(b) **Local consistency**: Consider two directly connected nodes $i$ and $j$ with information (data)sets, $\mathcal{I}_i$ and $\mathcal{I}_j$. Assuming that the two datasets have information about a few common elements, node $i$ may declare the entire $\mathcal{I}_j$ to be trusted if $\mathcal{I}_j$ is *statistically consistent* with $\mathcal{I}_i$ over the common elements.

(c) **Physical-layer feedback**: Any node $i$ may declare itself, $i$, (or a neighboring node $j$) to be *trusted* if the dataset, $\mathcal{I}_i$, (or $\mathcal{I}_j$ or at node $j$) is *statistically consistent* with the physical-layer feedback from the neighboring nodes.

It can be readily seen that the above threat modeling (i-ii) and trust notions (a-c) are highly relevant to CPES, where the nodes are different system modules that share, possibly very few, common elements between them and are inter-connected via a, possibly low-bandwidth, physical layer feedback. Similarly, the trust notion (c) above exploits the fact that the information sets at any agent are highly coupled to the physical-layer interconnections. It is worth mentioning that the trust notions (a-c) are not deterministic, but statistical. Finally, the notion of *statistically consistency* may refer to distribution shifts over time in (a), hypothesis testing and false-alarm rates towards establishing a level of trust in (b), and the statistical coupling between the physical-layer feedback and cyber data in (c).

Please note that the above cyber attack classification is different from predominant models in the literature. This is because most of the cyber attack modeling is restricted to computer networks where an underlying energy (physical) layer is either not present or ignored. For example, recent work on communication and consensus in the presence of adversaries, [6], [7], [10]–[12], does not consider the underlying physical layer; primarily because the system is only driven by information and there is no physical phenomenon.

### A. Secure networked inference

Secure estimation of *dynamical* systems is largely unexplored in the literature. In the context of power system state-estimation, this problem is typically cast as bad data detection, see [13]–[16] and references therein, where the estimation is of a static parameter. Recent extensions to dynamic estimation have been proposed in the purview of information-theoretic security constructs where analytical results are restricted to point-to-point communication, for example, see [17]–[20]. Of particular relevance is [21], which describes data-injection attacks and detection in smart grid but is restricted to static estimation of dc power flow model and only considers a subset of what we categorized as *compromised sensing*, i.e., the sensing model has an unknown constant shift with the collection of observations being observable in one time-step. On the contrary, we place no such assumptions in the following proposed strategies while the notions of *local consistency* and *nodal consistency* can also be verified to be novel in the context of *dynamical system* estimation.

We now describe our approach to address the cyber security issue in networked estimation, however, the solution can be extended to other related problems. Networked estimation is to estimate the state variable, $\mathbf{x}_k$, in the electric power grid with distributed observations, $\mathbf{y}_k^i$, where the superscript denotes the geographically distributed nodes, $i = 1, \ldots, N$. The node here implies a local workstation that has sensors, measuring some states in the state vector, and is connected to nearby nodes via wireless–or wired–communication, see Fig. 2. To cast the proposed formulation in a proper mathematical context, we assume the following discrete-time LTI dynamics, perhaps after linearization and discretization. The system state, $\mathbf{x}_k \in \mathbb{R}^n$, at time $k \geq 0$ is given by

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{v}_k, \tag{1}$$

$$\mathbf{y}_k^i = C_i\mathbf{x}_k + \mathbf{r}_k^i, \qquad i = 1, \ldots, N, \tag{2}$$

where $A$ is the system matrix–possibly linearized and discretized, and $\mathbf{v}_k \sim N(0, Q)$ is the normally distributed system noise; whereas at each node $i$: $\mathbf{y}_k^i \in \mathbb{R}^{m_i}$ is the local observation vector, $C_i \in \mathbb{R}^{m_i \times n}$ is the local observation matrix and $\mathbf{r}_k^i \sim N(0, R_i)$ is the local normally distributed observation noise. This simple discrete-time linear modeling is assumed to simplify the following discussion and cast the ideas in a precise mathematical framework, however, the models can be extended to non-LTI models.

*1)* **Compromised communication:** We now address case (i) in the aforementioned cyber attack categories, i.e., when the sensing is not compromised but the communication may be inflicted with a cyber attack. The following precisely establishes the notion of *local consistency* introduced before. From Eq. (2), note that any two local observation vectors, $\mathbf{y}_k^i \in \mathbb{R}^{m_i}$ and $\mathbf{y}_k^j \in \mathbb{R}^{m_j}$ are not directly comparable as: (i) the dimensions may be different; and (ii) the corresponding elements of $\mathbf{y}_k^i$ and $\mathbf{y}_k^i$ may represent different state-variables. To circumvent this issue, we construct the auxiliary observations, $\widetilde{\mathbf{y}}_k^i \in \mathbb{R}^n$ as

$$\widetilde{\mathbf{y}}_k^i = C_i^T \mathbf{y}_k^i, \tag{3}$$

which does not only make each local observation to have the same dimension but corresponding elements of $\mathbf{y}_k^i$ and $\mathbf{y}_k^j$ now represent the same state-variable across all auxiliary observations. The secure approach to networked estimation that we propose exploits the *commonness* among the auxiliary observations.

**Remark 1:** It can be easily verified that node $i$ can perform a meaningful estimation of the state-variables corresponding to the non-zeros in the auxiliary observation from its own measurements without relying on its neighbors. *However*, in order to estimate the state variables corresponding to the zeros in the auxiliary observations, node $i$ has to rely on its neighbors; this is where compromised communication can be detrimental.

With the commonness among auxiliary observations and Remark 1, we describe the following protocol at each node. Let $\mathcal{N}_i$ denote the neighborhood of nodes $i$, i.e., $\mathcal{N}_i = \{i\} \cup \{j \mid j \rightarrow i\}$, where $j \rightarrow i$ means that node $j$ can send information to node $i$. For each $j \in \mathcal{N}_i$, node $i$ tabulates the commonness in the auxiliary observations, defined as $X_{ij} = \{x_\ell \mid \widetilde{\mathbf{y}}^i(\ell) \neq 0 \textit{ and } \widetilde{\mathbf{y}}^j(\ell) \neq 0\}$, i.e., the collection of state variables for which both node $i$ and node $j$ has measurements. Subsequently, node $i$ assigns a *trust index*, $t_{ij}(k)$, to every neighboring node as follows:

$$t_{ij}(k) \propto \sum_{x_\ell \in X_{ij}} \sum_{m=1}^{k} (\widehat{x}^i_{\ell,m} - \widehat{x}^j_{\ell,m})^2, \tag{4}$$

where $\widehat{x}^i_{\ell,m}$ is the estimate of the $\ell$th state-variable at node $i$ and time $m$. Please note that the trust index is only defined on the common estimable states among node $i$ and $j$. With the help of the trust index, $t_{ij}(k)$, node $i$ declares the *trusted neighbors* at time $k$ as

$$\overline{\mathcal{N}}_i(k) = \{j \in \mathcal{N}_i \mid t_{ij}(k) < \varepsilon_{ij}\}, \tag{5}$$

and the *compromised neighbors* as $\underline{\mathcal{N}}_i(k) = \mathcal{N}_i \backslash \overline{\mathcal{N}}_i(k)$. Finally, node $i$ updates its state-estimate by assigning more weight to the trusted neighbors and less (or zero) to the compromised.

**Computation of $\varepsilon_{ij}$:** A significant question in the above formulation is how to compute $\varepsilon_{ij}$ as this *threshold* is a significant contributor to the set of trusted neighbors. With some care, the design of $\varepsilon$ can be cast in a precise statistical context. For this purpose, let us analyze the statistics of the trust index, $t_{ij}(k)$, in Eq. (4). Assuming that each state-estimate, $\widehat{x}^i_{\ell,k}$, for all $i$'s, is distributed as $N(x_{\ell,k}, \sigma^2)$, one can show that $t_{ij}(k)$ is distributed as $N\left(0, \sigma^2|X_{ij}|\right)$. Finally, the computation of $\varepsilon$ can be cast in terms of the false alarm rate of the following hypothesis testing problem:

$$H_0 : t_{ij}(k) \quad \sim \quad N\left(0, \sigma^2|X_{ij}|\right), \tag{6}$$

$$H_1 : t_{ij}(k) \quad \sim \quad N\left(\neq 0, \sigma^2|X_{ij}|\right). \tag{7}$$

The case when $\widehat{x}^i_{\ell,k} \sim N(x_{\ell,k}, \sigma_i^2)$ can be easily adjusted in the above scenario. Finally, it is noteworthy that $\sigma_i^2$ can be estimated using the signal-to-noise ratio at node $i$, i.e., from Eq. (1).

*2)* **Compromised sensing:** It can be argued that in the aforementioned description of compromised communication, the basis of the proposed secure networked estimation is to rely on self-sensing, and use this to exploit the commonness among the neighboring nodes that subsequently results into a trust. However, the procedure fails when there is an attack on the sensors, for example, node $i$ may declare each neighbor to be compromised without realizing that its own sensing was attacked. We use the *nodal consistency* notion defined before to address this scenario. Relying on the state-space description, i.e., Eq. (1), already available at node $i$, we propose each node $i$ to track the statistics of the following quantity: $\mathbf{z}^i_k = K_i(y^i_k - C_i A \widehat{\mathbf{x}}^i_{k-1})$, which are the innovations in the Kalman filter sense. Please note that a similar innovation can be formulated for non-LTI systems.

When the sensors at node $i$ are *not* compromised, it can be verified that $z^i_k$ is given by

$$\mathbf{z}^i_k = K_i C_i (\mathbf{x}_{k-1} - \widehat{\mathbf{x}}^i_{k-1}) + K_i C_i \mathbf{v}_{k-1} + K_i \mathbf{r}^i_k, \tag{8}$$

which can easily be shown to have zero-mean, i.e., $\mathbb{E}(\mathbf{z}^i_k) = \mathbf{0}$, since $\widehat{\mathbf{x}}^i_{k-1}$ is an unbiased estimate, see [22] for details. Now consider that the sensors at node $i$ are compromised at some time $k \geq k^c_i$. The shift in the statistics of the sequence $\mathbf{z}^i_k$ contains meaningful information in order for node $i$ to realize that its sensors are no longer contributing useful information. An important research question is to investigate how long does it take after the attack initiation at time $k^c_i$ for node $i$ to declare that its sensors are compromised. This problem can be cast as statistical modeling of a time-series and detection of a shift in its statistics. It remains to be investigated that given the underlying system models and noise statistics, what is the minimal time-period that is required to declare an attack (following the attack initiation at $k^c_i$) with a desired confidence.

*3)* **Integrating the physical-layer feedback:** The nodal and local consistency cast earlier in the context of the threat models (i-ii) can be further augmented with, possibly partial, physical-layer models available at any node. For this purpose, let us define $\overline{\mathcal{I}}_i$ to be the collection of all measurements available in the set $\mathcal{N}_i$, i.e., at node $i$ and its neighbors $i \leftarrow j$, and $\widetilde{\mathcal{I}}_i$ to be some subset of $\overline{\mathcal{I}}_i$. We propose to compute the following quantity: $\widetilde{p}_i \triangleq \mathbb{P}\left(\widetilde{\mathcal{I}}_i \notin f(\widetilde{\mathbf{x}}_i)\right)$, where $\widetilde{\mathbf{x}}_i$ is the (partial) system-wide information available at node $i$ due to physical-layer interconnections, and $f(\cdot)$ is the projection of $\widetilde{\mathbf{x}}_i$ to the information (data)set, $\widetilde{\mathcal{I}}_i$. That $f(\cdot)$ can be readily computed at node $i$ is obvious as each node $i$ has (partial or complete) knowledge of the underlying physical model, i.e., Eq. (1). The index of confidence, $\widetilde{p}_i$, over the information set of interest, $\widetilde{\mathcal{I}}_i \subseteq \overline{\mathcal{I}}_i$, can now be incorporated in the aforementioned strategies for compromised communication and sensing.

It is noteworthy that the security paradigm we propose above for *dynamical system estimation* has not been explored in the given setup of cyber attack modeling and the proposed consistency framework. The setup described in this section is further not restricted to linear time-invariant models but can generalized in the larger framework of non-LTI dynamics as the consistency notions and proposed trust indices are not restricted to linear models. As we described before, much of the related work in the literature assumes simplified cyber attacks and relatively simple attack detection strategies while being restricted *only to static estimation*. Our formulation does not only provide a complete framework to describe and detect cyber attacks but also derives appropriate remedial actions in a proper mathematical context, see [23] for more details.

## III. Simulations

Consider a simple $n = 5$ state,

$$\mathbf{x}_k = \left[\begin{array}{cccc} x_{k,1} & x_{k,2} & \dots & x_{k,5} \end{array}\right]^T,$$

DT-LTI system with 5 nodes such that the $i$th node observes the $i$th state, $x_{k,i}$, and the $i+1$th state, $x_{k,i+1}$, except node 5 that observes $x_{k,5}$ and $x_{k,1}$. The nodes are connected as in Fig. 3. For example, node 3's observation model, $\mathbf{y}_k^{(3)} \in \mathbb{R}^2$, is

$$\mathbf{y}_k^{(3)} = \left[\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right] \mathbf{x}_k + \mathbf{r}_k^{(3)}, \tag{9}$$

where $\mathbf{r}_k^i$ is chosen to be $\mathbb{N}(0, I)$, $\forall i$ for simplicity. Similarly, $\mathbf{y}_k^{(2)}$ is also a vector in $\mathbb{R}^2$ but observes $x_{k,2}$ and $x_{k,3}$ and thus $\mathbf{y}_k^{(2)}$ and $\mathbf{y}_k^{(3)}$ cannot be compared directly. To avoid this issue, we construct auxiliary observations so that each $\widetilde{\mathbf{y}}_k^i \in \mathbb{R}^n$, e.g., $\widetilde{\mathbf{y}}_k^{(2)} = C_2^T C_2 \mathbf{x}_k + \mathbf{r}_k^{(2)}, \widetilde{\mathbf{y}}_k^{(3)} = C_3^T C_3 \mathbf{x}_k + \mathbf{r}_k^{(3)}$, where $C_2^T C_2$ is diagonal with 1's at $(2,2)$ and $(3,3)$ locations and zeros everywhere else; and $C_3^T C_3$ is diagonal with 1's at $(3,3)$ and $(4,4)$ locations and zeros everywhere else. This establishes the commonness among each node as the common non-zeros on the auxiliary observation matrices, $C_i^T C_i$. From Fig. 3, it is clear that node $i$ and $i+1$ share an observation on the state $x_k^{\max(i,i+1)}$ (except for node 1 and 5, which share the first state, $x_{k,1}$).

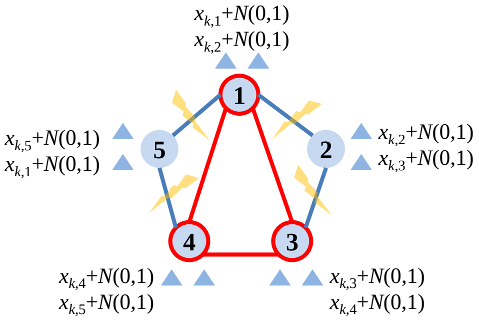


Fig. 3. Simulation setup.

We assume that the communication links from node 2 and node 5 are compromised such that instead of sending meaningful information to their neighbors, the adversary sends $\mathbb{N}(0, \sigma_a^2)$. In order for node $1, 3, 5$ to continue uninterrupted operation, node 3, for example, proceeds as the following, with respect to node 2: Let the local estimates of the common state, $x_{k,3}$, between node 3 and node 2 be denoted by $\widehat{x}_{k,3}^{(3)}$ and $\widehat{x}_{k,3}^{(2)}$, respectively, where the estimator we employ is the single time-scale estimator from [22]. Since the estimator in [22] is linear and unbiased, we have $\widehat{x}_{k,3}^{(\cdot)} \sim \mathbb{N}(x_{k,3}, \times)$, where $\times$ represent that the variance is ignored. This further leads to:

When there is no attack,

$$t_{32}(k) = \sum_{m=1}^{k} \left(\widehat{x}_{m,3}^{(3)} - \widehat{x}_{m,3}^{(2)}\right)^2 \sim \mathbb{N}(0, \times).$$

When there is attack,

$$t_{32}(k) = \sum_{m=1}^{k} \left(\widehat{x}_{m,3}^{(3)} - \mathbb{N}(0, \sigma_a^2)\right)^2 \sim \mathbb{N}(\neq 0, \times).$$

Over a sequence of time-steps $k$, node 3 thus keeps track of the quantity $t_{32}(k)$ and follows the local consistency procedure described in Section II-A.1. The precise hypothesis testing formulation requires a detailed computation of the corresponding co-variances (denoted as $\times$) that is beyond of the scope of this paper. However, regardless of the knowledge of $\sigma_a^2$, an effective zero-mean comparison can be devised on the sequence of $t_{ij}(k)$'s, see Fig. 5.

We simulate an $n = 5$-dimensional DT-LTI system with $\sigma_a^2 = 1$ and plot the sum of squared errors at each agent using the non-secure estimator, Eq. [22] in Fig. 4 (Top and middle) for stable and unstable dynamics. Subsequently, Fig. 4 (Bottom) shows the secure estimation established in Section II-A.1. Finally, we show a typical evolution of $t_{32}(k)$ from Eq. (4) in Fig. 5 under attack and no attack cases. It can be verified that for stable dynamics (Fig. 4 (Top)), attack or no-attack results in bounded estimation error (the performance under no attack is obviously better); this is because stable dynamics will eventually die out and the state itself remains bounded and hence a trivial estimate (e.g., 0) results in bounded estimation error.

The more interesting case is when the dynamics are unstable as the nodes under attack cannot perform a meaningful estimation while further degrading the performance of the non-attacked nodes (Fig. 4 (Middle)). Finally, please note that Fig. 4 shows an average over 5000 Monte Carlo trials.

## IV. Conclusions

In this paper, we provide a novel security paradigm that is cast in a concrete setup of cyber attack models and the statistical consistency framework. This setup is particulary useful for Cyber-Physical Energy Systems (CPES) as the distribution of the data and future transition to a cyber infrastructure will heavily rely on secure protocols.

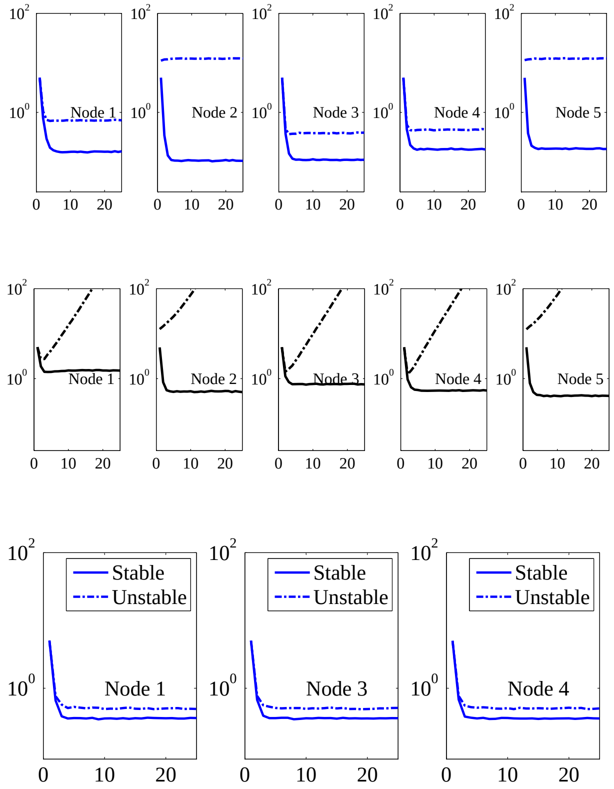


Fig. 4. Stable/unstable dynamics, largest eigenvalue of $A$ is 0.75 (Top) and 1.15 (Middle). MSE (vertical) plotted against time-step, $k$ (horizontal). (Top and Middle) Solid curve is no attack and dashed curve is cyber attack. (Bottom) Secure estimation: Solid curve is stable dynamics and dashed curve is unstable dynamics.

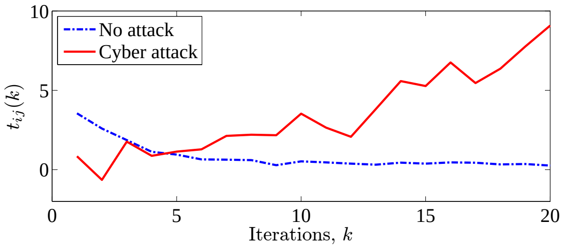


Fig. 5. Evolution of the trust index, $t_{32}(k)$.

## References

[1] Lui Sha, Sathish Gopalakrishnan, Xue Liu, and Qixin Wang, "Cyber–physical systems: A new frontier," in *Machine Learning in Cyber Trust*, pp. 3–13. Springer US, 2009.

[2] J. A. Stanković, I. Lee, A. Mok, and R. Rajkumar, "Opportunities and obligations for physical computing systems," *Computer*, vol. 38, no. 11, pp. 23–31, Nov. 2005.

[3] Alvaro A. Cárdenas, Saurabh Amin, and Shankar Sastry, "Research challenges for the security of control systems," in *Proceedings of the 3rd conference on Hot topics in security*, Berkeley, CA, USA, 2008, pp. 1–6.

[4] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *Critical Infrastructure Protection II*, M. Papa and S. Shenoi, Eds., vol. 290 of *The International Federation for Information Processing*, pp. 71–85. Springer, 2009.

[5] G. T. Heydt, "The next generation of power distribution systems," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 225–235, Dec. 2010.

[6] C.-Y. Koo, "Broadcast in radio networks tolerating byzantine adversarial behavior," in *Proceedings of the twenty–third annual ACM symposium on Principles of distributed computing*, New York, NY, USA, 2004, pp. 275–282.

[7] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate byzantine consensus in arbitrary directed graphs," in *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, New York, NY, USA, 2012, PODC '12, pp. 365–374.

[8] H. Zhang and S. Sundaram, "A simple median–based resilient consensus algorithm," in *50th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, Oct. 2012.

[9] H. J. LeBlanc and X. Koutsoukos, "Resilient asymptotic consensus in asynchronous robust networks," in *50th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, Oct. 2012.

[10] R. M. Kieckhafer and M. H. Azadmanesh, "Low cost approximate agreement in partially connected networks," *Journal of Computing and Information*, 1993.

[11] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, Apr. 1985.

[12] S. Sundaram, S. Revzen, and G. J. Pappas, "A control-theoretic approach to disseminating values and overcoming malicious links in wireless networks," *Automatica*, vol. 48, no. 11, pp. 2894–2901, 2012.

[13] F. F. Wu and W.-H. E.Liu, "Detection of topology errors by state estimation," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 176–183, Feb. 1989.

[14] A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation*, NewYork: Marcel Dekker, 2004.

[15] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM conference on Computer and communications security*, New York, NY, USA, 2009, pp. 21–32.

[16] G. Dan and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *2010 First IEEE International Conference on Smart Grid Communications*, Oct. 2010, pp. 214–219.

[17] L. Lai, K. Liu, and H. El Gamal, "The three-node wireless network: Achievable rates and cooperation strategies," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 805–828, Mar. 2006.

[18] L. Lai and H. El Gamal, "The relay eavesdropper channel: Cooperation for secrecy," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 4005–4019, Sep. 2008.

[19] H. Li, L. Lai, and W. Zhang, "Communication requirement for reliable and secure state estimation and control in smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 476–486, Sep. 2011.

[20] V. Aggarwal, L. Sankar, R. A. Calderbank, and H. V. Poor, "Secrecy capacity of a class of orthogonal relay eavesdropper channels," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 1–14, Mar. 2009.

[21] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor, and A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 106–115, Sep. 2012.

[22] U. A. Khan, S. Kar, A. Jadbabaie, and J. M. F. Moura, "On connectivity, observability, and stability in distributed estimation," in *49th IEEE Conference on Decision and Control*, Dec. 2010, pp. 6639–6644.

[23] U. A. Khan and A. Stanković, "Secure distributed estimation in cyber-physical systems," in *38th IEEE International Conference on Acoustics, Speech, and Signal Processin*, Vancouver, BC, May 2013, accepted for publication.

# Hierarchical application of model-predictive control for efficient integration of active buildings into low voltage grids

Alexander Schirrer[1], Oliver König[1], Sara Ghaemi[2], Friederich Kupzog[2], Martin Kozek[1]

[1] Institute of Mechanics and Mechatronics
Vienna University of Technology
Vienna, Austria
{alexander.schirrer|oliver.koenig}@tuwien.ac.at

[2] Electric Energy Systems, Energy Department
AIT Austrian Institute of Technology
Vienna, Austria
{sara.ghaemi|friederich.kupzog}@ait.ac.at

*Abstract*— **Active buildings are important contributors in urban power grids and they are expected to be more integrated with renewable energy resources. At the same time, the amount of additional renewable sources that can be integrated in existing urban distribution grids is restricted by technical limitations. This work introduces a novel hierarchical predictive model controller architecture that takes both the power grid and the building domains into account. In order to maximize the hosting capacity of existing grids, considering the role of the buildings' energy performance in the active distribution grid is essential. Minimal grid congestion and maximum self-coverage of building energy demand are two goals that potentially can be traded off efficiently by this approach.**

***Keywords— Model-predictive control, smart grids, building energy optimisation, thermal building models, building-to-grid system interaction***

## I. Introduction

Technical boundaries of the electricity system such as grid congestions and capacity constraints or voltage limits represent considerable barriers which limit the integration of Renewable Energy Sources (RES) and Distributed Generation (DG) [1]. In urban scenarios, different restrictions play a role compared to rural areas [2]. Without considering adequate measures for active integration of RES, such as energy management in the building domain, peak shaving or reactive power management in the generation domain or optimal topology switching in the grid domain, only a small fraction of the theoretical maximum generating potential is possible.

In order to reach a global optimum in terms of self-consumption and efficient use of renewable energies, not only the performance of the individual buildings but also the operation of the distribution grid should be optimized simultaneously. Therefore, this work proposes a coordinated energy management system for buildings and distribution grids based on model-predictive control (MPC).

The concept of using energy management measures to improve efficiency in buildings in the context of smart grids is not new. When managing resources inside the building for the sake of improvements on the power grid side, this is classical demand side management [3]. Often, these systems work on variable energy prices (see e.g. [4]) and many of them use multi-agent approaches [5] [6]. From a power grid perspective, the building can be technically enabled to provide ancillary services and participate in the respective markets [7]. Significant loads that can be managed in buildings are electrical heating, ventilation and cooling systems, or heat pumps [8].

This paper introduces the project SmartCityGrid: CoOpt (COordinated OPTimization of renewable energies in buildings and distribution grid). The goal of this project is to optimize the energy consumption and demand coverage in buildings considering on-site generation. The project analyzes how building-integrated renewables can be implemented in large quantities in urban low-voltage grids using MPC. By an optimized use and integration of thermal and potentially electrical storages and passive storage masses (buildings) into the urban energy management it will be possible to integrate the locally produced renewable energy optimally – in the sense of energy efficiency and sustainability – into the operation of the building and of the network.

The contribution of this paper is the concept on how to integrate grid-level control with building-level control and how individual controllers can be designed. Most other approaches do only take a single perspective either from the building or power grid angle of view; the approach proposed here is novel in that it aims to address all stakeholders' needs. This is achieved by coordinating the optimization processes within the buildings as well as on the grid level.

MPC is an attractive method to optimally use available resources with the help of forecasted information (e.g., weather, demand and RES production) and in the presence of constraints. A series of applications of MPC for building and power grid management can be found. Gong et al. propose an

This work is funded by the Austrian Climate and Energy Funds and is conducted within the program "NEUE ENERGIEN 2020".

MPC-based technique to prevent the voltage from collapsing in grid restoration situation [9]. A classical MPC application in building energy management is to control operation of heating, ventilation, and air conditioning (HVAC) in the presence of time-variable energy tariffs ([10], [11], [12]). Even without dynamic pricing, MPC finds its application in minimizing the energy consumption in buildings (see e.g. [12]).

Active control measures supporting renewable integration into distribution grids do however require locally diversified and faster responses than a system based on dynamic pricing could achieve. Therefore this work considers a hierarchical MPC approach that allows not only improvements for efficient building operation but also maximizes the amount of renewables that can be integrated into urban low voltage grids. The goal is to find a global optimum of building and distribution grid operation. Therefore, a hierarchical MPC architecture is proposed to span both domains.

The remainder of this paper is organized as follows: Chapter II introduces the objectives and overall technical concept implemented in the CoOpt project. Chapter III outlines the system models of smart buildings and distribution grid. Chapter IV presents the hierarchic MPC methodology and the co-simulation onsets used to realize and validate the control concept. Chapter V shows first simulation results of a grid MPC implementation, and Chapter VI concludes with a discussion and outlook.

## II. The Project “SMARTCITY GRID: COOPT”

CoOpt aims to develop a Smart City Grid Controller which optimizes the operation of the distribution grid as well as the building’s performance and their active components, see Fig. 1.

In the proposed hierarchical MPC, for each building a separate MPC control is provided, which aims at the local optimization of the building and associated components. One hierarchy level above, the MPC control on grid level aims to optimize the complete system by providing variable set-points and/or adapted objective weights and constraints to the building MPCs. Individual building consumption profiles will be optimized by the building controllers, which are connected to the building management systems (BMS) in order to optimize the operation of the building energy components as shown in Fig. 2.Within the CoOpt project, two low-energy buildings are utilized for modeling, model validation, and hardware-in-the-loop (HiL) test bed to assess control performance compared to their currently implemented state-of-the-art BMS.

### A. Control Problem Statement

Two goal formulations have to be distinguished in the present problem setting: on the building level, the building operators are assumed to be driven by local economic cost objectives where the net power consumption and/or load peaks at the grid interface should be minimized. On the grid level, the grid operators follow more global (economic) objectives, such as load balancing within the grid infrastructure, peak load reduction, or global electric energy loss minimization. These sets of objectives do overlap, but it is anticipated that several sub-goals are in conflict. To improve overall smart building/grid efficiency, both from the building and from the grid perspectives, these relations have to be identified and the savings and load shifting potential quantified. This justifies the development and investigation of a detailed system simulation, including the distribution grid infrastructure and connected buildings with integrated on-site generation, which are driven (controlled) according to distinct, different optimization criteria.

### B. Building Control Objectives & Constraints

- Total load reduction: this objective is already followed and it is not expected to achieve further improvement by the proposed control concept.
- Peak load reduction via load shifting & local consumption
- Guarantee fulfillment of thermal comfort criteria (according to established standards such as EN 15251 [13] and EN ISO 7730 [14])

The building control strategy should exploit available weather and load forecasts/predictions to maximize control performance.

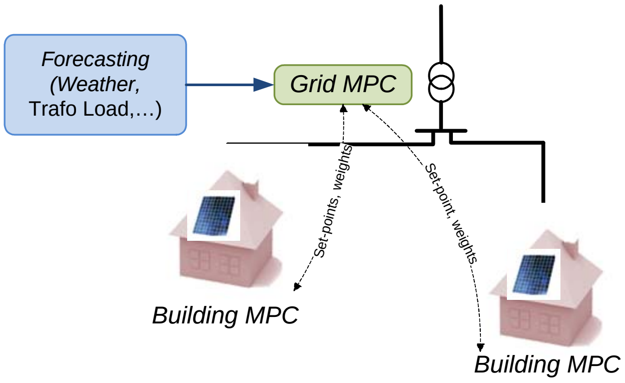


Fig. 1. Schematic view on cooperative grid / building optimization onset

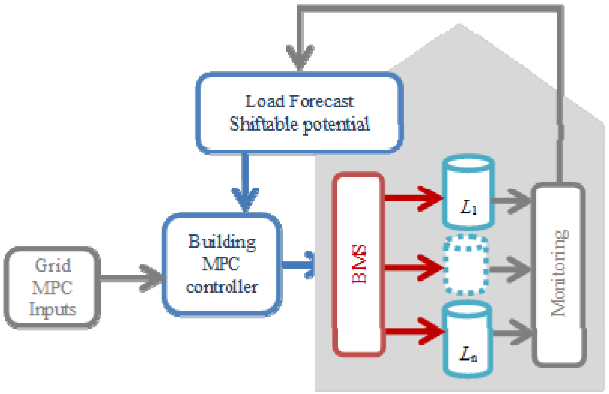


Fig. 2 Conceptual architecture of the hierarchical MPC onset

### C. Power Grid Control Objectives & Constraints

- Reduction of transmission losses
- Peak load reduction via load shifting on the grid scale
- Guarantee fulfillment of electric quality of service criteria (e.g. voltage bands according to established standards such as EN 50160 [15]
- Satisfy technical constraints and obey safety limits (e.g., maximum transformer load)

The grid control strategy should exploit available forecasts and accordingly predicted system responses (building energy demand prediction) to maximize control performance.

## III. System Models

### A. Smart Building Model

The relevant dynamic behavior of the building for the given control task can be divided into the thermal dynamics (room temperatures, thermal capacities, heat inputs and losses) as well as the relevant building services, such as HVAC systems, control subsystems, ventilation, or photovoltaic (PV) systems, see Fig. 3. Modeled subsystems are:

- Zonal thermal dynamics of the building at a fine granularity (approx. 100 thermal states for a mid-sized smart building with 5 floors)
- Building services model describing the heat flow into the thermal sub-system as well as the electric energy consumption of the HVAC system, the concrete core activation system, the associated heat pumps, well pumps, and sub-control loops, as well as the ventilation system and PV contributions.

Models are derived for two low-energy non-residential buildings equipped with BMS and considerable actuation and sensing capabilities. The models are calibrated and validated using monitoring / measurement data of the last years, as well as using indirect validation by matching a validated high-fidelity thermal model available for certain building parts of one of the considered buildings. Where possible, the building services model is also validated against monitoring data, however, recorded data is sparse for these systems. Due to the inherent difficulty in validating such complex systems based on few data records, it is vital to estimate the range of model validity and to quantify the model's uncertainties.

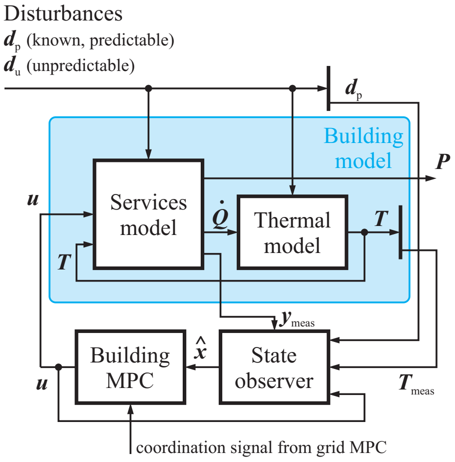

Disturbances
$\boldsymbol{d}_\mathrm{p}$ (known, predictable)
$\boldsymbol{d}_\mathrm{u}$ (unpredictable)
$\boldsymbol{d}_\mathrm{p}$
Building model
$P$
$\boldsymbol{u}$
Services model
$\dot{Q}$
Thermal model
$\boldsymbol{T}$
$\boldsymbol{T}$
$\boldsymbol{y}_\mathrm{meas}$
Building MPC
$\hat{x}$
State observer
$\boldsymbol{T}_\mathrm{meas}$
$\boldsymbol{u}$
coordination signal from grid MPC

Fig. 3. Smart building model predictive controller structure for a building modeled by a services and a thermal part

Finally, the model is represented in the form of interconnected (generally non-linear) function blocks, with the following minimum set of inputs and outputs for the MPC:

- Non-controlled heat and electric loads as well as environment parameters (considered as predictable disturbance inputs $\boldsymbol{d}_\mathrm{p}$ and as unpredictable disturbance inputs $\boldsymbol{d}_\mathrm{u}$)
- Control input signals $\boldsymbol{u}$ for MPC-controlled devices and sub-systems (heat pump demand, set-points for subsystem controllers)
- Measured outputs (temperature measurements $\boldsymbol{T}$), total power demand of the building $P$

It is evident that an extensive amount of input data must be generated to parameterize the model (weather conditions, heat and electric loads). Also, these signals need to be predicted and made available to the MPC during operation so that the predictive model inside the MPC can be parameterized. While on-line data collection of weather-prediction data is straightforward given access to corresponding meteorological sources, the prediction of heat and electric load profiles must be performed specifically for the considered building.

### B. Distribution Grid Model

Several smart grid model variants are derived to represent various grid characteristics (residence-dominated, commerce-dominated, and mixed). The goal is to investigate slow dynamics of the smart grid participants (sampling time $T_\mathrm{s} = 15$ min), so all electric relations are modeled in a static fashion. The central modeling onset is to express the nodal voltage drop vector $\Delta\boldsymbol{V}$ to the supply voltage $V_0$ as a function of the grid current vector $\boldsymbol{I}$, coupled by the linear impedance relation

$$\Delta\boldsymbol{V} = \boldsymbol{Z}\,\boldsymbol{I}, \qquad (1)$$

Where $\boldsymbol{Z}$ is the grid impedance matrix, and $\Delta\boldsymbol{V} = V_0 - \boldsymbol{V}$ with the grid node voltage vector $\boldsymbol{V}$. The grid furthermore consists of supplier/consumer nodes (sources/sinks), which are modeled through time-variant power demand/production. Producers are the transformer node(s) and local producers, such as PV systems or smart buildings, and consumers are distinguished into passive consumers (non-controllable) and active participants (which can be controlled or affected in their behavior) such as smart buildings or controllable transformers. Each smart building can thus take either a producer or a consumer role over time. Fig. 4 shows the grid MPC structure.

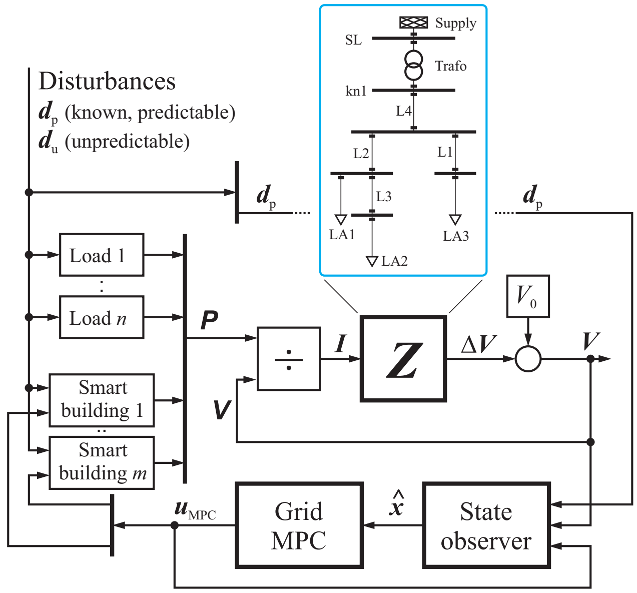


Fig. 4. Smart grid model predictive controller structure for a smart grid consisting of a grid model, passive loads, and smart participants

## IV. Methodology

### A. Model Predictive Control (MPC) Formulation

Buildings with large thermal capacities are good candidates for demand side management since variations in cooling or heating power have negligible effect on interior temperature in the short term. However, predictive optimization is required in the long term to effectively utilize the thermal energy storage and to keep the temperature within specified limits. Model predictive control is a promising method to perform building control with energy storage management. Experimental test results from MPC-based building cooling and heating control presented in [16] and [17], respectively, demonstrate significant energy savings. If weather conditions and building usage are taken in to account either by using weather forecasts [18] or by exploiting daily periodicity [19], additional improvements can be expected.

In order to leverage the full potential of load shifting, a coordination of buildings is necessary on the grid level. Simulation studies with MPC for building control based on dynamic energy pricing is carried out in [20], but without feedback to the electricity price. In [21] also the effect of load shifting on a load-based tariff is considered. The economic MPC proposed in [20] requires extensive information exchange between producers and consumers and is thus not practical.

The onset chosen in this work employs a hierarchic architecture of state-of-the-art model predictive control (MPC) of smart buildings as well as of the distribution grid, and aims to study possible interfaces and characterize the interaction between these MPC layers.

Detailed models of a smart building or the smart grid, respectively, are being developed to represent the relevant dynamics accurately as presented below. While time-domain simulation of this dynamics is possible, these models do not lend themselves to direct inclusion into the MPC. Thus, a simplified version of the detail models must be derived to serve as prediction models. These are represented in standard state-space representation of time-varying dynamic discrete-time systems:

$$\boldsymbol{x}_{k+1} = f(k, \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}_k), \tag{2}$$

With state vector $\boldsymbol{x}_k$, vector of manipulated control variables $\boldsymbol{u}_k$ and vector of disturbance inputs $\boldsymbol{d}_k$. The index $k$ denotes equally spaced time samples with specified interval $T_s$ = 15 min. For a horizon of $N$ = 192 samples (corresponding to a 48 hour prediction window), the considered general MPC problem formulation is

$$\min_{\boldsymbol{U}_k} \quad J = J(k, \boldsymbol{x}_k, \boldsymbol{U}_k, \boldsymbol{D}_k) \qquad \text{... objective}$$

such that

$$\begin{aligned} &\boldsymbol{x}_{k+i+1} = f(k, \boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}, \boldsymbol{d}_{k+i}) && \text{... dynamics} \\ &\boldsymbol{g}(k, \boldsymbol{x}_k, \boldsymbol{U}_k, \boldsymbol{D}_k) = \boldsymbol{0} && \\ &\boldsymbol{h}(k, \boldsymbol{x}_k, \boldsymbol{U}_k, \boldsymbol{D}_k) \leq \boldsymbol{0} && \text{... constraints} \end{aligned} \tag{3}$$

where $\boldsymbol{U}_k = [\boldsymbol{u}_k^T \ \boldsymbol{u}_{k+1}^T \dots \boldsymbol{u}_{k+N-1}^T]^T$ denotes the sequence of future control inputs and $\boldsymbol{D}_k = [\boldsymbol{d}_k^T \ \boldsymbol{d}_{k+1}^T \dots \boldsymbol{d}_{k+N-1}^T]^T$ denotes the sequence of predicted disturbances. Constraints are usually stated (approximated) as linear inequality constraints in the form of

$$\boldsymbol{A}_k \boldsymbol{x}_k \leq \boldsymbol{b}_k \tag{4}$$

The chosen objectives and constraints for building and grid, respectively, are outlined in the following.

*1) Smart Building Control Problem:* Two chosen objectives are to minimize the absolute building net power consumption $p = \Sigma\, p_k$ and the peak load $p_{\text{max,B}} = \max(p_k)$:

$$J_B(u_j,t) = \alpha\, p + (1-\alpha)\, p_{\max} \tag{5}$$

The set of constraints is comprised of requirements on thermal comfort in selected rooms (admissible temperature intervals, i.e. state constraints), constraints on thermal storage, and technical constraints of building services states and signals, as well as on the chosen control inputs. The actual formulation of these constraints (including the selection of appropriate, representative building sections) needs to be delayed until the detailed building model has been developed and validated.

*2) Smart Grid Control Problem:* Two chosen objectives areto minimize total electric transmission losses $\Delta P_L = \boldsymbol{I}^T\boldsymbol{Z}\boldsymbol{I}$ and relative peak loads $p_{\text{max,G}} = \max(p_{\text{max,B},k}/p_{\text{ref,B},k})$ (goal attainment methodology, see *[22]*):

$$J_G(v_j,t) = \beta\, \Delta P_L + (1-\beta)\, p_{\text{max,G}} \tag{6}$$

The constraints entail the admissible current and voltage bands (quality requirements and safety limits) at selected critical nodes and algebraic constraints implementing the grid balance equations. Also, the grid control problem needs to include each smart building's closed-loop dynamics to a simplified degree to model their actuation of the grid system and use the smart buildings as system actuators. This hierarchic onset and the inter-MPC communication ideas are sketched below.

### B. *MPC Hierarchy and Inter-MPC Communication Onsets*

A decentralized, hierarchical MPC approach is desired due to the following general conditions:

- Deep knowledge of all involved subsystems is required for centralized MPC, leading to intractable complexity for control of larger grid sections.
- Building operators should keep sovereignty over their own control systems.
- Information exchanged between building- and grid-level control should be kept to a minimum not only due to limited communication bandwidth but also to avoid transmitting sensitive personal or corporate data such as occupation profiles.

In [23], a number of distributed MPC architectures are reviewed and classified. Decentralized control with local optimization tends to move towards a Nash equilibrium that may be unstable and far away from the global optimum, even in the case of full communication between local controllers. Convergence towards a Pareto optimum can be achieved by distributed or centralized coordination [24], [25], [26]. Thus, an onset with hierarchical MPC for coordination (in the sense of [23]) is chosen for CoOpt in contrast to decentralized MPC (e.g. [27]) or distributed MPC with full communication (e.g. [20]). A central MPC on the grid level coordinates the building MPCs by means of modifying their cost functions (e.g. via variable pricing) based on optimization with simplified models of the buildings (see Fig. 5). An example of such a hierarchy is reported in [28], where consumers (i.e. buildings) are simply modeled as energy storage systems that report bounds on their charge/discharge rates to the central coordinator.

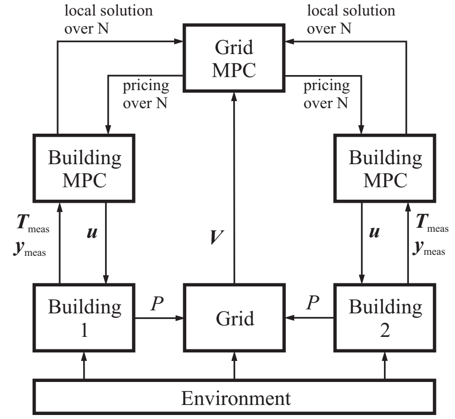


Fig. 5. Hierarchic interaction/coordination between grid and building MPCs

### C. *Co-Simulation framework*

One of the key aspects of this work is to implement the developed framework for co-simulation of power networks, building and MPC controller by DIgSILENT/PowerFactory, Dymola and MATLAB respectively. Since there is no universal tool to analyze the whole energy system, each individual component has been simulated with related, specific software packages.

Among available standard communication protocols like OPC, FMI, tool-specific APIs, or even IEC 61850, etc., OPC was chosen as a low-complexity communication interface enabling tool interoperability [29].

## V. First Results

Simulation of a ficticious smart grid with 3 passive loads and 2 smart buildings with controllable energy storage. A generic building model (bounded storage state, control input is the power consumption differing from simple storage-less operation) is utilized as an interface to the building controller (not considered yet).

Grid MPC minimizes transmission losses, based on load predictions and considering constraints on building storage levels, control signals, grid currents, and voltage drops.

Load trajectories and predictions have been derived from scaled, standard residential and office load profiles.

This grid MPC concept has been implemented using MATLAB and the YALMIP optimization framework. Prediction and control horizons are chosen as 192 samples (48 hours). The receding-horizon optimization could be carried out in less than 5 sec/sample on a modern desktop PC.

The results (see Fig. 6) give a qualitative insight into the proposed load shifting scheme and allow further investigation of potential problems such as

- the treatment of infeasibility of the optimization problem,
- the effect of errors in load prediction, and
- the amount of required storage to achieve a significant reduction of transmission losses.

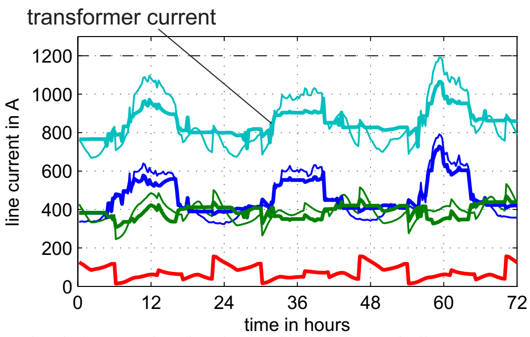


Fig. 6. Simulation results showing reduction in peak line currents (bold: with proposed grid MPC; thin: no load shifting). Here: *N*=96 samples

## VI. Conclusion and future work

This paper proposes a novel solution to address not only the energy performance in individual buildings but also the global energy and environmental problem of the distribution grid. Since smart grids and renewable energy sources came into wide use, the importance of such hierarchical control systems increases as they potentially enable an efficient trade-off in this multi-domain optimization problem.

One particular aspect to be considered is the assessment of stability of the overall grid/buildings closed-loop system. Potential threats to stability are (i) competition between different controlled buildings due to grid coupling, and (ii) conflicting goals between controlled buildings and grid coordination. Hence, stability is not guaranteed a priori for the proposed concept, but careful design of the coordination mechanism is expected to ensure stability in practice which has to be verified in detailed multi-domain co-simulation studies.

Preliminary studies show that suitable control-, coordination-, and co-simulation methods and tools exist; however, they need to be adapted and extended to fit this particular problem setting. The Hardware-in-the-Loop validation will help quantify actual control system performance to a high degree of realism.

## References

[1] M. Braun, G. Arnold, and H. Laukamp, “Plugging into the Zeitgeist,” *IEEE Power and Energy Magazine*, vol. 7, no. 3, pp. 63 –76, Jun. 2009.

[2] A. M. J. C. Hernández, “Impact comparison of PV system integration into rural and urban feeders,” *Energy Conversion and Management*, no. 6, pp. 1747–1765.

[3] P. Palensky and D. Dietrich, “Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381 –388, Aug. 2011.

[4] K. Park, Y. Kim, S. Kim, K. Kim, W. Lee, and H. Park, “Building Energy Management System based on Smart Grid,” in *Telecommunications Energy Conference (INTELEC), 2011 IEEE 33rd International*, 2011, pp. 1 –4.

[5] J. K. Kok, C. J. Warmer, and I. G. Kamphuis, “PowerMatcher: multiagent control in the electricity infrastructure,” in *Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2005, pp. 75–82.

[6] P. Zhao, S. Suryanarayanan, and M. G. Simões, “An Energy Management System for Building Structures Using a Multi-Agent Decision-Making Control Methodology,” in *2010 IEEE Industry Applications Society Annual Meeting (IAS)*, 2010, pp. 1 –8.

[7] S. Kiliccote, M. A. Piette, E. Koch, and D. Hennage, “Utilizing Automated Demand Response in commercial buildings as non-spinning reserve product for ancillary services markets,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 4354 –4360.

[8] N. J. Hewitt, “Heat pumps and energy storage – The challenges of implementation,” *Applied Energy*, vol. 89, no. 1, pp. 37–44, Jan. 2012.

[9] B. Gong and I. A. Hiskens, “Two-stage model predictive control for voltage collapse prevention,” in *Power Symposium, 2008. NAPS ’08. 40th North American*, 2008, pp. 1 –7.

[10] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jorgensen, “Economic Model Predictive Control for building climate control in a Smart Grid,” in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, 2012, pp. 1 –6.

[11] Y. Zong, D. Kullmann, A. Thavlov, O. Gehrke, and H. W. Bindner, “Application of Model Predictive Control for Active Load Management in a Distributed Power System With High Wind Penetration,” *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 1055 –1062, Jun. 2012.

[12] A. Kelman, Y. Ma, and F. Borrelli, “Analysis of local optima in predictive control for energy efficient buildings,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 5125 –5130.

[13] *Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics*. European Committee for Electrotechnical Standardization, EN 15251:2007, 2007.

[14] *Ergonomics of the thermal environment*. European Committee for Electrotechnical Standardization, EN ISO 7730:2005, 2005.

[15] *Voltage characteristics of electricity supplied by public electricity networks*. European Committee for Electrotechnical Standardization, EN 50160:2011, 2011.

[16] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, “Model Predictive Control for the Operation of Building Cooling Systems,” *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 3, pp. 796 –803, May 2012.

[17] S. Prívara, J. Široký, L. Ferkl, and J. Cigler, “Model predictive control of a building heating system: The first experience,” *Energy and Buildings*, vol. 43, pp. 564 – 572, 2011.

[18] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, “Use of model predictive control and weather forecasts for energy efficient building climate control,” *Energy and Buildings*, vol. 45, no. 0, pp. 15 – 27, 2012.

[19] R. Gondhalekar and J. Imura, “Least-restrictive move-blocking model predictive control,” *Automatica*, vol. 46, no. 7, pp. 1234 – 1240, 2010.

[20] T. G. Hovgaard, K. Edlund, and J. B. Jorgensen, “The potential of Economic MPC for power management,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 7533 –7538.

[21] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, and M. Morari, “Reducing peak electricity demand in building climate control using real-time pricing and model predictive control,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 1927 – 1932.

[22] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek, “Robust H-infinity control design parameter optimization via genetic algorithm for lateral control of a BWB type aircraft,” in *Intelligent Control Systems*, 2010, vol. 8, pp. 57–63.

[23] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Control – A review,” *Journal of Process Control*, vol. 19, no. 5, pp. 723 – 731, 2009.

[24] A. N. Venkat, J. B. Rawlings, and S. J. Wright, “Stability and optimality of distributed model predictive control,” in *Decision and Control, CDC-ECC 2005. 44th IEEE Conference on*, 2005, pp. 6680 – 6685.

[25] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, “Distributed MPC Strategies With Application to Power System Automatic Generation Control,” *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 6, pp. 1192 –1206, Nov. 2008.

[26] J. B. Rawlings and B. T. Stewart, “Coordinating multiple optimization-based controllers: New opportunities and challenges,” *Journal of Process Control*, vol. 18, no. 9, pp. 839 – 845, 2008.

[27] F. Oldewurtel, A. Ulbig, M. Morari, and G. Andersson, “Building control and storage management with dynamic tariffs for shaping demand response,” in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2011, pp. 1 –8.

[28] J. Bendtsen, K. Trangbaek, and J. Stoustrup, “Hierarchical model predictive control for resource distribution,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 2468 –2473.

[29] F. Andren, M. Stifter, T. Strasser, and D. Burnier de Castro, “Framework for co-ordinated simulation of power networks and components in Smart Grids using common communication protocols,” in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 2011, pp. 2700 –2705.

# Integrated Coordination of AC Power Flow Controllers and HVDC Transmission by a Multi-Agent System

Sven C. Müller, *Student Member, IEEE,* Ulf Häger, and Christian Rehtanz, *Senior Member, IEEE*

***Abstract*—Due to highly volatile feed-in by renewable energy sources (RES), a large amount of potential controls in future smart grids as well as the striving for gains in efficiency, power system operation becomes increasingly complex. This complexity is a particular challenge for real-time operation as centralized optimizations of the non-linear system are computationally intensive and rely on adequate data sets. The application of distributed control systems as a complement or substitute for certain centralized control decisions offers advantages regarding adaptiveness, robustness and real-time performance. In this paper, the progress on developing a distributed coordination system for power flow control by HVDC links and AC power flow controllers is presented. Currently, the coordination of power flow controllers is performed manually from a control center based on time-consuming centralized calculations. Due to its decentralized structure, the proposed coordination system enables a robust real-time response. This benefits operational security as counteractions can be initiated immediately in case of unforeseen events and enables the allocation of more transmission capacity in congested network areas.**

***Index Terms*—congestion management, distributed control, HVDC, multi-agent systems, power flow control.**

## I. Introduction and Background

Transmission system operators (TSOs) need to apply adequate congestion management methods in order to efficiently master the trade-off between offering transmission capacity at the electricity market (resulting in increased social welfare due to market integration and price convergence between different price zones) and ensuring system stability. For the latter, it needs to be assured that overloading of equipment (e.g., a transmission line or a transformer) is avoided or quickly mitigated even in case of contingencies (e.g., loss of equipment or power plants) because overloads could cause the disconnection of equipment which could in turn lead to a cascade of tripping transmission lines.

Power flows in the transmission system derive from:

i. the configuration of generation (power feed-in at network nodes);
ii. the configuration of loads (power consumption at network nodes); and
iii. the transmission system topology (characteristics and configuration of network branches).

Generally, avoidance of overloads is already addressed in the operational planning and only generation schedules are allowed in the day-ahead market clearing process for power plants that do not cause overloads even in case of any single contingency (N-1 security). However, these schedules are based on forecasts and deviations from the projected system state and thus overloading can occur. During operation, the loading of equipment can generally be alleviated by:

i. changing the load and generation configuration ('redispatch'); or
ii. taking topological actions, in particular deploying power flow controlling devices (in the following generally termed Power Flow Controllers (PFCs)), if available.

TSOs are requested by regulators to exploit topological actions before taking redispatch actions. For this reason, we primarily focus on the operational use of PFCs in the following. PFCs include equipment such as Phase Shifting Transformers (PSTs), High-Voltage Direct Current (HVDC) technology, and Flexible AC Transmission System (FACTS) devices. PSTs and power flow controlling FACTS devices (both in the following termed 'AC PFCs') are installed in series with an AC transmission line and are able to control the power flow on this branch, e.g., by changing transformer tap settings. Thereby, adjacent power flows are impacted. These devices are often installed in structurally congested regions, in particular at cross-border transmission corridors, and in several cases (e.g., in the Benelux region or at the border between France and Spain) multiple PFCs are installed which impact the power flows in overlapping areas. In order to avoid counter-productive or overcompensating actions of these PFCs, coordination between them is needed. Until now, PFC settings are agreed upon by the TSOs in the operational planning (day-ahead) and during operation by phone calls or suggestions by coordination service centres like CORESO [1]. These agreements are based on central network calculations (e.g., optimal power flow (OPF) methods) or expertise of operators. Due to limited data exchange between different TSOs and significant OPF computation times, a coordinated change of PFC settings based on central calculations cannot be executed before several minutes have passed.

Nonetheless, it is desirable to enable real-time response of the coordination system due to two reasons: On the one hand, the flexibility of reacting in real-time can support system stability by adaptively and automatically counteracting emergency situations, in particular in case of N-1 situations

This work was supported by the German Research Foundation DFG as part of research unit FOR1511 (http://www.for1511.tu-dortmund.de).

S. C. Müller, U. Häger and C. Rehtanz are with the Institute of Energy Systems, Energy Efficiency and Energy Economics, TU Dortmund University, Dortmund, Germany (e-mail: see http://www.ie3.tu-dortmund.de).

or higher. On the other hand, real-time response enables to account for the execution of corrective actions in the planning process, thus strict N-1 planning can be relaxed and thereby available transmission capacity for the coupling of electricity markets can be increased [2]. In order to allow the coordination system to counteract in emergency situations, the requirement for reaction time is given by the protection system settings. Common timer settings for zone three distance protection relays are in the range of $2s$ [3]. Among potential coordination methods it can be distinguished between two general concepts:

i. coordination for determining PFC settings based on (mostly centralized) OPF methods ([4], [5], [6], [7]); and
ii. distributed coordination based on simple decentralized algorithms and being independent of central entities (first approach in [8]).

Proposed OPF based coordination methods can provide essential data for the operational planning but as convergence is not guaranteed and as they require long computation times, a predefined set of contingency scenarios as well as system-wide data, they are not suitable for real-time coordination of PFCs in the current technological and organizational framework. In a centralized structure the collection of data and the state estimation alone lasts several seconds for power systems with many hundreds of nodes. On top of this comes the computation time for the OPF calculation. In total this is too time consuming for fast protection functions (compare timer setting of protection relays above). Distributed control, on the other hand, can react quickly, even in case of N-2 scenarios and higher, and it provides significant benefits by its real-time adaptiveness and robustness. For these reasons, the focus is set on developing a fully distributed control system for PFCs in the following.

In this paper, we first present our approach for the distributed coordination of PFCs based on a Multi-Agent System (MAS) in section II. Second, exemplary simulation results for the coordination of AC PFCs and HVDC links are presented. In section IV, we comment on some key aspects of the approach, discuss potential further developments and address the extension of the coordination to close to real-time redispatch. Finally, we close with a conclusion and an outlook.

## II. Agent-Based Coordination of AC Power Flow Controllers and HVDC Transmission

In this section, we present an overview of the agent-based coordination of PFCs followed by detailed outlines of the process steps. The section integrates main aspects of the distributed control concept for PFCs presented in [9], [10] and [11] and details the integration of a new type of active agents for HVDC links.

### A. Overview

For our concept, we assume to install software agents at the substation level for each serial network element (most importantly AC transmission lines, AC PFCs and HVDC links) in a relevant area which are enabled to communicate with each other as well as to execute rather simple decentralized algorithms for autonomous decision making. By this, the complex coordination problem becomes scalable also for a potentially higher degree of power flow controlling capabilities in future smart grids.

Generally, it can be distinguished between three kinds of agents:

- active AC agents (installed at AC PFCs);
- active HVDC agents (installed at HVDC links); and
- passive agents (installed for other serial network devices).

Both active and passive agents send messages to their direct neighbours that contain current state information about their corresponding device and also forward the information of messages received. Based on this information about the devices in the area, active agents carry out a system analysis, perform decision making in order to ensure coordinated action of all PFCs, and finally take action by changing a PFC setting. Fig. 1 gives an overview on the main steps executed by an active agent and the methods applied which will be described in more detail in the following subsections. The MAS has been implemented in JAVA and an interface for retrieving network measurements as well as for triggering events has been developed as presented in [12].

### B. Agent Communication

A main feature of the agent-based approach is the distributed provision of updated topological information to decision making entities which are located at the decentralized level. Due to this, the system state information is updated rapidly after a change in the system (e.g., disconnection of a line) and the controllable elements (PFCs) can react adaptively to this new system state. To enable this, agents generate regularly (e.g., every 10 ms) so called *StateInformMessages* and send them to their direct neighbours. Each time an agent receives a *StateInformMessage* it adds information about the current device and forwards them to all neighbouring agents (visualized in Fig. 2, for a modification of the forwarding process for HVDC agents see subsection II-C).

The *StateInformMessage* contains the following information:

- loading of the device;
- direction of power flow through the device;
- impedance of the device;
- time stamp; and
- information of preceding forwarders.

As the network is flooded with messages, it needs to be ensured that the messages are discarded under certain conditions in order to avoid congestion in the communication network. We introduced two conditions for discarding that can be set depending on the investigated network: a *maximum hop limit* (maximum number of forwarding of a message) and a *maximum impedance limit*. The last criterion corresponds to the fact that if the sum of the impedances of forwarders already passed through exceeds a certain limit the message can be discarded as this path is of low significance for the sensitivity of a device.

### C. Topology Analysis

Active agents evaluate the collected *StateInformMessages* and perform a distributed topology analysis at a certain

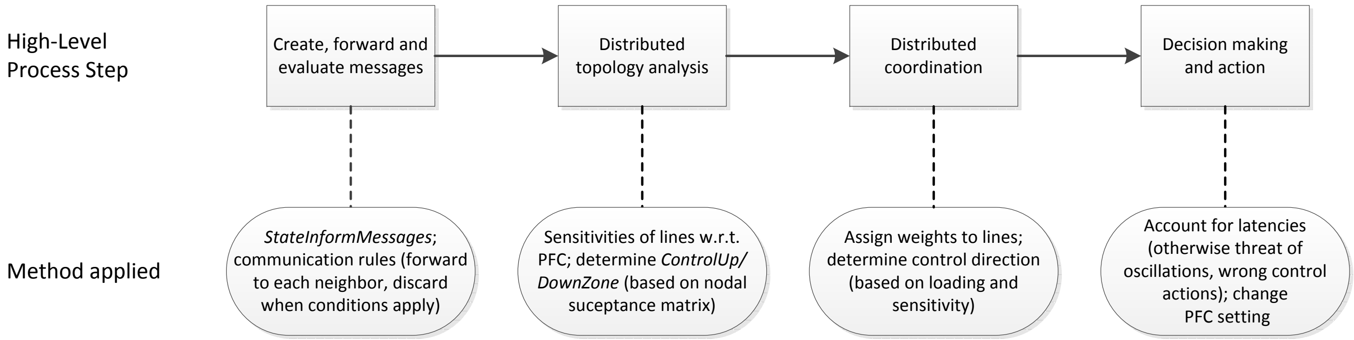

High-Level Process Step
Create, forward and evaluate messages
Distributed topology analysis
Distributed coordination
Decision making and action
Method applied
StateInformMessages; communication rules (forward to each neighbor, discard when conditions apply)
Sensitivities of lines w.r.t. PFC; determine ControlUp/ DownZone (based on nodal suceptance matrix)
Assign weights to lines; determine control direction (based on loading and sensitivity)
Account for latencies (otherwise threat of oscillations, wrong control actions); change PFC setting

Fig. 1: Overview on active agents' process steps and applied methods

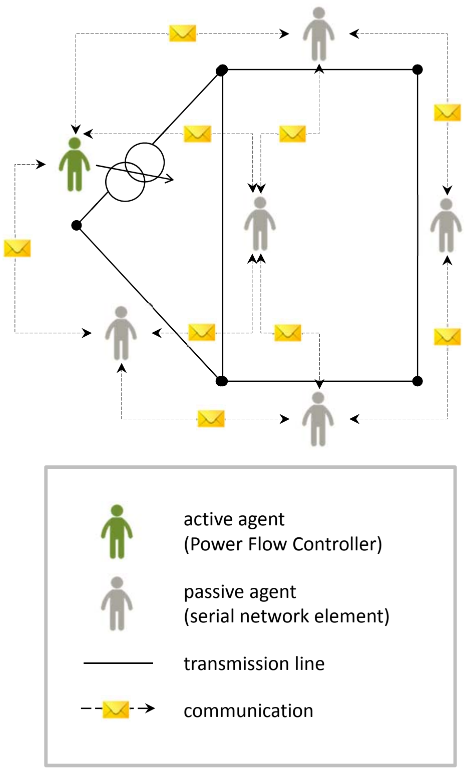


Fig. 2: Communication among agents

frequency (e.g., every 10m). In this step, agents determine sensitivities of transmission lines with respect to a control of the PFC corresponding to the agent and group the lines as either belonging to a *ControlUpZone* or *ControlDownZone*.

For determining the sensitivities of network devices with respect to PFC controls, the agent evaluate a nodal susceptance matrix based on the impedances $X_L$ of surrounding network devices known from the *StateInformMessages*. The matrix is set up as follows:

$$\underline{B} = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix} \tag{1}$$

$$b_{ii} = \sum_k -\frac{1}{X_{\mathrm{L},ik}} \tag{2}$$

$$b_{ik} = \frac{1}{X_{\mathrm{L},ik}}; i \neq k \tag{3}$$

The first two rows and columns of $\underline{B}$ correspond to the buses of the PFC corresponding to the agent. For each additional network device known to the agent an additional row and column is inserted. Next, we determine the sensitivity $sens$ of a transmission line $l$ with respect to a control action of a PFC $c$ as the change of power flow through $d$ divided by the change of power flow through $c$:

$$\mathrm{sens(l,c)} = \frac{\Delta P(\mathrm{l})}{\Delta P_{\mathrm{PFC}}(\mathrm{c})} \tag{4}$$

The change of power $\Delta P$ can be approximated by Direct Current (DC) load flow analysis based on the nodal susceptance matrix. Use of the DC load flow provides the advantages of being computationally inexpensive (linear equations) and of converging always in the first iteration (in contrast to AC load flow). This substantially contributes to the robustness of the control system. For representing a change of the PFC setting, we introduce a loop flow $\underline{p}$ between the two buses of the PFC, observe the change of voltage angle $\underline{\delta'}$, and can thus determine the sensitivity $sens(l, c)$ for a line $l$ situated between nodes $i$ and $j$ as follows:

$$\underline{p} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5}$$

$$\underline{\delta'} = \underline{B}^{\dagger} \cdot \underline{p} \tag{6}$$

$$\mathrm{sens(l,c)} = \frac{1}{X_{\mathrm{L},ik}} \cdot \delta'_i - \frac{1}{X_{\mathrm{L},ik}} \cdot \delta'_k \tag{7}$$

Here, $\underline{B}^{\dagger}$ represents the pseudoinverse of the nodal susceptance matrix $\underline{B}$ and $\delta'$ the change of the voltage angle. At this point it is important to account for the technological characteristics of HVDC technology in order to enable correct execution of the distributed topology and sensitivity analysis. From the perspective of the AC system, the impedance of the HVDC line is irrelevant for the distribution of power flows, instead the link can be interpreted as a branch with an almost perfectly controllable power flow through it. The sensitivities of AC elements with respect to the HVDC control can be calculated by the nodal susceptance matrix and the induced loop flow as described above. On the other hand, the HVDC link exhibits no sensitivity to the control actions of AC PFCs as it forms a decoupled part of the system. As a consequence for the communication among agents, no messages should be forwarded along the HVDC link as only transmission paths of the AC system matter for the analysis of sensitivities.

Next, the transmission lines are categorized as belonging to the *ControlUpZone* or *ControlDownZone*. Belonging to the *ControlUpZone* means that a positive increment of the PFC set-point would decrease the loading of the line, and vice versa for the *ControlDownZone*. This categorization can be concluded from the sign of the sensitivity $sens$ and the sign of the direction of the power flow on the line (which is part of the information of the *StateInformMessage*). If both signs are equal, a line belongs to the *ControlUpZone*, otherwise to the *ControlDownZone*.

### D. Distributed Coordination

As discussed above, the aim of the control system is to decrease the loading of overloaded lines by coordinated actions of PFCs which redirect power flows to paths with lower loading. An agent corresponding to a PFC has a set of three actions it can choose from: increase the control set-point, decrease the set-point, or keep the current set-point. Relevant control set-points are, e.g., the tap position of a PST or the controller set-point for transmitted active power over an HVDC link. For determining the direction of a set-point modification, the controlling agent assigns weights $f_{weight}(l)$ to all known transmission lines $l$. The line with the highest weight determines the direction of control, thus if the highest weighted line belongs to the *ControlUpZone* the tap position is increased by the agent, and if it belongs to the *ControlDownZone* the tap position is decreased. If the difference of the highest weights of lines in the *ControlUpZone* and the *ControlDownZone* is within a certain deadband, no alteration of the set-point is executed in order to avoid oscillations and to ensure that the control action is not counter-productive as it would decrease loading on one critical device but would simultaneously increase stress on another similarly weighted device.

Obviously, an adequate method for the assignment of weights to the lines is critical. The idea is to give a line an increasing weight in the decision making process along with (a) a more critical loading (thus, if two lines show the same sensitivity with respect to the control action of a PFC, the PFC acts in order to relieve the stress of the line with the more critical loading) and (b) a higher sensitivity (e.g., if one line becomes critically loaded this issue is primarily resolved by the PFC with the largest impact on the line). Below, the function for assigning weight to transmission lines is given:

$$f_{\mathrm{weight}}(\mathrm{l}) = \begin{cases} & ,(high \leq \mathrm{load}(\mathrm{l}) \leq crit) \\ f_{\mathrm{load}}(\mathrm{l}) \cdot \mathrm{sens}(\mathrm{l},\mathrm{c}) & \wedge(\mathrm{sens}(\mathrm{l},\mathrm{c}) > f_{\mathrm{sens}}(\mathrm{l})) \\ & \vee(\mathrm{load}(\mathrm{l}) > crit) \\ 0 & , else \end{cases} \tag{8}$$

The general set-up of the weighting function is illustrated as block diagram in figure 3. The weighting function performs a multiplication of the two decisive input values for the execution of control actions, which are the loading $load(\mathrm{l})$ and the sensitivity $sens(\mathrm{l},\mathrm{c})$. Furthermore, filters are used to cutoff non-relevant input values for the control action. The idea of these filters is derived from electronic filters, which cutoff certain frequency domains from an input signal (e.g. low-pass filters and high-pass filters) [13].

The filter for the device loading (Fig. 4) behaves as a high-pass filter to filter out loadings which are significantly below the thermal limit of the device. Furthermore, it also includes a non-linear amplification of high loadings, in order to achieve a stronger weighting the higher the overloading of the device. In addition, devices which are not overloaded should only be controlled precautionary, if the sensitivity of the PFC on this device as well as the loading of the device are sufficiently high. This function is implemented by another filter (Fig. 5) in combination with a logical block whose output signal is either 0 or 1.

Besides the two functions, $f_{load}(l)$ and $f_{sens}(l)$, the weighting function features the parameters $high$ and $crit$. The parameter $high$ indicates a level of loading of a line (e.g., $high = 90\%$) when first preventive control actions could be taken in case the PFC has a high impact on the line, whereas $crit$ indicates a critical level of loading (e.g., $crit = 100\%$) that justifies to take actions into account regardless of a high sensitivity. Fig. 4 shows an example for $f_{load}(l)$ and Fig. 5 an example for $f_{sens}(l)$. For more details on the functions in the weighting process the reader is referred to [10].

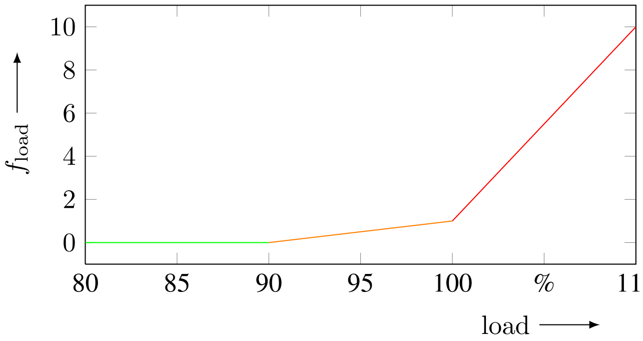


Fig. 4: Loading function for the valuation of the line loading at the example of $high = 90\,\%$ and $crit = 100\,\%$

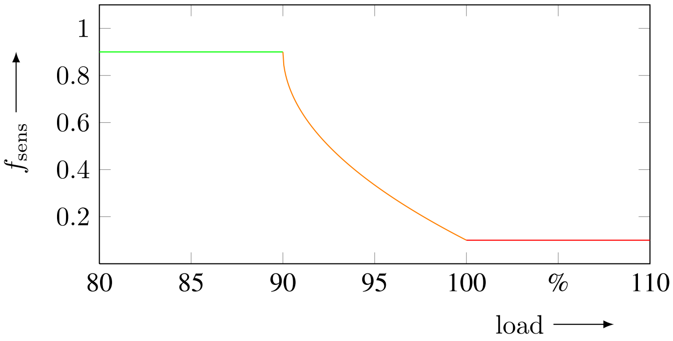


Fig. 5: Sensitivity function for the consideration of devices depending on their loading at the example of $high = 90\,\%$ and $crit = 100\,\%$

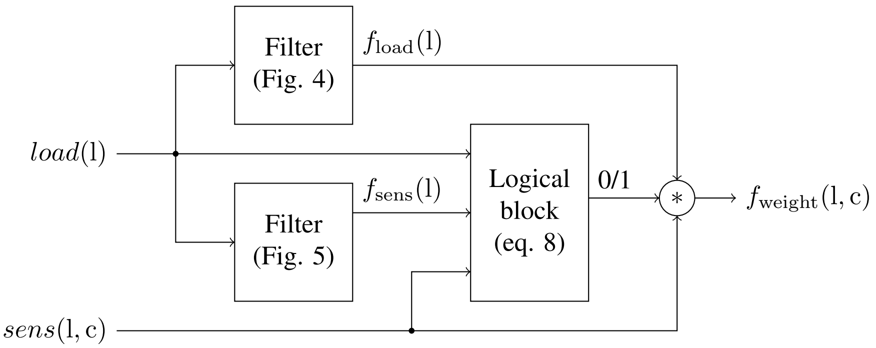

Filter
(Fig. 4)
$f_{\text{load}}(\text{l})$
$load(\text{l})$
Logical
block
(eq. 8)
0/1
$*$
$f_{\text{weight}}(\text{l}, \text{c})$
$f_{\text{sens}}(\text{l})$
Filter
(Fig. 5)
$sens(\text{l}, \text{c})$

Fig. 3: Block diagram of the weighting function

### E. Decision Making and Control Action

After assigning weights to the lines, an active agent would change the corresponding PFC set-point in order to decrease the loading of the line with the highest weight (unless the dead-band criterion holds, see subsection II-D). However, in order to avoid wrong initial control actions and oscillations between PFC set-points, it must be accounted for communication latencies. For this reason, after a change of loading (greater than a certain threshold) is detected, the agent is only allowed to change tap positions after it has received updated information from all other network devices because the change of loading might be due to a change of topology. A change of topology could lead to a reversal of sensitivities of transmission lines and could thus cause wrong initial control actions if old data was used. After waiting for updated information and evaluation of the system state, the agent can initiate the change of a PFC setting. In case of a PST, incrementing or decrementing the tap position by one tap takes considerable time (e.g., $6s$), so it has to be waited for an amount characteristic for the type of PFC before the next action can be initiated.

## III. Simulation Results

In this section, exemplary simulation results in the New England Test System (IEEE 39-bus 10-machine system) extended by four AC PFCs and one HVDC line as presented in [11] are discussed (see Fig. 6).

The AC PFCs are modelled as Thyristor Controlled Phase Angle Regulators (TCPARs, from the family of FACTS devices) which are controlled by power electronics with stepwise alteration of the injected voltage by 0.12%. The active power transmission over the HVDC link is controlled by the corresponding agent in steps of $\Delta P = 2MW$. A minimum waiting time between two actions of $0.1s$ is set for all active agents (compare also subsection IV-A). In order to examine the performance of the control system in a case worse than N-1, the exemplary scenario is set up as follows: at $t = 1s$ line TL0304 is disconnected, followed by an outage of the load at node 15 at $t = 20s$. At $t = 25s$ line TL0304 is reconnected.

Fig. 7 depicts the main results of the simulation. First, Fig. 7-(a) shows the loading of three relevant transmission lines in the scenario described above but without the activation of the MAS. Line TL0405 gets loaded above 100% with the loss of line TL0304 ($t = 1s$) and keeps being overloaded until the loss of the load ($t = 20s$) alleviates the stress on the line. The stress on line TL0414 is increased by both outages, temporarily with a loading above 100%, and finally alleviated by the reconnection of the line. The loading of line TL1718 is reduced by the loss of the line but then increased by the load outage and the reconnection of the line. Significant power swings can be noted which are typical in the New England Test System. The extent of this oscillating system behaviour can be reduced by an improved system damping, e.g., by the implementation of Power System Stabilizers (PSS), which is not part of this work.

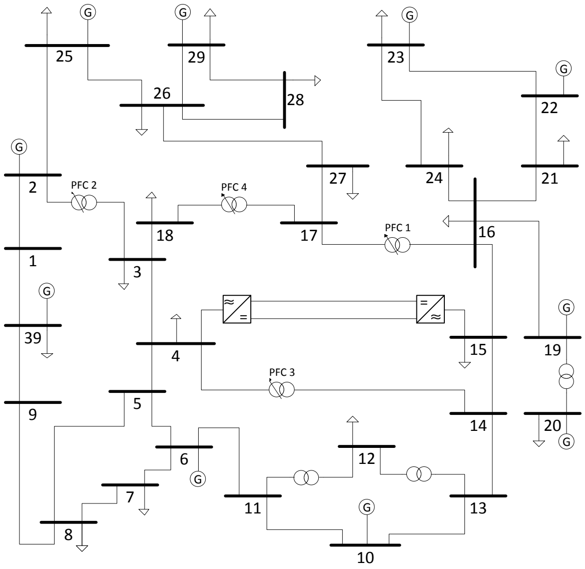
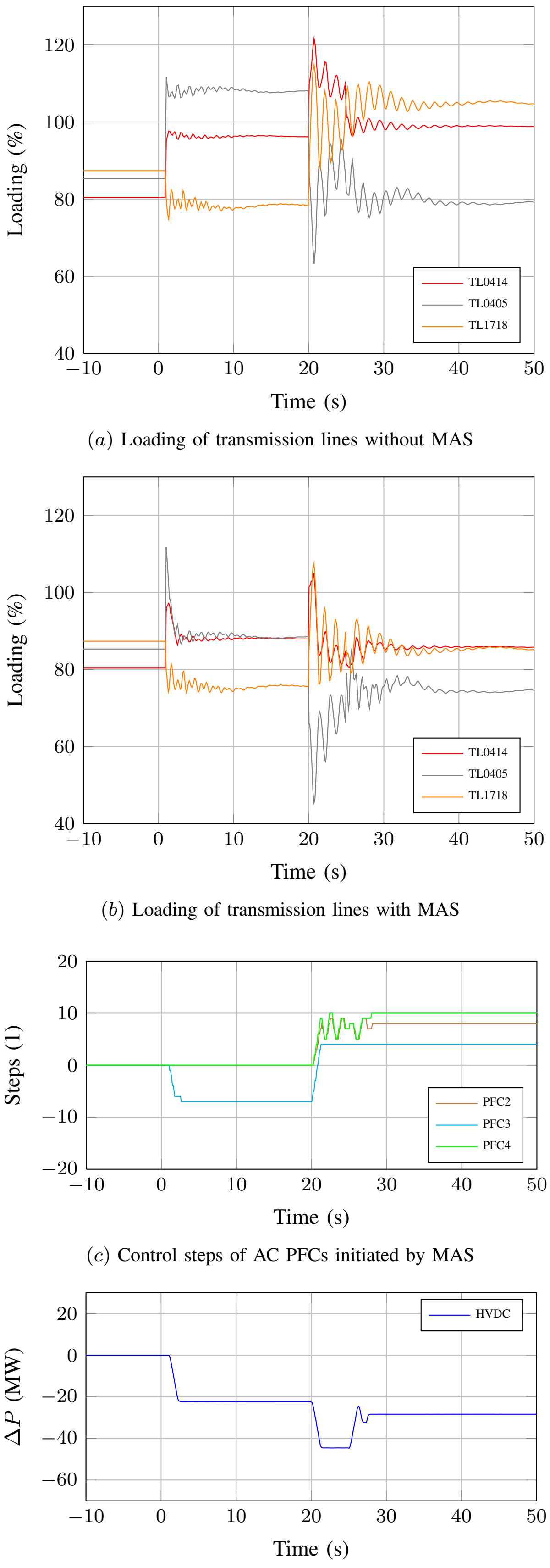


Fig. 6: New England Test System extended by four AC PFCs and one HVDC link

The results discussed above illustrate the case when overloads are not met with fast countermeasures which could lead to cascading outages due to line sagging or triggering of protection devices. With the MAS being activated, the loadings shown in Fig. 7-(b) result due to the control actions initiated by the MAS. Both the active agents for AC PFCs (Fig. 7-(c)) as well as for the HVDC link (Fig. 7-(d)) quickly adapt their control set-point in a stepwise manner. The coordinated action of the fast controlling devices successfully reduces the loading of the lines within seconds below a critical level, thereby



(*a*) Loading of transmission lines without MAS

(*b*) Loading of transmission lines with MAS

(*c*) Control steps of AC PFCs initiated by MAS

(*d*) HVDC adaptation of active power flow initiated by MAS

Fig. 7: Simulation results for coordinated control of AC PFCs and HVDC

possibly avoiding cascading failures.

## IV. Discussion and Comments

In the previous sections we have presented the recent developments towards a robust and adaptive control system that enables coordinated real-time response of AC PFCs (FACTS, PSTs) as well as HVDC links within areas of mutual impact. The implementation of the system could enhance the utilization of the existing transmission network and contribute to power system stability by relieving overloads quickly even in N-1 cases or higher before cascading outage of several branches due to overloading. In the following, we discuss and comment further aspects and potential developments of the distributed power flow control approach.

### A. *FACTS*

FACTS devices are network devices based on power electronics that are designed to support power system operation and stability by fast and precise control (for details see [14]). In contrast to conventional PSTs with a time needed for tap change around $6s$, power flow controlling FACTS devices can change its set-point below $50ms$ and feature a large number (e.g., larger than 100) of discrete control set-points. FACTS devices can easily be integrated in the coordination system proposed above under the assumption that a constant increment (or decrement, respectively) is used for the FACTS devices in each decision making step, analogously to the change of a PST tap setting by one tap position (as presented in the scenario above). However, it would also be of interest whether an intelligent adaptation of the increments (value of the increments and timely variation) could further improve performance, e.g., even faster relief of the overloading. Inspired by [15], an interesting way to pursue could be the adaptation of decentralized access logics in communication networks, such as in TCP (additive increase, multiplicative decrease (AIMD)), Wireless Local Area Networks (IEEE802.11e, value-dependent probabilistic method), or in Ethernet (exponential backoff). The idea of AIMD, e.g., is to increase a value in small, additive steps, and in case of a bad event the value is decreased by a factor smaller than 1. Additional research is required to investigate different alternatives and select an appropriate logic for the timely variation of increments.

### B. *HVDC*

In the past, HVDC links have primarily played a role in Europe for the connection of two asynchronous systems (e.g., connection of off-shore windparks, DC connections to other interconnected systems such as continental Europe to Great Britain and Scandinavia). From a power flow control perspective on the continental European transmission system, these HVDC links can be seen as a power feed-in (or consumption) at a single node in the system and control of these links could be seen as a redispatch measure (compare subsection IV-C). Recently, also development of HVDC transmission lines as point-to-point connections between two nodes in the same synchronous AC transmission system has begun (e.g., between

Spain and France as well as inside Germany). These HVDC links constitute an inherently interdependent power consumption at one node and feed-in at another node only depending on the intended control set-point and not on the surrounding AC network. Therefore, HVDC links constitute a significant potential to change the power flows in the system. However, in contrast to FACTS devices which are primarily dedicated to support power system stability, HVDC links are built primarily with the intention to increase transmission capacity. For this reason, it is a question of operational policies how much flexibility on an HVDC link can be used for power flow control because this flexibility depends on the current utilization of the line (if the link is fully loaded, we cannot increase the power flow on the HVDC link in order to relieve overloading of adjacent AC lines). Integrating HVDC links in the coordination system is essential for implementation of the MAS in the European transmission system as, e.g., the link between Spain and France is situated in an area where also PSTs are installed, thus coordination of power flow control is needed.

Building on power electronics technology, HVDC also enables fast and precise control with a variety of set-points, therefore the question of selecting increments or decrements is similar as in the case of FACTS.

### *C. Redispatch*

As mentioned above, topological actions are not always sufficient to mitigate overloads and a reconfiguration of generation and loads between nodes of the transmission system level (redispatch) constitutes a powerful though more costly measure to control power flows. Redispatch in future smart grids could possibly be enabled by a change of generation or load of a variety of underlying entities (power plants, RES, storages, electric vehicles, industrial loads, consumer loads). To account for this complexity, the redispatch possibilities underlying a node of the transmission network could be aggregated as a certain 'redispatch flexibility' for each node providing a possible power gradient that can be associated with a cost curve and a time frame for reliable execution.

The integration of redispatch in the proposed coordination system would be a powerful extension but poses new challenges, both from a technical and an economic point of view. First, agents in the proposed MAS are associated with branch elements while redispatch occurs at nodes. Therefore, a new concept of nodal agents would have to be developed including an investigation which data would be needed and which consequences for topology analysis and coordinated decision making would arise. Furthermore, redispatch introduces a new dimension of complexity as the balancing of active power in the reconfiguration has to be coordinated and the regulatory framework for power system operation becomes a vital part in the conceptualization. While control of PSTs, HVDC and FACTS belong to the set of topological actions of the TSOs (which have to be exploited primarily as discussed in section I) and have no consequences beyond transmission system operation, redispatch directly impacts market participants. In particular, redispatch has to comply with regulatory guidelines for economic efficiency, e.g., it has to be based on costs, a market, or sensitivities. Dedicated research is necessary to design a coordination that accounts for these new technical and economical aspects of redispatch. However, the inclusion of redispatch in the coordination system offers considerable potential for enabling coordinated real-time congestion-management because redispatch exhibits significant impact on power flows in and it could also be applied in areas where no PFCs are installed, e.g., to counteract internal congestion.

### *D. General Aspects*

With respect to the communication among agents, it is of importance to keep the traffic on the communications network as low as possible. Intelligent routing or fine tuning of the criteria for discarding messages could contribute to meet this challenge. Furthermore, the impact of incomplete (e.g., loss of packages, loss of communication links, or defect of agents) and asynchronous data on the performance of the coordination system should be investigated in detail and concepts for robustly dealing with these effects should be integrated. For this, it is planned to integrate the coordination system in the co-simulator for power systems and communications networks developed in FOR1511 which promises to contribute significantly to enhance and validate the system (for details see [16], [17]). Also, it is worthwhile to investigate how the precautious waiting times of the agents due to communication delays could be decreased.

Besides the adaptive contribution to system stability, the coordination system could also provide economic benefits by enabling to account for corrective actions in the operational planning and thus increase available capacity for transmission corridors. This benefit can only be exploited when the performance of the coordination system is predictable and the response of the automatic control is close to the results of an security constrained OPF. It has to be investigated how the MAS performs compared to OPF results and if parameters can be tuned to provide a reasonably predictable system performance.

Last, it could be of interest to investigate the assumptions and fine-tune system performance. The DC load flow assumption is generally plausible for the approximation of sensitivities and offers the important benefit of having no risk of non-convergence (in contrast to AC load flow). If the distributed topology analysis is extended to full load flow estimations, e.g., for estimating adequate increments, the validity of DC load flow should be counterchecked.

## V. Conclusion and Outlook

In this paper, we summarized the progress on developing a distributed coordination system for automated power flow control in critical network situations. The coordination system is based on a Multi-Agent System and distributes information by sending messages containing current measurements and states of devices. Based on these messages sent and forwarded between network nodes, a distributed topology and sensitivity analysis can be executed by the agents corresponding to substations where power flow controllers are situated. Coordinated action of power flow controllers with areas of mutual impact is achieved by weighting functions based on loading and sensitivities. Simulation results for a test case have been presented to examine the performance of the coordination systems, nonetheless, several aspects and enhancements, e.g., extension to redispatch, have been pointed out in the discussion that require further investigation.

The distributed approach proposed above promises to enable advantages regarding real-time capability, adaptiveness and robustness compared to current centralized practice. This also rises fundamental questions of operational philosophy: what extent of automation of decision making is acceptable from the system operator's point of view and under which conditions are distributed solutions an admissible alternative for traditional centralized control schemes? Our approach is meant to be a complement to conventional practice: in normal operation, the MAS does not take action at all, it only intervenes in case of a critical network situation identified by high loading of lines. In the latter case (e.g., in an N-2 situation), the agents take immediate action and mitigate overloads to avoid cascading outages.

As future works, the MAS will be tested in the co-simulation framework for power and communication systems (compare subsection IV-D) in order to examine the real-time performance of the system with a closer look at the communication network and the interdependence with other applications of power system operation (e.g., system protection). Further, the integration of redispatch will be subject of future work as this would enable a comprehensive real-time congestion management for critical network situations.

## References

[1] O. Arrivé, F. Boulet, C. Moran-Pena, P. Johnson, J. Van Roost, N. Schuster, O. Ziemann, H-D. Ziesemann, M. Heinz, M. Pierchalla, and D. Klaar, "Improved TSO coordination in the Central West European region: CIGRE C2-204," in *CIGRE Session 2012, Paris*, CIGRE 2012, Ed., 2012.

[2] C. Rehtanz, "Dynamic power flow controllers for transmission corridors," in *2007 iREP Symposium - Bulk Power System Dynamics and Control - VII. Revitalizing Operational Reliability*. IEEE, 2007, pp. 1–9.

[3] S. H. Horowitz and A. G. Phadke, "Third zone revisited," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 23 – 29, 2006.

[4] F. Capitanescu, J. Martinez Ramos, P. Panciatici, D. Kirschen, A. Marano Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-art, challenges, and future trends in security constrained optimal power flow," *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, 2011.

[5] A. Marinakis, M. Glavic, and T. van Cutsem, "Minimal reduction of unscheduled flows for security restoration: Application to phase shifter control," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 506–515, 2010.

[6] J. Verboomen, "Optimisation of transmission systems by use of phase shifting transformers," Ph.D. dissertation, Technische Universiteit Delft, Delft, 2008.

[7] G. Hug-Glanzmann and G. Andersson, "Decentralized optimal power flow control for overlapping areas in power systems," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 327–336, 2009.

[8] U. Hager, S. Lehnhoff, C. Rehtanz, and H. F. Wedde, "Multi-Agent System for Coordinated Control of FACTS Devices," in *15th International Conference on Intelligent System Applications to Power Systems (ISAP '09)*, 2009, pp. 1–6.

[9] U. Hager, S. Lehnhoff, and C. Rehtanz, "Analysis of the robustness of a distributed coordination system for power flow controllers," in *17th international Power Systems Computation Conference (PSCC)*. IEEE, 2011.

[10] S. Lehnhoff, U. Hager, T. Zimmermann, and C. Rehtanz, "Autonomous distributed coordination of fast power flow controllers in transmission networks," in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*. IEEE, 2011, pp. 1–8.

[11] S. C. Müller, A. Kubis, S. Brato, U. Häger, C. Rehtanz, and J. Götze, "New applications for wide-area monitoring, protection and control," in *IEEE PES ISGT Europe 2012*. IEEE, 2012.

[12] S. C. Müller, U. Häger, H. Georg, S. Lehnhoff, C. Rehtanz, C. Wietfeld, H. F. Wedde, and T. Zimmermann, "Einbindung von intelligenten entscheidungsverfahren in die dynamische simulation von elektrischen energiesystemen," *Informatik-Spektrum*, vol. 36, no. 1, pp. 6–16, 2013.

[13] R. Diffenderfer, *Electronic devices: systems and applications*. Thomson/Delmar Learning, 2004.

[14] X.-P. Zhang, C. Rehtanz, and B. Pal, *Flexible AC transmission systems: Modelling and control*. Berlin: Springer, 2006.

[15] A. Hartmanns, H. Hermanns, and P. Berrang, "A comparative analysis of decentralized power grid stabilization strategies," in *The 2012 Winter Simulation Conference*, 2012.

[16] S. C. Müller, H. Georg, C. Rehtanz, and C. Wietfeld, "Hybrid simulation of power systems and ict for real-time applications," in *accepted for publication at the IEEE PES ISGT Europe 2012*. IEEE, 2012.

[17] H. Georg, S. C. Müller, C. Rehtanz, and C. Wietfeld, "A HLA Based Simulator Architecture for Co-simulating ICT Based Power System Control and Protection Systems," in *3rd IEEE International Conference on Smart Grid Communications (SmartGridComm 2012)*. IEEE, 2012.

**Sven Christian Müller** received the Diploma degree in Industrial Engineering and Management at TU Dortmund, Germany, and a Master of Science in Industrial Engineering at the Georgia Institute of Technology, USA. Since 2011, he is working on his Ph.D. degree at the Institute of Energy Systems, Energy Efficiency and Energy Economics (ie$^3$) at TU Dortmund. His research is focused on modelling and simulation of power systems and electricity markets, power system operation and network integration of renewable energies.

**Ulf Häger** received his Ph.D. in electrical engineering from the TU Dortmund University, Germany, in 2012. Currently he works as the head of the department 'Electrical Transmission and Distribution Networks' at (ie$^3$). His fields of interest are wide area power flow control and wide area protection systems as well as the application and development of FACTS devices for power flow control. Furthermore he is involved in national and international network expansion studies.

**Christian Rehtanz** (M'96–SM'05) received his diploma degree in electrical engineering in 1994 and his Ph.D. in 1997 from the TU Dortmund University, Germany. From 2000 he was with ABB Corporate Research, Switzerland and from 2003 Head of Technology for the global ABB business area Power Systems. From 2005 he was Director of ABB Corporate Research in China. From 2007 he is professor and head of the Institute of Energy Systems, Energy Efficiency and Energy Economics (ie$^3$) at the TU Dortmund University. His research activities include technologies for network enhancement and congestion relief like stability assessment, wide-area monitoring, protection, and coordinated FACTS- and HVDC-control.

# CALL FOR PAPERS

## IEEE Workshop

# Modeling and Simulation of cyber-physical energy systems

May 20 2013, Berkeley CA

Workshop chairs:
Peter Palensky (Austrian Institute of Technology), Edward A. Lee (University of California, Berkeley)

Modern energy systems combine information technology, electrical and thermal infrastructure, autonomous roles and interact with other systems like markets and regulations. Existing modeling and simulation tools are not capable to cover such systems in all of their aspects, new languages, methods and tools are necessary. A combination of universal modeling languages like Modelica and established, specialized tools like grid simulators and telecommunication simulators is necessary. This leads to modeling and co-simulating hybrid systems where for instance a multi-agent framework and an electric grid simulator are combined to investigate smart electric vehicle charging algorithms. It is especially the potential size of such systems that constitute a challenge for modeling and simulation. Implementing these future CPS are another substantial challenge. The designed algorithms need to be compact, computationally inexpensive, potentially self-organizing and intrinsically stable if applied to real energy systems.

This workshop is a platform for researchers and developers to exchange ideas to the following (not exhaustive) list of topics:

- Hybrid modeling and simulation
- Co-Simulation
- High-performance computing
- Analytics of system data
- Ontologies for energy systems
- Applications of cyber-physical energy systems
- Distributed algorithms and control
- Standards in interfacing components
- Numerics for hybrid and co-simulation
- Formal languages for energy systems
- Smart Grid modeling
- Demand response and power quality
- Information and communication technology for intelligent energy systems

**Program Committee**

- Christoph Grimm (Uni Kaiserslautern, Germany)
- Michael Wetter (LBNL, USA)
- Sven Christian Mueller (TU Dortmund, Germany)
- Hiroaki Nishi (Keio University, Japan)
- Seung Ho Hong (Hanyang University, Korea)
- Wolfgang Kastner (TU Vienna, Austria)
- Wilfried Elmenreich (Uni Klagenfurt, Austria)
- Pierluigi Siano (Uni Salerno, Italy)
- Jeannie Falcon (National Instruments, USA)
- Kishor S. Trivedi (Duke University, USA)
- Jenny Yan Liu (Concordia University Montreal, Canada)

**Important Dates**

Full Paper submission: Jan 31, 2013
Notification of acceptance: Feb 22,2013
camera-ready papers: March 22,2013

Full paper submissions are peer-reviewed by at least 3 reviewers. Workshop language is English.
Technical co-sponsorship by the IEEE Industrial Electronics Society
Details: www.palensky.org/mscpes

Sponsor: AUSTRIAN INSTITUTE OF TECHNOLOGY

Venue: Berkeley UNIVERSITY OF CALIFORNIA