BERKELEY LAB

LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

VirGIL

A Demand Response Platform for Smart Grids

Spyros Chatzivasileiadis, Sila Kiliccote, Michael Wetter, David Lorenzetti, Marco Bonvini, Rongxin Yin, Javier Matanza, Zhenhua Liu, Emre Kara, Emma Stewart, Thierry Nouidui, Rajiv Parmar

Noon-Talk at LBNL
March 12, 2015

Outline

- VirGIL Co-simulation Framework
- Master Algorithm – Quantized State System
- Building Modeling
- Communications
- Power System
- Use Cases
- Conclusions and Future Steps
- VirGIL Demo

BERKELEY LAB

VirGIL Scope

Develop a *modular* co-simulation platform integrating:

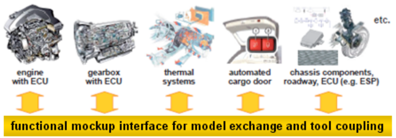
- Power systems
- Communication networks
- Building models
- Control and Optimization

Objectives:

- Capture the interactions between building dynamics and power systems for real-time grid operation and planning
- Build an efficient co-simulation platform
- Integrate legacy tools to decrease the barriers for adoption to the industry

Motivation

- Most demand response tools focus either on the grid (neglecting building dynamics), or on the buildings (neglecting their impact on power flows)
- Distribution system operators need tools that they can *seamlessly integrate* with their simulation platforms to control DER and deploy Demand Response
- Modular architecture based on open-standard → integrate future tools; e.g. EV simulation, convex optimization, and PV inverter control, Hardware in the Loop, etc.
- Reduce barriers for the commercialization of LBNL tools by offering full functionality with widely accepted software



VirGIL Co-Simulation Framework

Master Algorithm

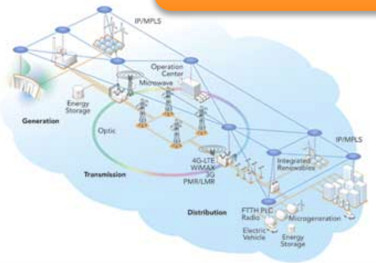
FMI

FMI

FMI

Power Systems

Communications



Buildings

FMI: Functional Mockup Interface

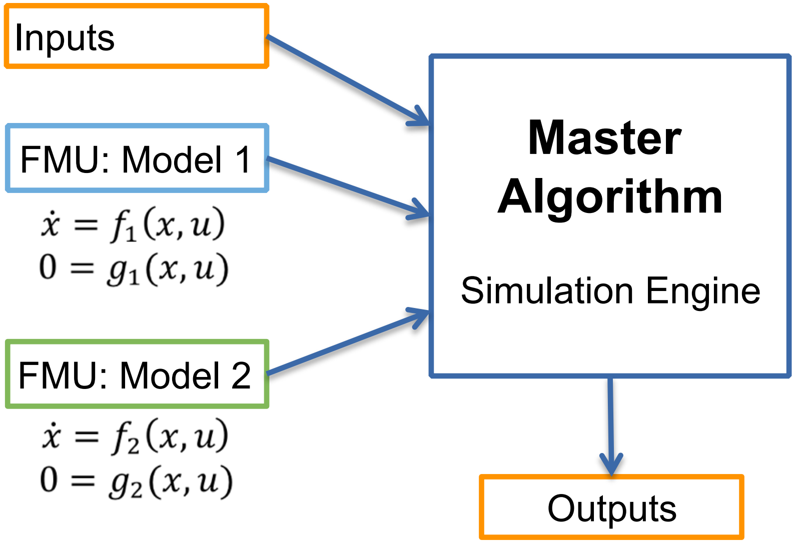
BERKELEY LAB

What is FMI?

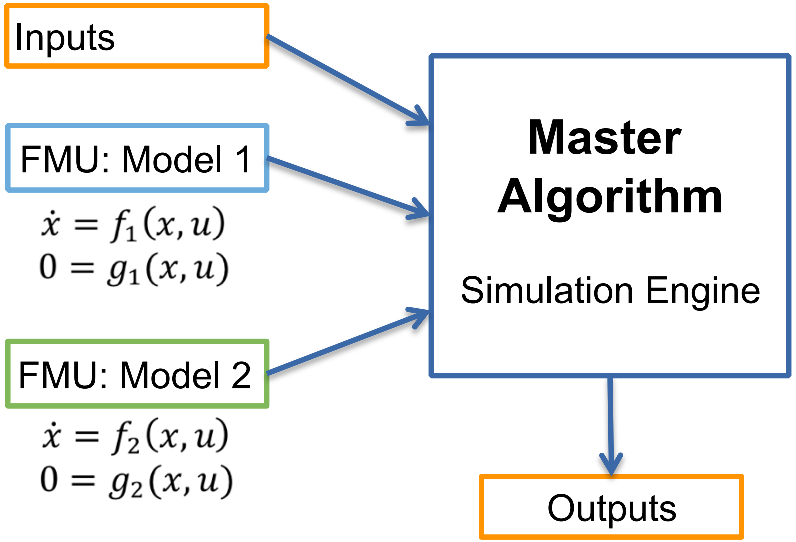
- **F**unctional **M**ockup **I**nterface
- Standardized interface to couple different models and simulation engines in a common simulation platform
- FMI 1.0 released in 2010; FMI 2.0 released 2014; LBNL active in the development of FMI 2.1 specs
- FMU: Functional Mockup Unit
  - Every model or simulation engine is exported as FMU

FMI for Model Exchange

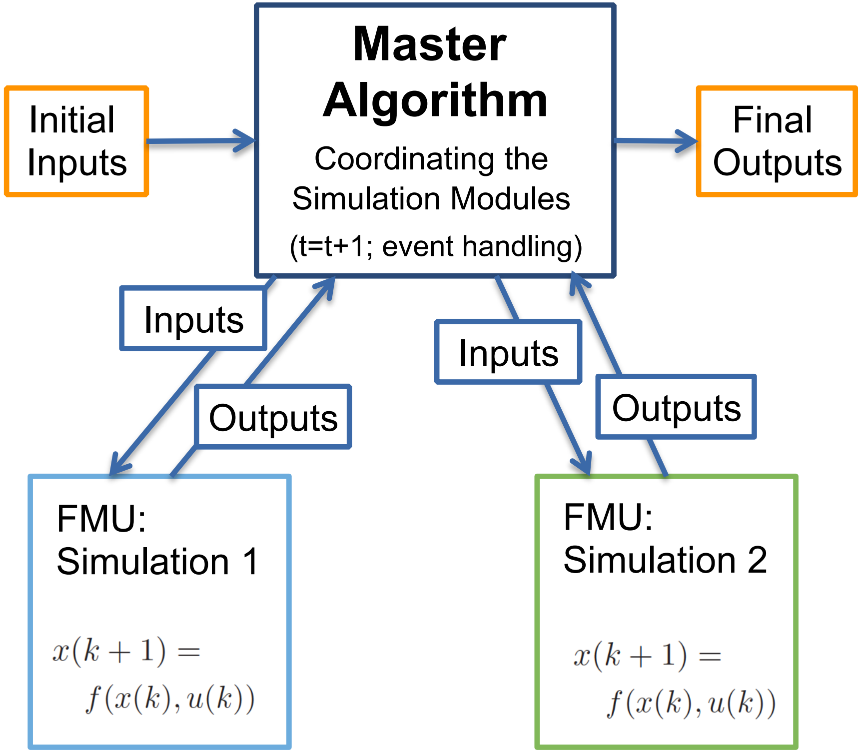
FMI for Co-Simulation



FMI for Model Exchange



FMI for Co-Simulation



Use Cases

Milli-secs → Seconds → Minutes → Hours → Days

Short-term (ms to s)

Transient phenomena
Comm. Role: Critical

Main components: Inverter-connected elements with storage (i.e. storage devices), Evs, PVs, small wind turbines

1. Symmetrical injection of 1-phase inverters
2. Virtual Inertia
3. Provision of reactive power

- Autonomous operation of the agents
- Periodically receive a setpoint or a band within which they should move

Medium-term (min to h)

Regulating/Balancing
Congestion mgmt.
Voltage problems

Main components: Storage elements, both inverter-connected and not, such as buildings

1. Overloadings
2. Overvoltages
3. Voltage drops

- Simulation and optimal control
- Communicate the optimal setpoint to the devices

(Model predictive control could also be used for a short time horizon)

Long-term (h to days to years)

Planning

Day-/Week-ahead:
Provide incentives for specific DR units to enter the market

Years:

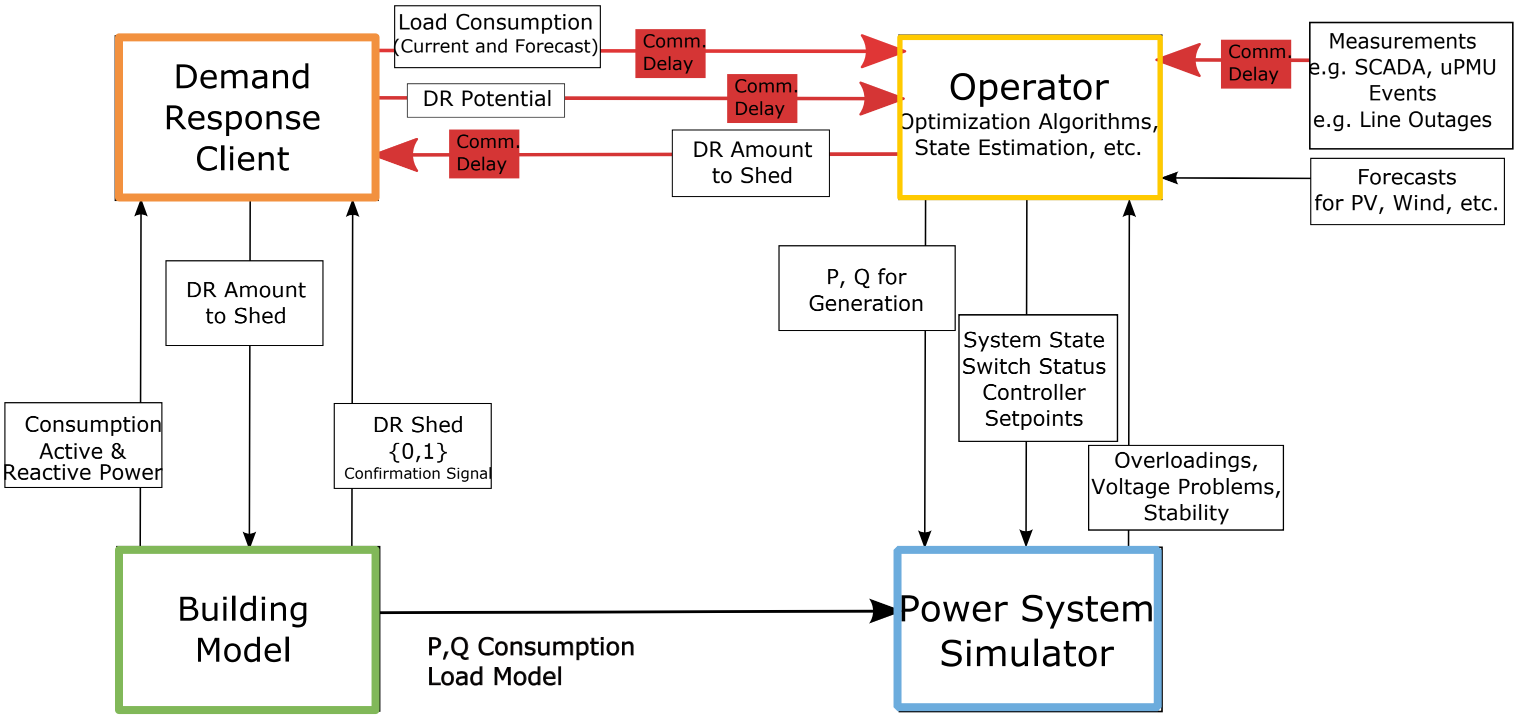
1. Deferral of new T&D Investments
2. Impacts/Investment on PVs vs DR

Day/Week Ahead:

- Model predictive control to plan the operation of all storage elements, including thermal storage (buildings, water heaters, etc.)

BERKELEY LAB

Implementing DR* for buildings in VirGIL



DR: Demand Response

Software

Master Algorithm
Ptolemy II

Grid Simulation Software	Communications Simulation Software	Buildings Simulation Software
Powerfactory	OMNET++	Modelica
GridLabD Matlab/Simulink Modelica Powerworld CYMDIST PSS/SINCAL PSCAD	ns-3 Modelica ns-2 Matlab/Simulink	EnergyPlus Etc.

BERKELEY LAB

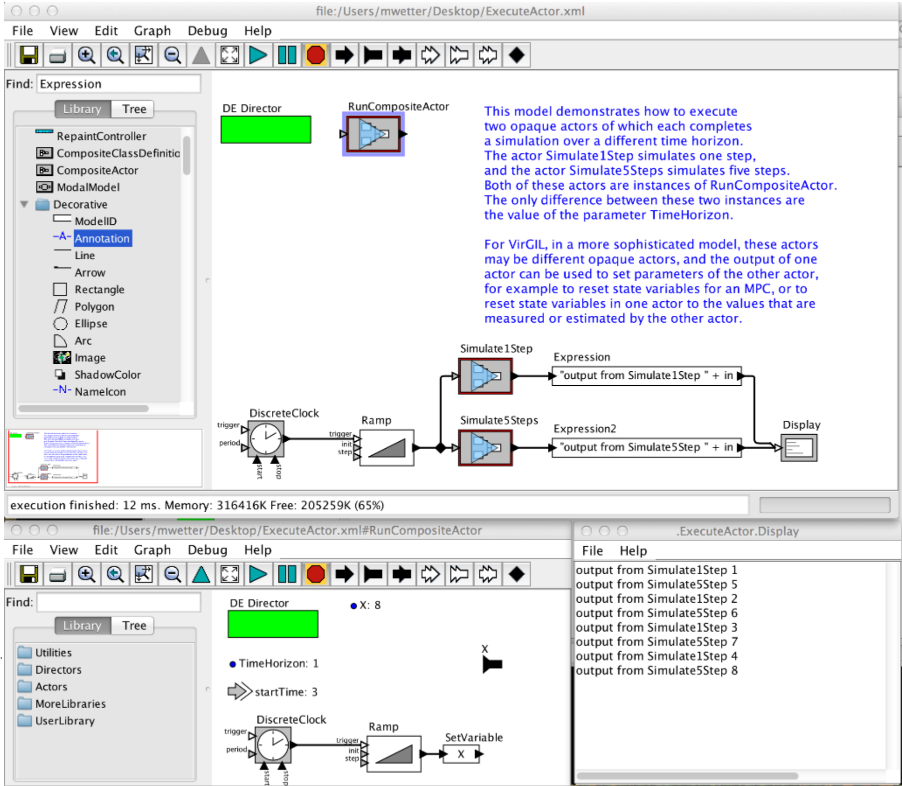
Outline

BERKELEY LAB

Ptolemy II – Coordination of FMUs

- Ptolemy II can handle both FMI for co-simulation and Model Exchange
- Discrete Event Simulator
- It coordinates the FMUs over time
- Ptolemy II integrates QSS* solvers for Model Exchange FMUs (Building)

Example of simulating two opaque Ptolemy II actors over different time horizons using the DE domain



*QSS: Quantized State System

Master algorithm uses Quantized State Systems methods to advance through simulation time

QSS solves a differential equation of the form

$$\dot{x} = f\left(x, u, t\right)$$

QSS rewrites this system as

$$\dot{x} = f\left(q, \mu, t\right)$$

where:

- *q*: *quantized state;* low-order polynomial estimate of the state trajectory
- *μ*: quantized version of inputs u(t)

BERKELEY LAB

QSS reduces the work needed to integrate the system

- QSS decouples slow and fast dynamics

Each FMU shares *q*, rather than *x*, with the rest of the system

- Cuts the cost of communication and computation between FMUs

The system steps forward in time until some *q* has to be refreshed

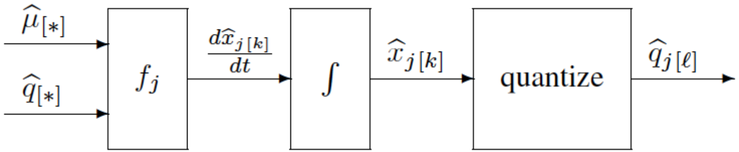
- Allows integration to take the longest step possible

BERKELEY LAB

Quantized State Systems

Two main modules:

1. Integrator of $\dot{x} = f\left(q, \mu, t\right)$, where *q* is just a polynomial
2. Quantizer that, given x(t), calculates q(t)



- If ΔQ larger than a specified tolerance, requantize the state and produce a new polynomial

$$q_i(t) = \begin{cases} x_i(t) & \text{if } |q_i(t^-) - x_i(t)| = \Delta Q_i, \\ q_i(t^-) & \text{otherwise.} \end{cases}$$

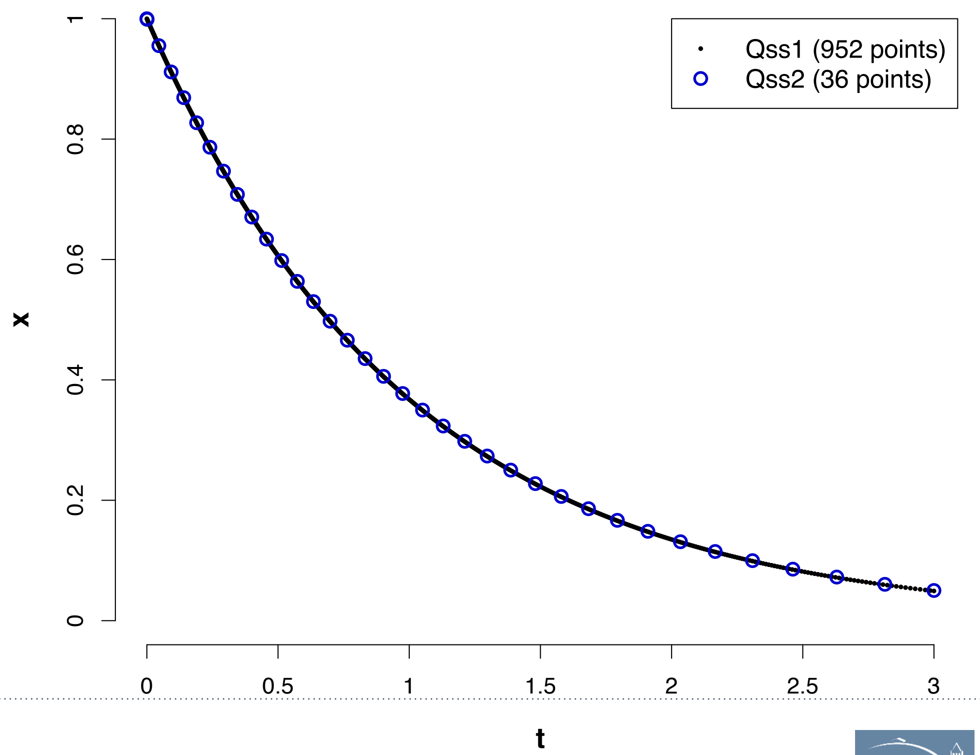
- Broadcast this polynomial to any other component
- QSS integrates each ODE through a series of discrete events

QSS1 vs. QSS2

- Solving the exponential problem:

$$\dot{x} = -x$$
$$x = e^{-t}$$

- QSS1: piecewise constant
  QSS2: piecewise linear
- QSS2 can be much more efficient than QSS1
- Currently using QSS2; planning to move to QSS3

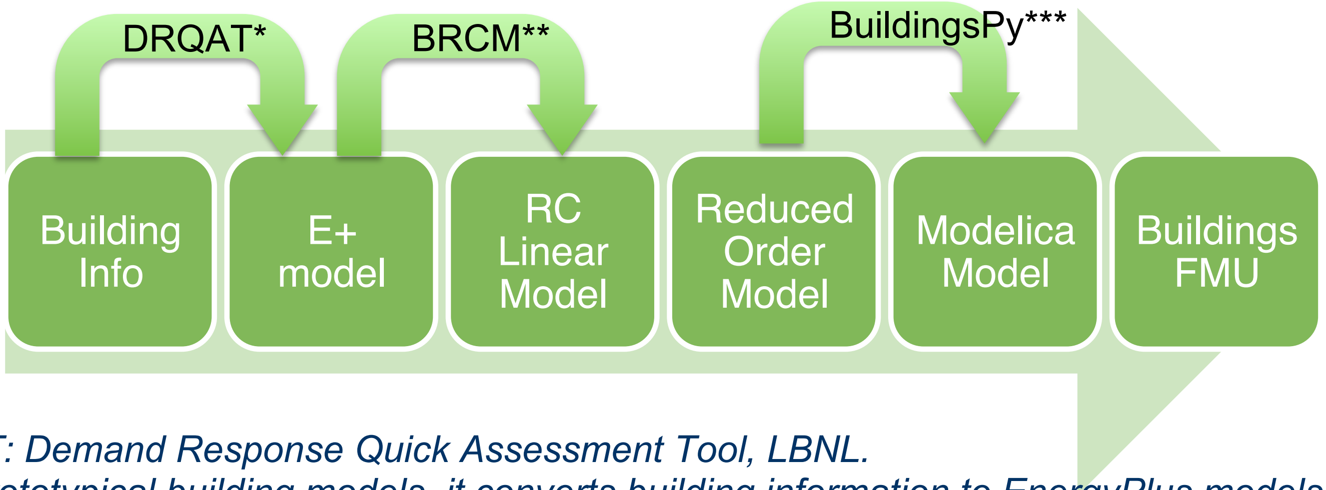


Outline

1. VirGIL Co-simulation Framework
2. Master Algorithm – Quantized State System
3. **Building Modeling**
4. Communications
5. Power System
6. Use Cases
7. Conclusions and Future Steps

Buildings – Model Flow

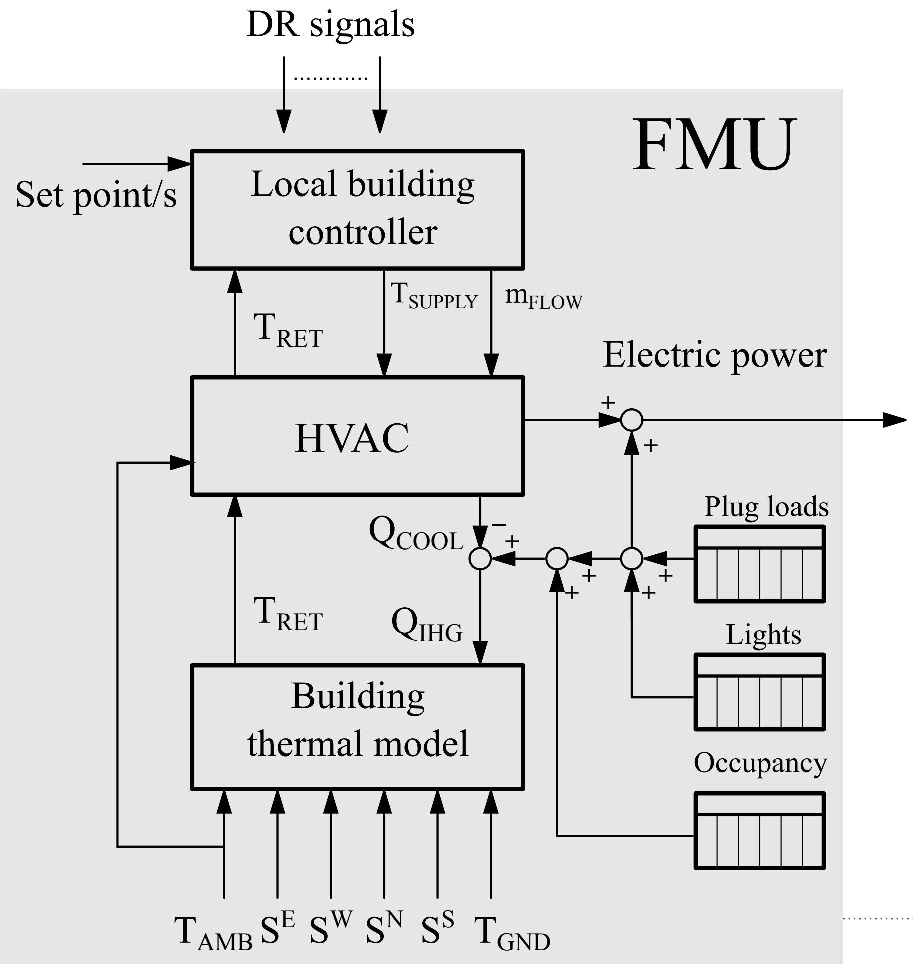
VirGIL needs a building model that captures all the relevant dynamics, but without placing an undue burden on the solver



**DRQAT: Demand Response Quick Assessment Tool, LBNL.*
Using prototypical building models, it converts building information to EnergyPlus models.
***BRCM: Matlab Toolbox developed by ETH Zurich.*
Facilitates the physical modeling of buildings in R-C equivalent models.
****BuildingsPy: LBNL tool to convert to Modelica models*

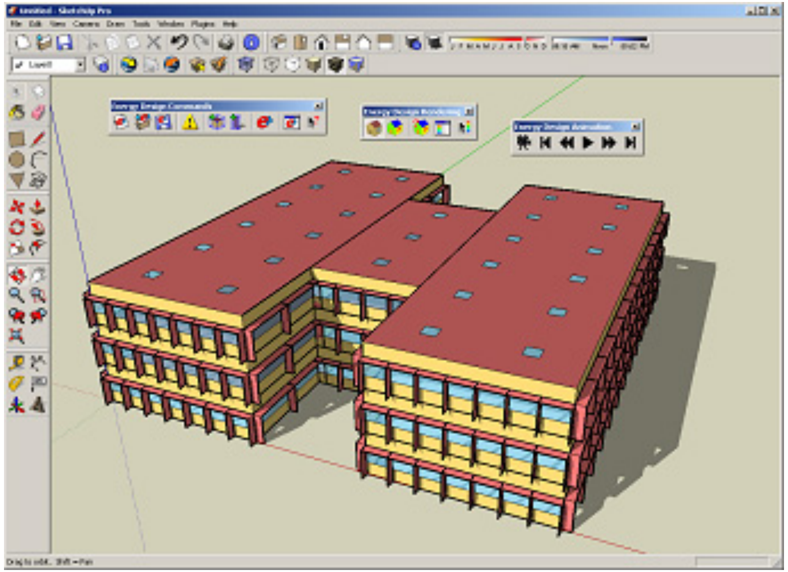
BERKELEY LAB

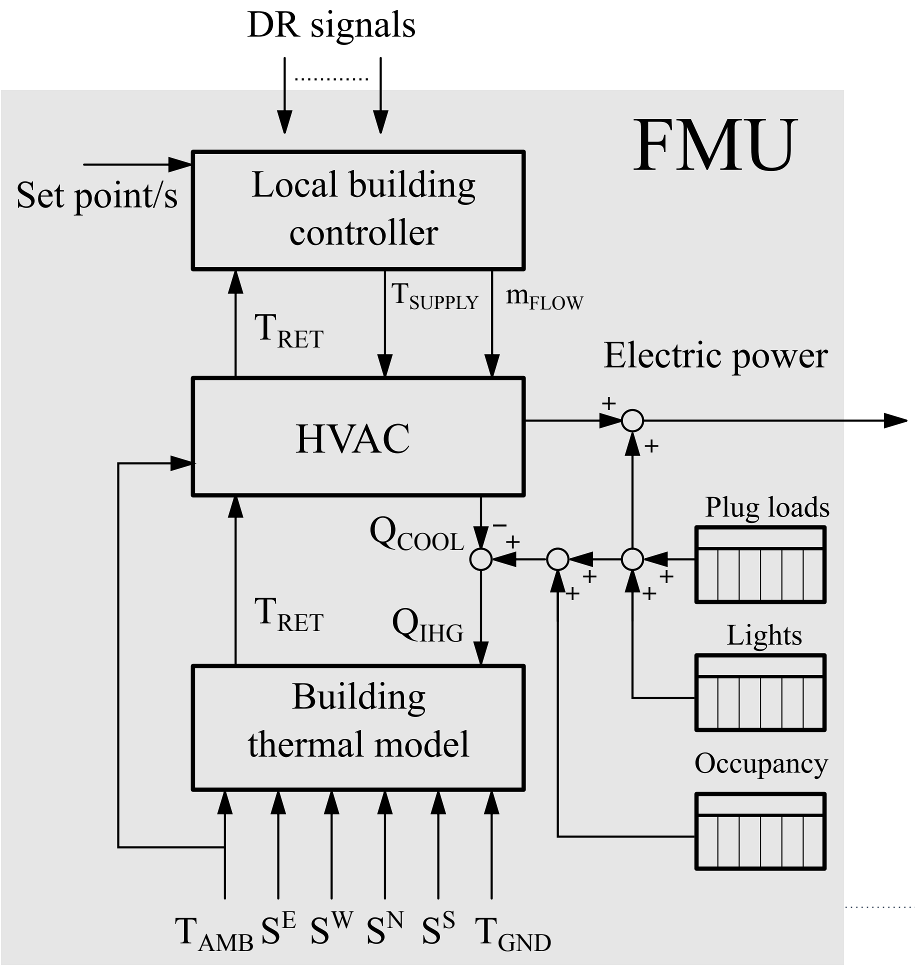
Building FMU



EnergyPlus (E+) Model

- Entire description of building geometry
- Models conductivity of the wall layers, their thermal capacitances, solar heat gain coefficients of the windows, etc.
- This information will be used to generate a simplified first-principles model of the building





Building FMU

Simplifications EnergyPlus→RC Model

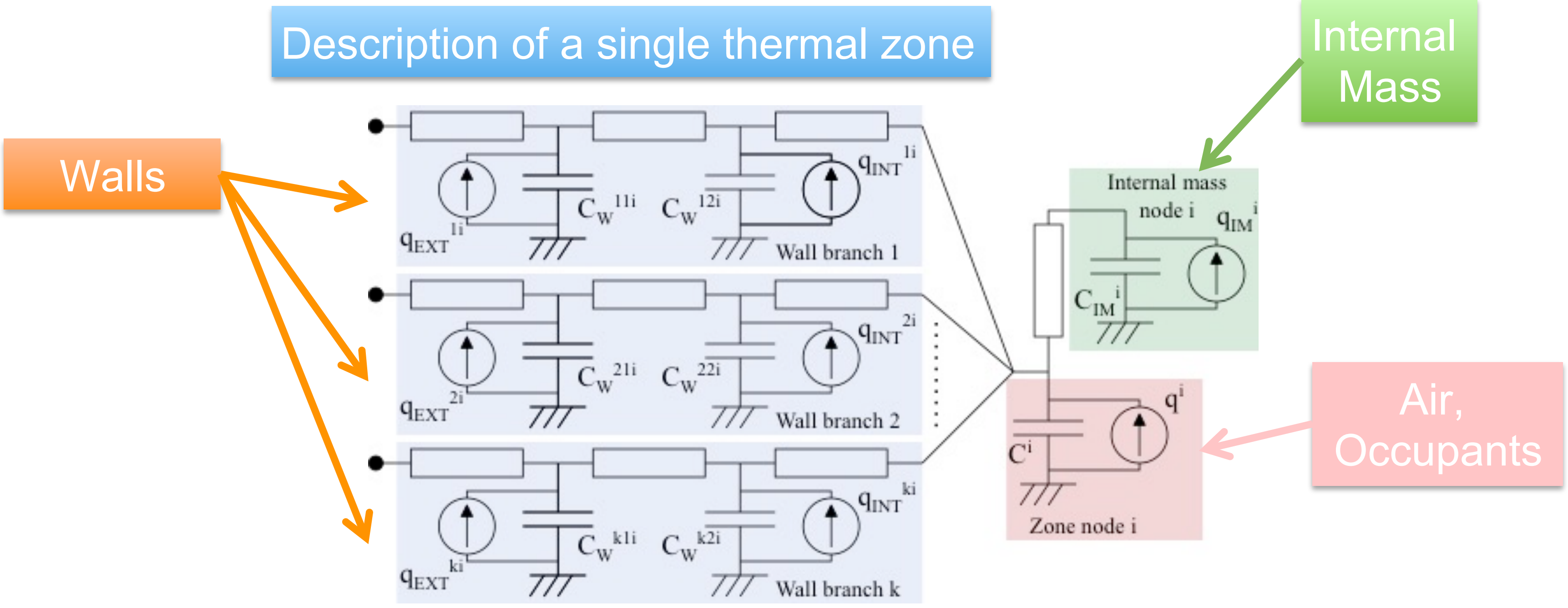
- HVAC and controller not part of the building thermal model (control input modeled as "disturbance")
- Output: Tret; weighted average of the thermal zones temperature

$$\dot{x}(t) = Ax(t) + B_u u(t) + B_v v(t)$$
$$y(t) = Cx(t) + D_u u(t) + D_v v(t)$$

$$\dot{x}(t) = Ax(t) + B_v v(t)$$
$$y(t) = Cx(t)$$
$$C = \begin{pmatrix} \frac{V_1}{V_{TOT}} & \cdots & \frac{V_{nx}}{V_{TOT}} & 0 & \cdots & 0 \end{pmatrix}$$

RC Model

- Models thermal mass and effect of solar radiation

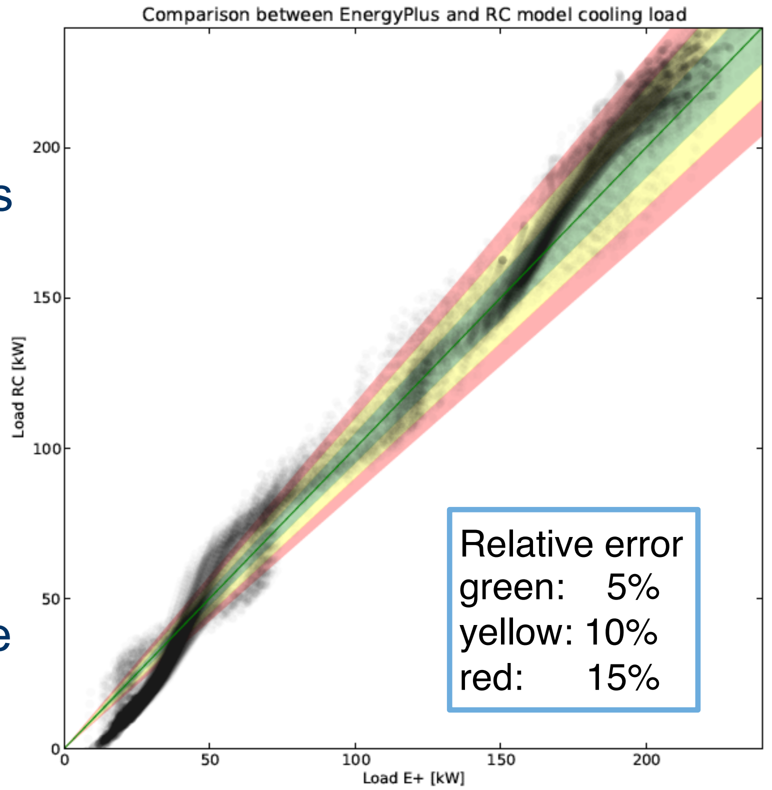


BERKELEY LAB

EnergyPlus vs. RC Model

- LBNL Building 71
- RC model calibration for the summer period → highest probability for DR events
- Sensitivity analysis to rank the key RC parameters to tune
- Data points: 5 min resolution May-Oct.
- Smaller relative error in high consumption
- Larger error during night-time (low consumption); RC model does not capture the nonlinearities of the long-wave radiation

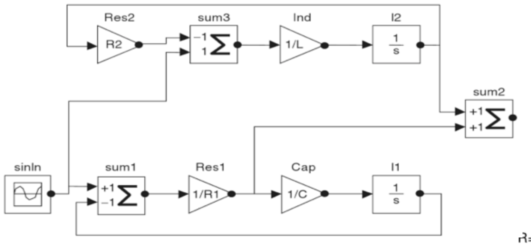
Cooling Load (kW)
EnergyPlus vs. RC



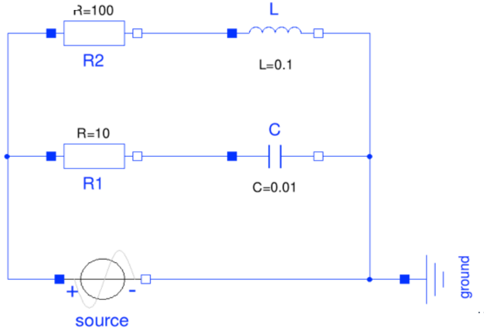
BERKELEY LAB

Building Model

- RC Model Reduction: 108 states → 8 states
  - Less than 1% difference between the full and the reduced RC model

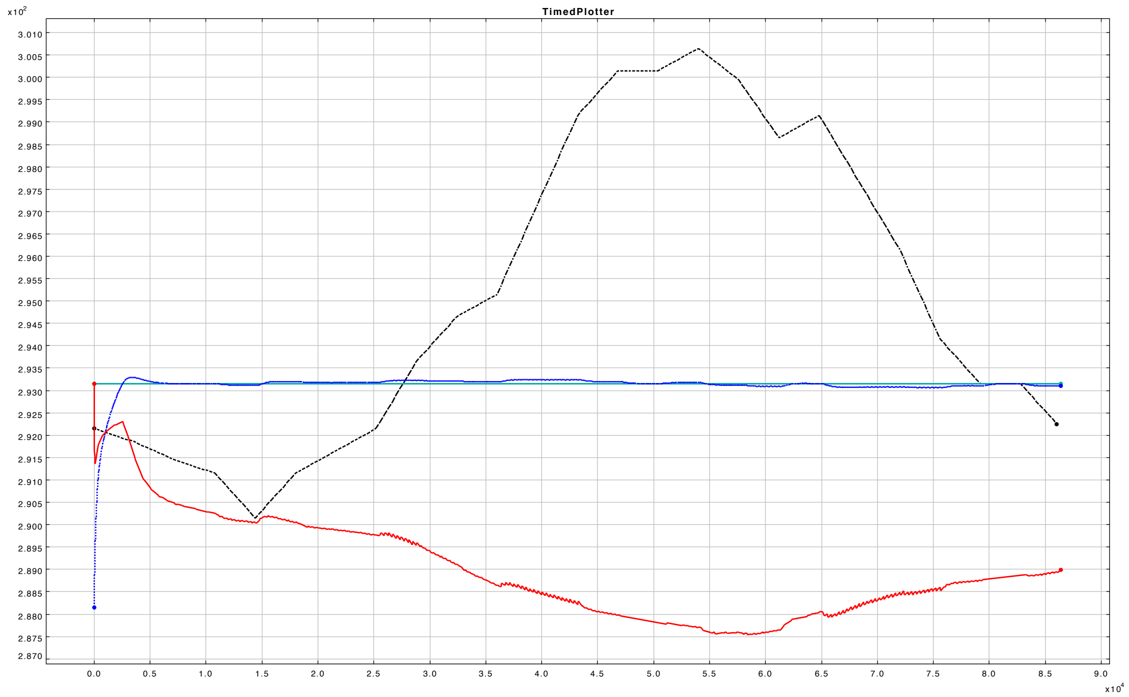


- Model transferred to Modelica
- Modelica is an acausal object-oriented modeling language
- Models can be inverted for specific analysis or control purposes



Building 71

- 1 day (86400 sec)
- Temp. setpoint: 20 C
- Black: ambient temp.
- Blue: building temp
- Red: HVAC setpoint



BERKELEY LAB

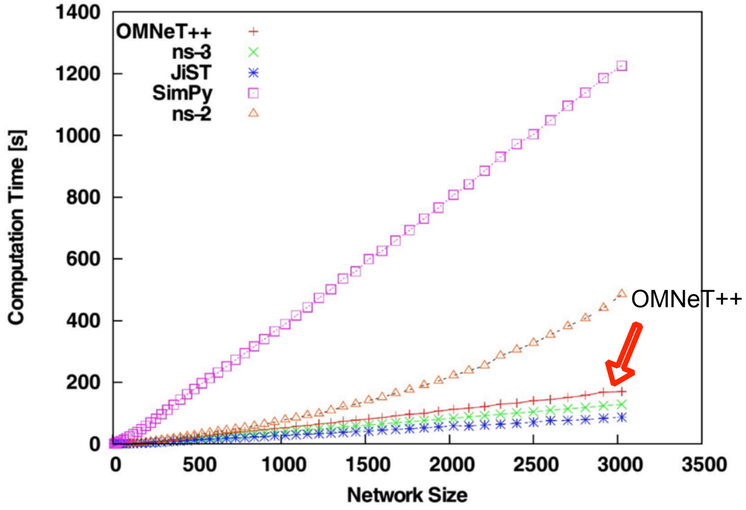
Outline

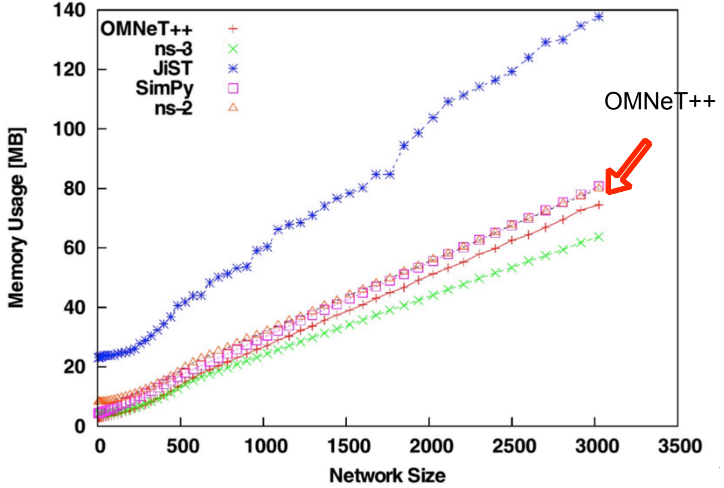
1. VirGIL Co-simulation Framework
2. Master Algorithm – Quantized State System
3. Building Modeling
4. **Communications**
5. **Power System**
6. Use Cases
7. Conclusions and Future Steps

BERKELEY LAB

Why OMNET++ for Communications?

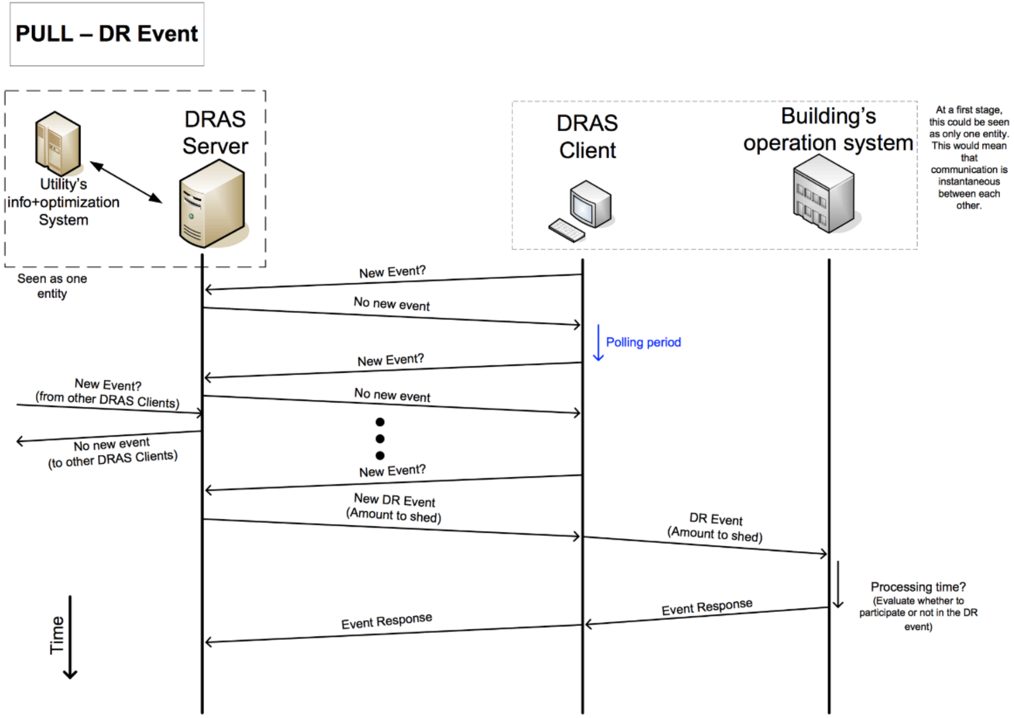
- OMNeT++ has good performance and scales well
- Validated modules for both wired and wireless communication
- Good documentation

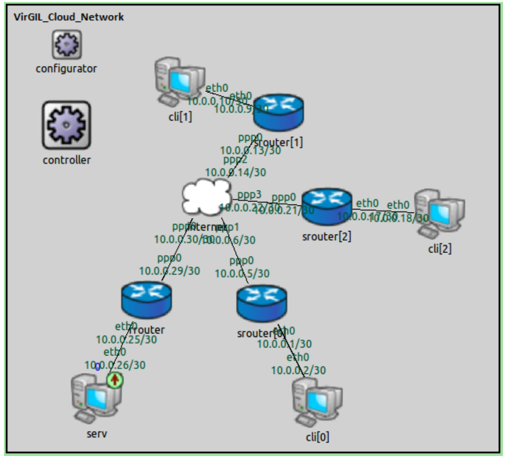


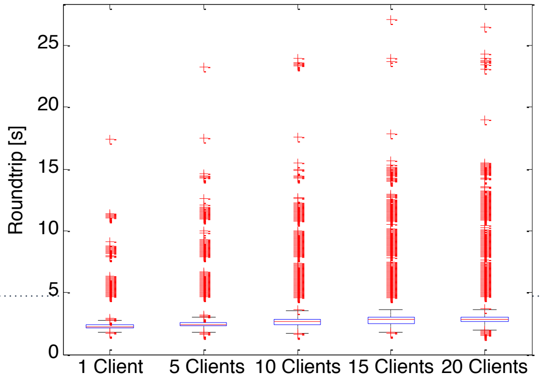


Communications in OMNET++

Modeling the OpenADR Protocol

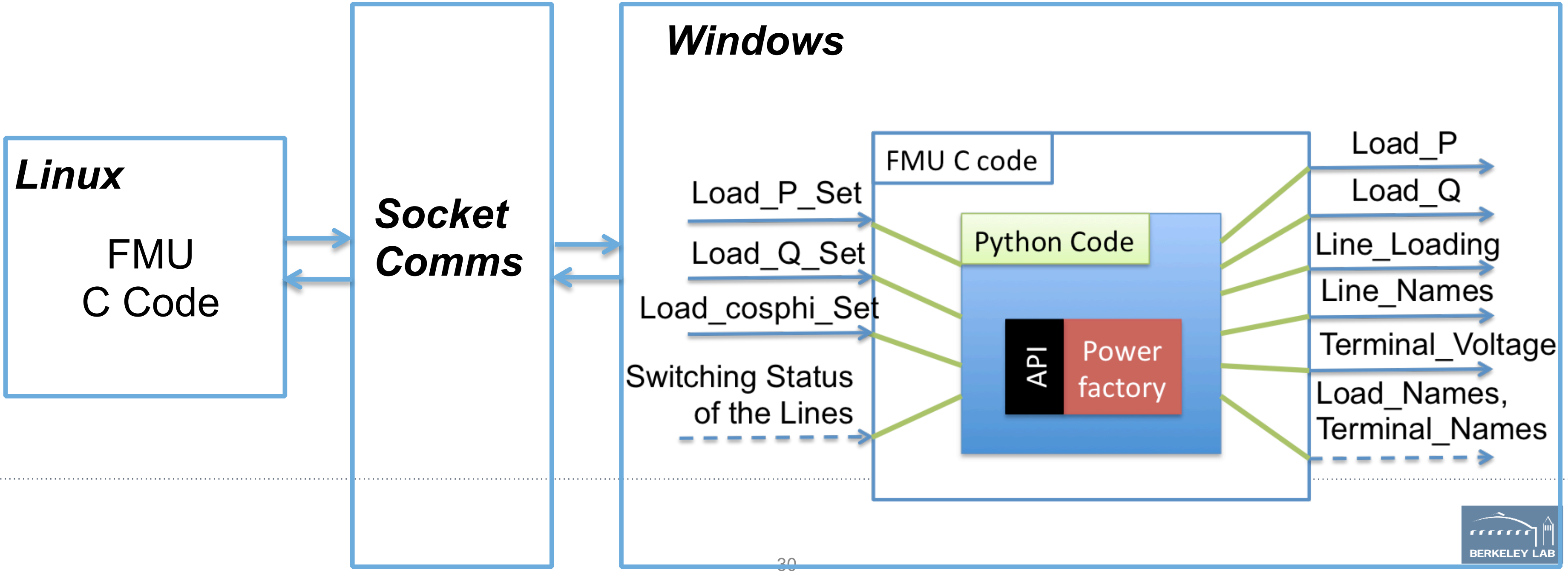






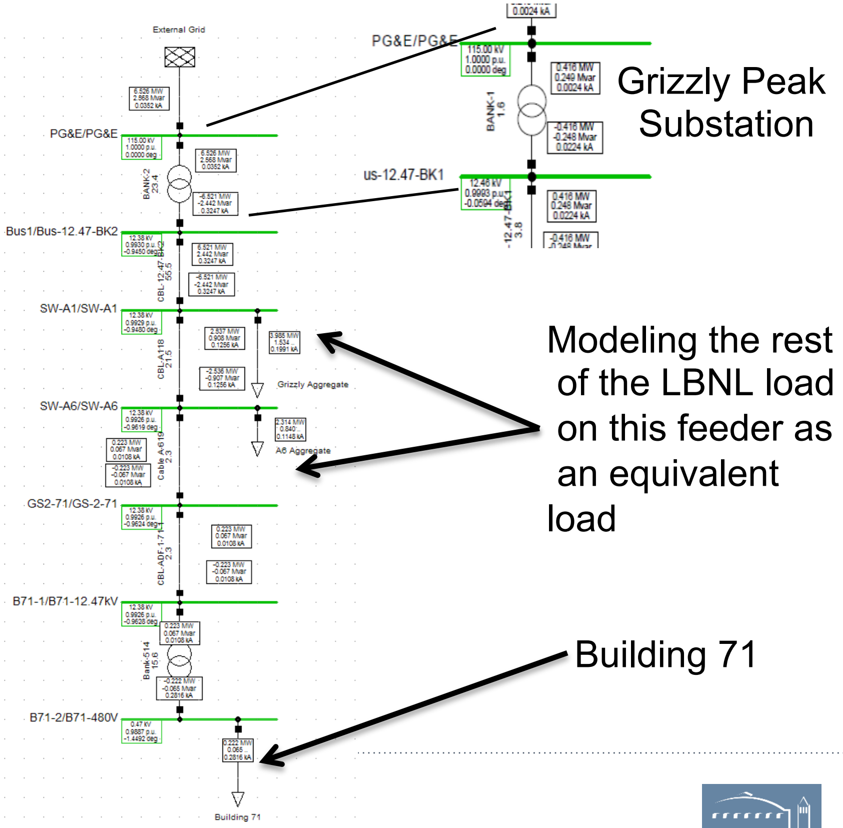
Power FMU

- Powerfactory requires Windows. The rest of our tools operate on Linux.
- Developed a Python interface between Powerfactory API and the FMU
- FMU development in C



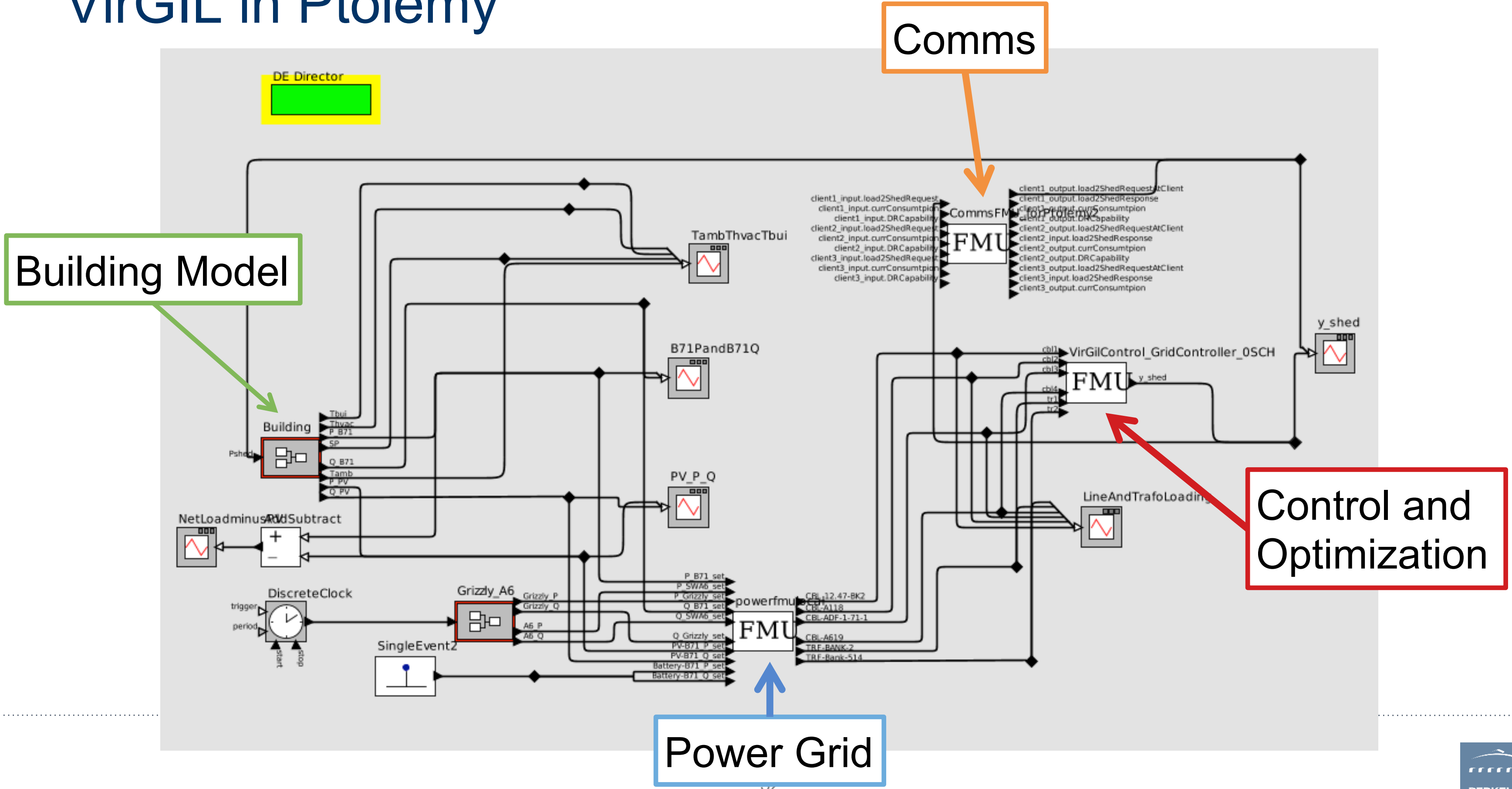
LBNL Distribution Grid to Building 71

- Model developed in DigSilent Powerfactory
- Real Network Data from LBNL facilities
- Real Consumption Data for the Aggregated Loads



BERKELEY LAB

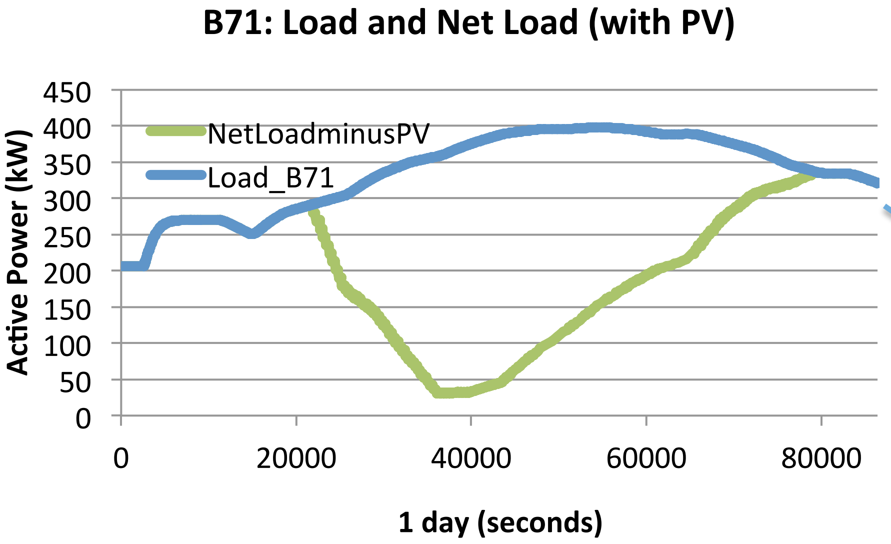
VirGIL in Ptolemy

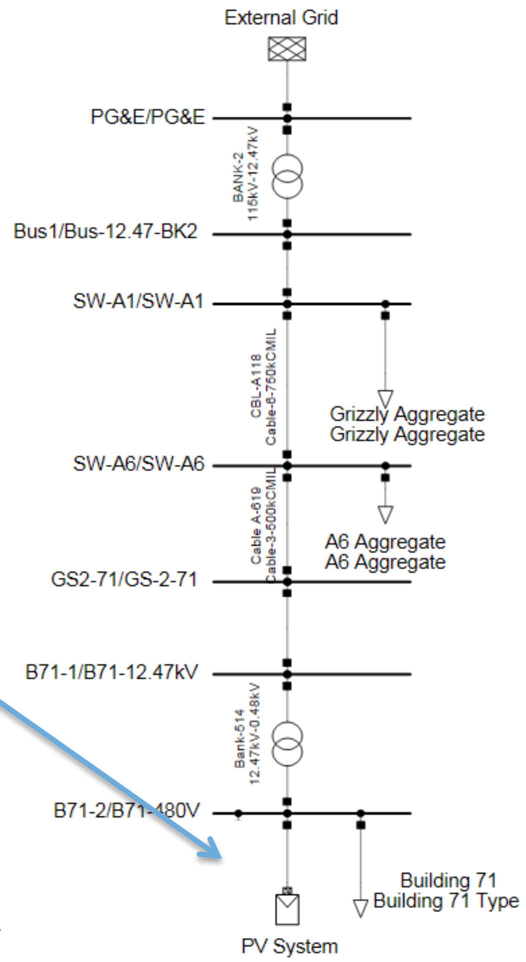


Outline

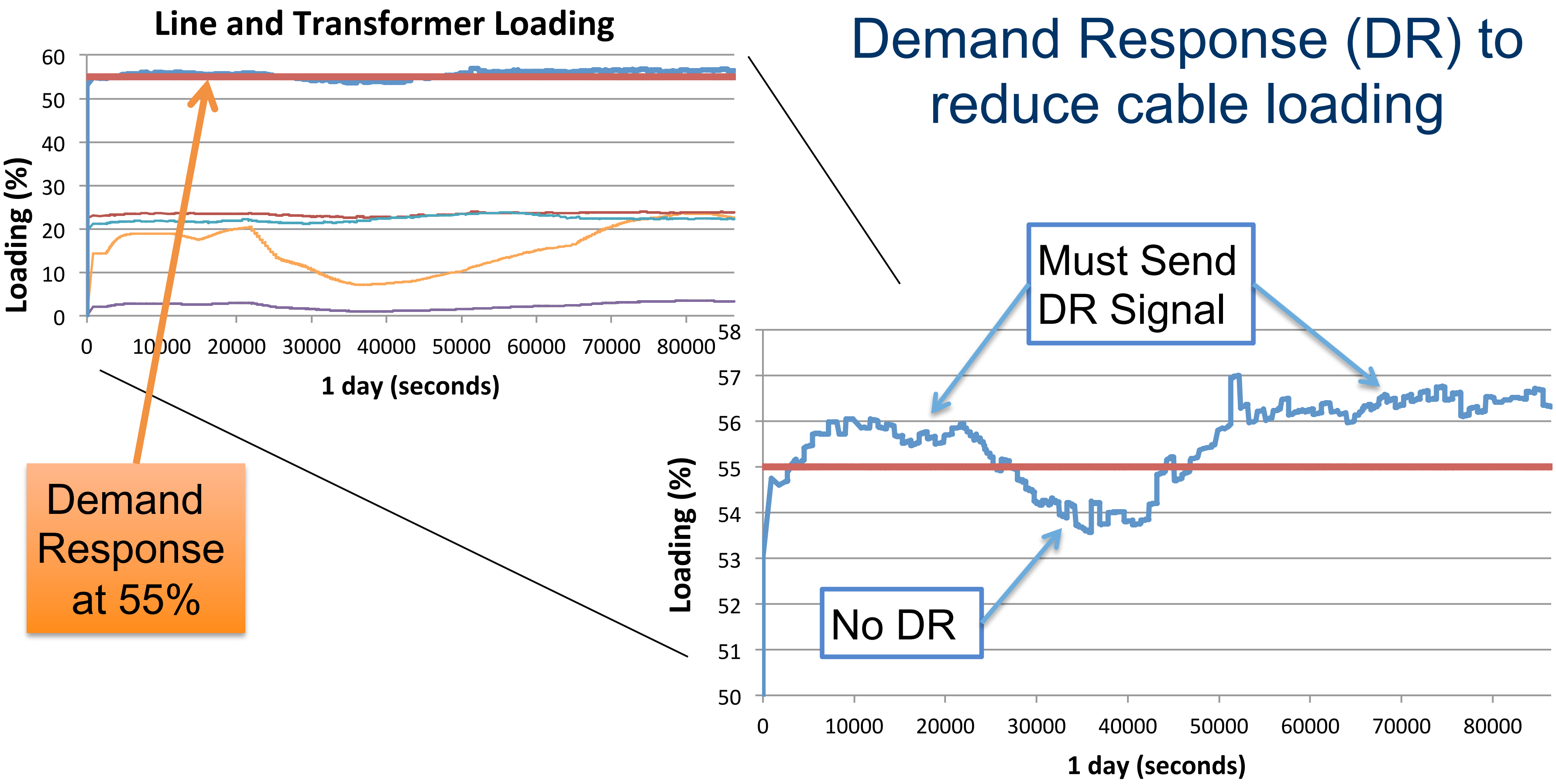
1. VirGIL Co-simulation Framework
2. Master Algorithm – Quantized State System
3. Building Modeling
4. Communications
5. Power System
6. **Use Cases**
7. Conclusions and Future Steps

Case #1: Demand Response (DR) to reduce cable loading

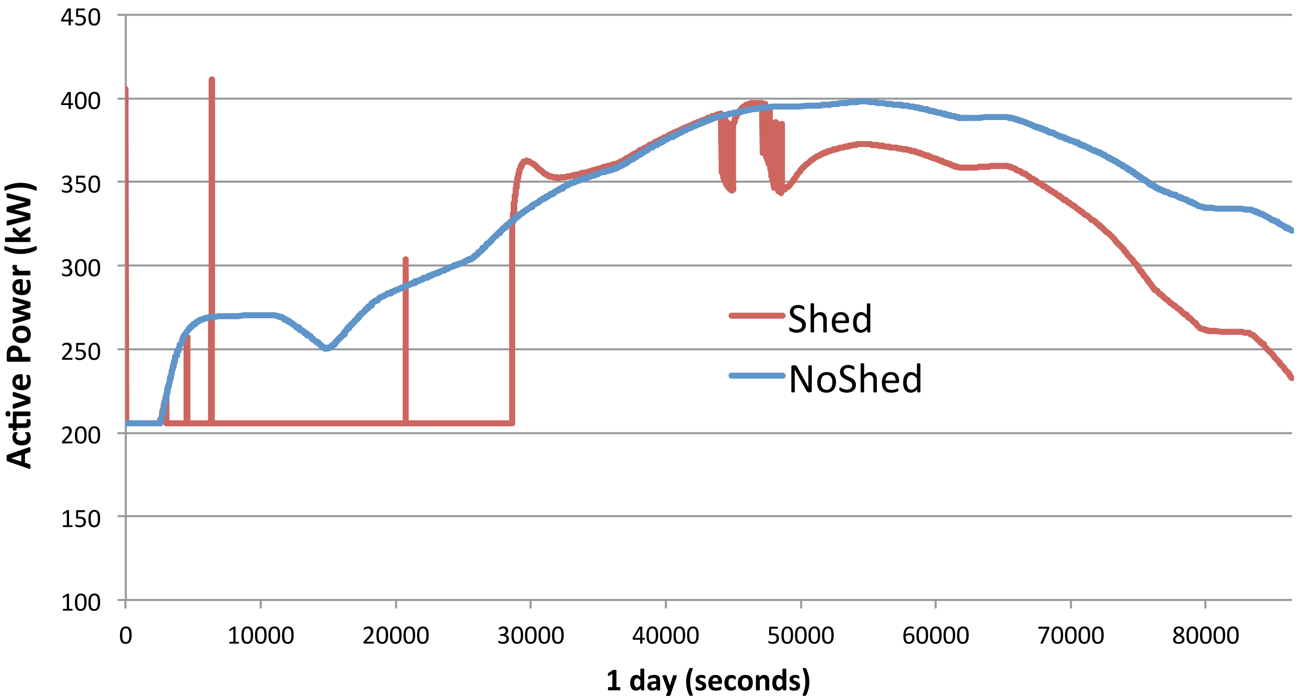




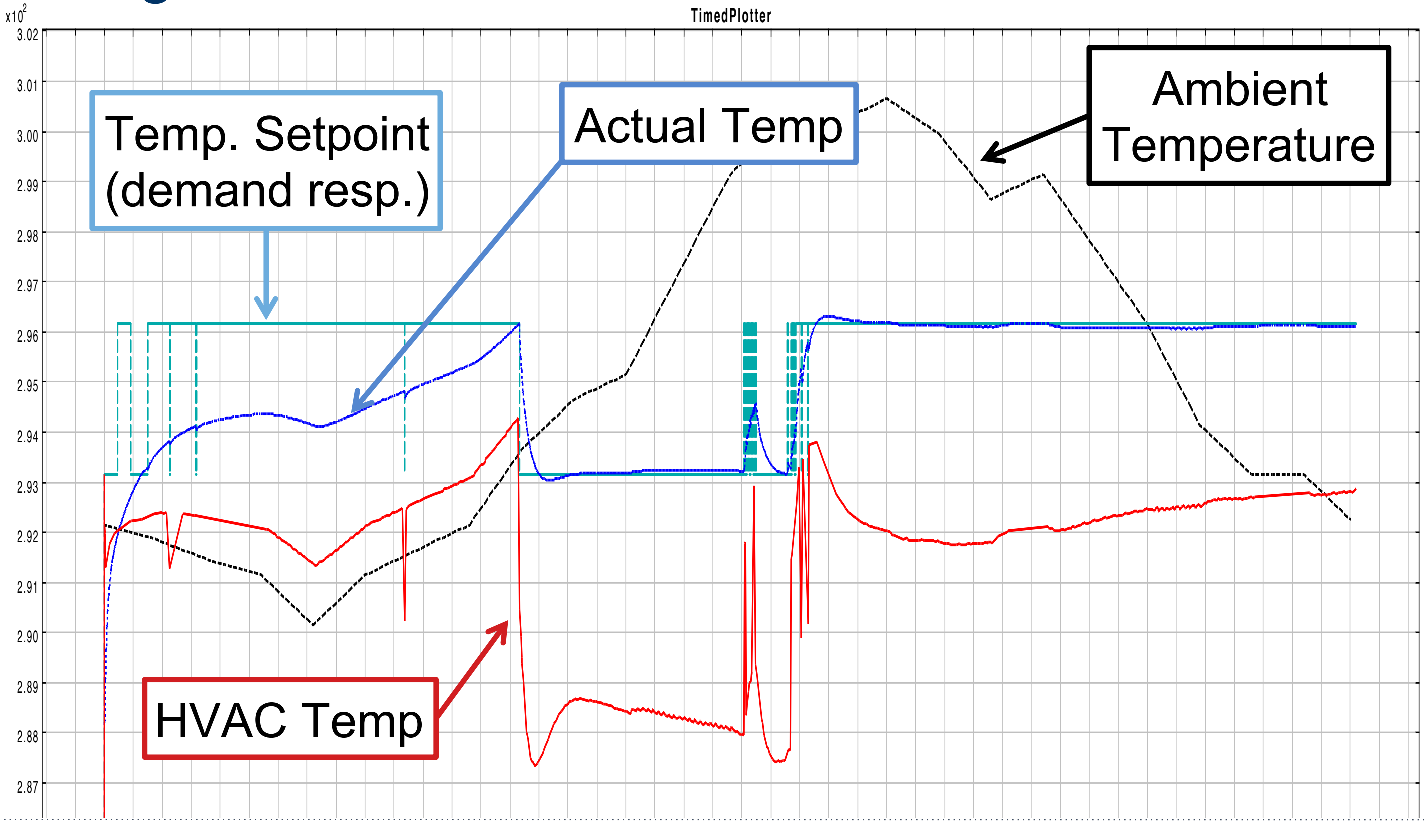
Demand Response (DR) to reduce cable loading



Building 71 consumption with and without Demand Response



Building 71 Control Actions



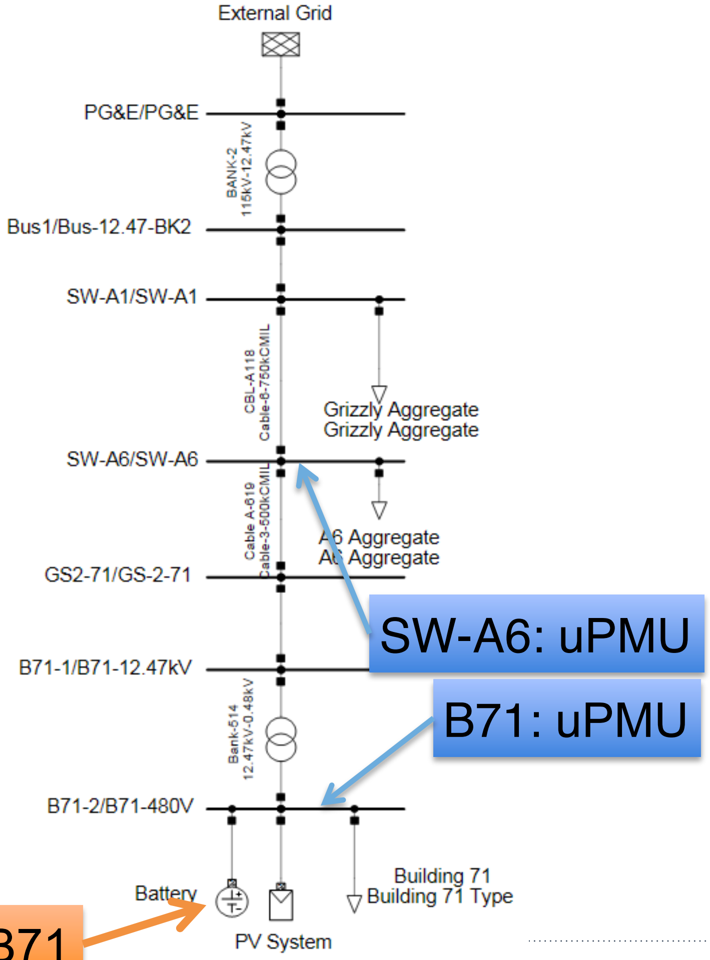
BERKELEY LAB

Case #2: Voltage Control with Battery

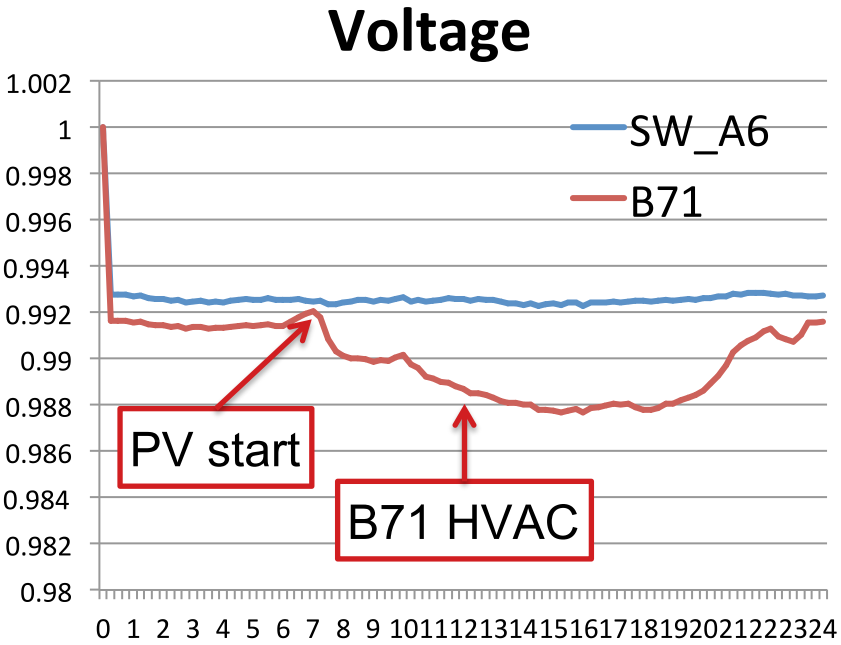
uPMU = micro-synchrophasor

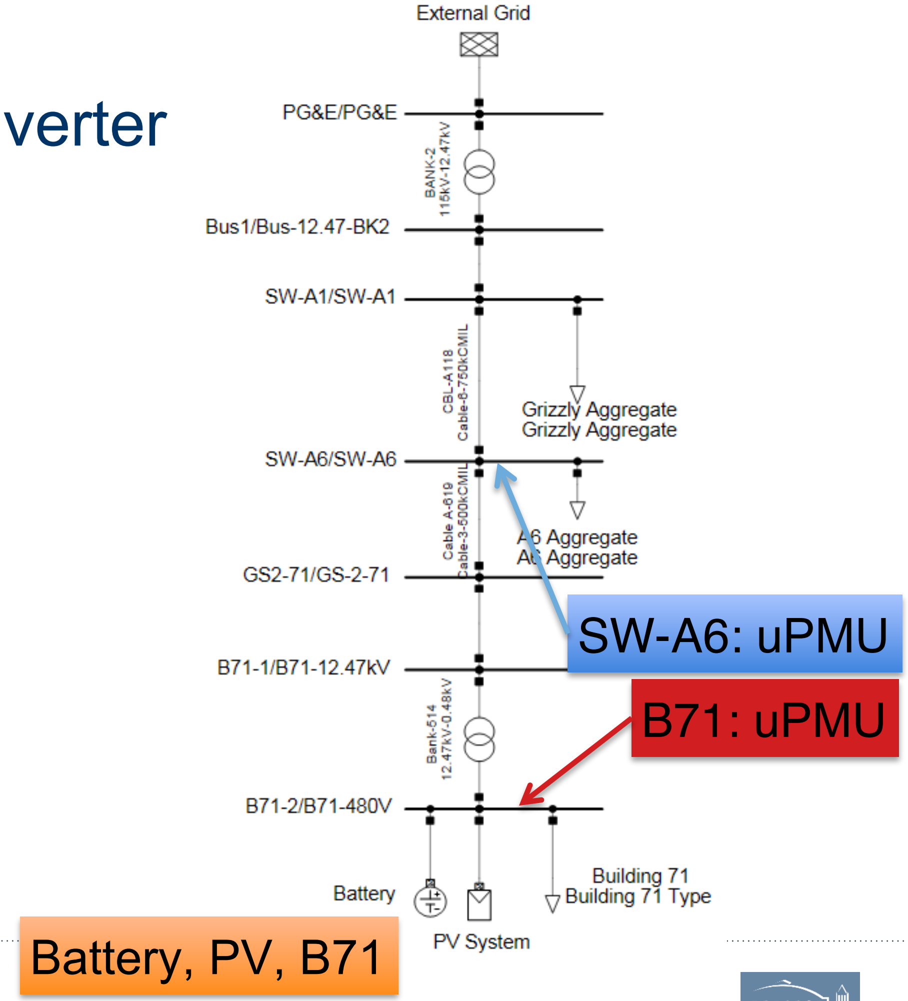
- Measure Voltage, Current, and *Voltage Angle* with High Fidelity
- LBNL has installed 3 microPMUs in the LBNL system

Control the Battery Inverter
→ Voltage at B71 should be equal to Voltage at SW-A6



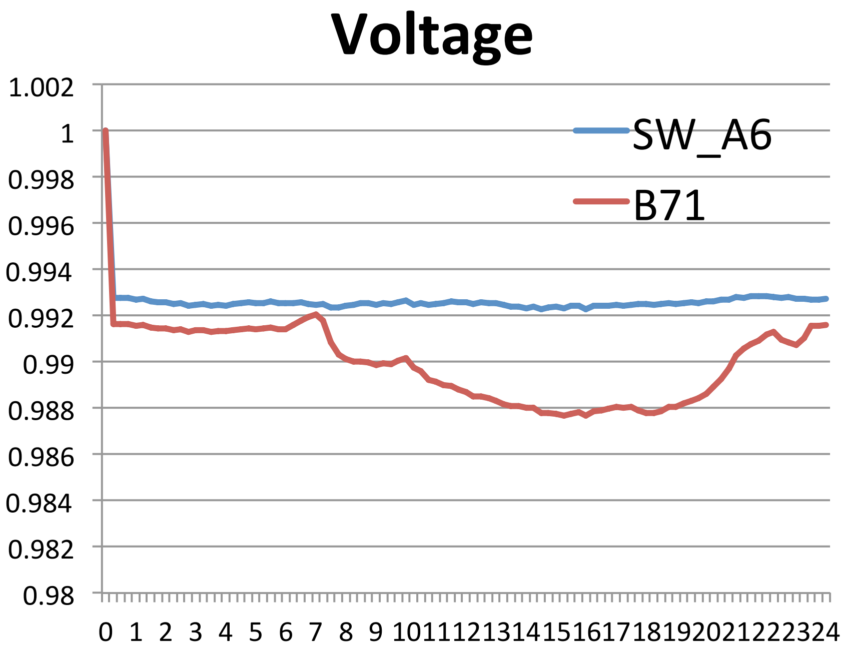
Voltage Control with Battery Inverter

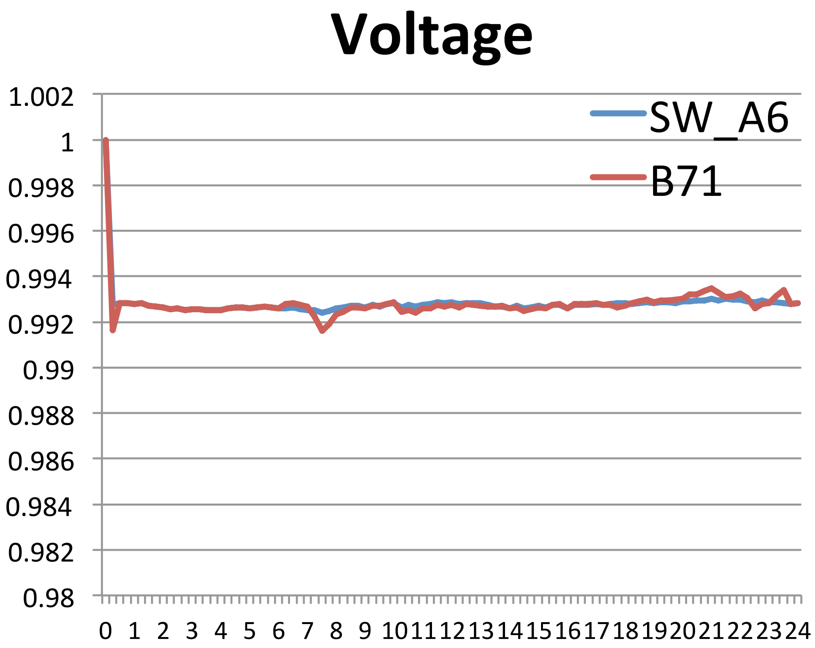




BERKELEY LAB

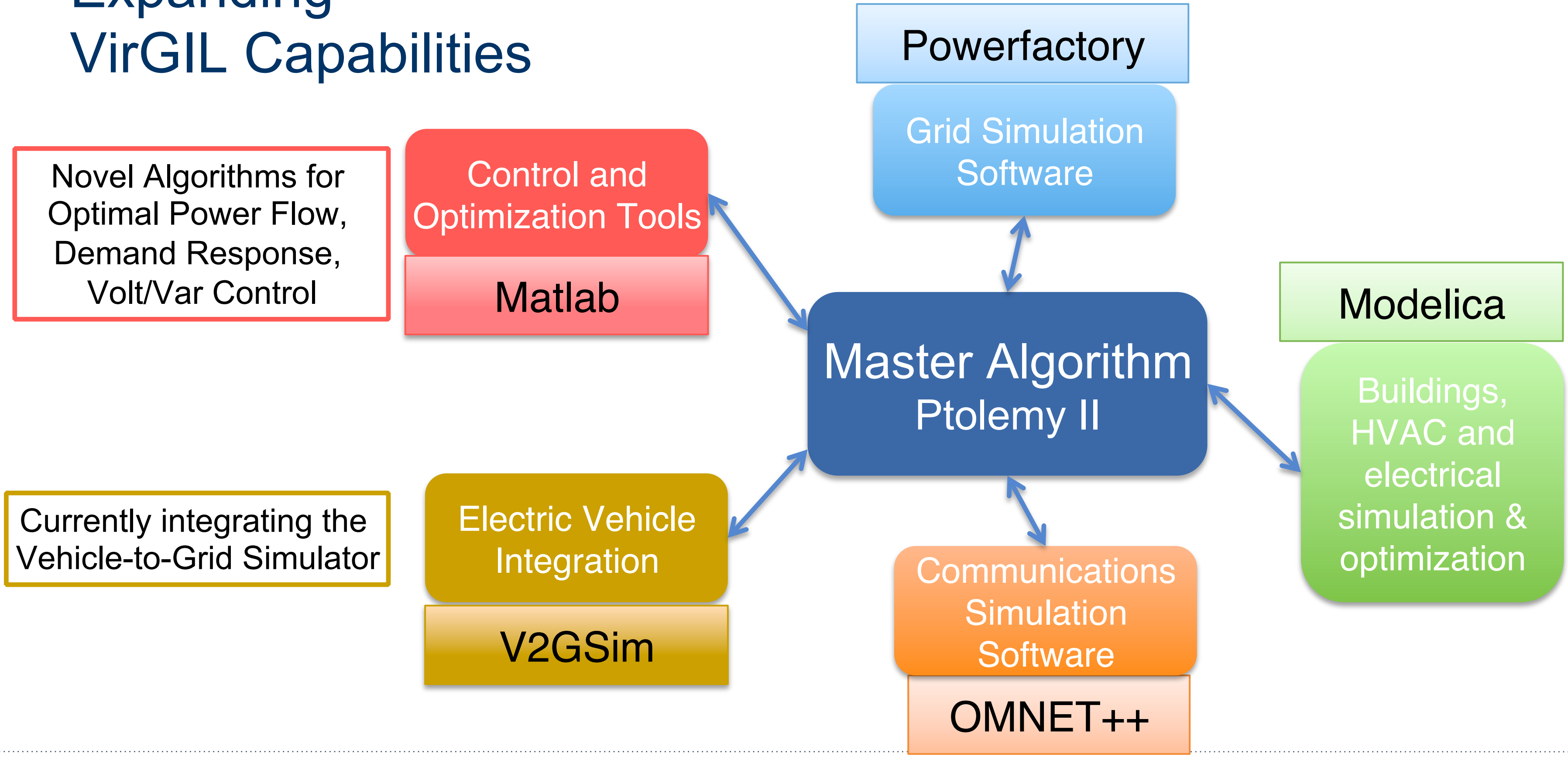
Voltage Control with Battery Inverter





BERKELEY LAB

Expanding VirGIL Capabilities

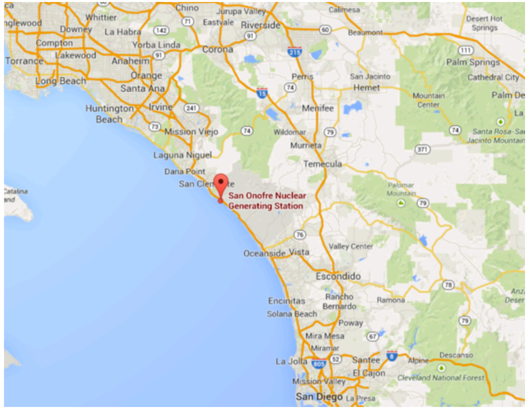


Next Steps

- Paper on VirGIL submitted to “Proceedings of the IEEE”

- Application of VirGIL to the SONGS* Region
  - Use of micro-synchrophasor data for advanced real-time demand response and control algorithms with VirGIL
  - Solar PV penetration scenarios coupled with Demand Response

- SONGS went offline in 2012 (steam generator problems)
  - Generated on average 16 million megawatt-hours annually (~8% of the electricity generated in California)
  - Close to Los Angeles and San Diego



*SONGS: San Onofre Nuclear Generating Station

BERKELEY LAB

Next Steps

- CyDER
  - SunShot SuNLaMP DOE Solicitation; concept paper accepted → full proposal
  - Extending VirGIL to perform QSTS (quasi-static time series), hardware-in-the-loop, EV charging, and *have seamless integration with the utilities tools*

- ExaGrid
  - Plans for LDRD FY16, FY17 with Computational Research Division
  - Parallelizing VirGIL: simulating ODEs in a discrete event domain on exascale architectures
  - Stochastic Optimization
  - Bringing VirGIL to the “cloud” (NERSC) to offer real-time services to utilities

Conclusions

- VirGIL allows us to investigate in detail the effects of Demand Response and DER to the power system
  - Captures the interactions between different complex systems
  - Enables the design of appropriate policies and control strategies for further integration of Demand Response and DER in power system operation
- Provides the foundations for a *real-time Demand Response Dispatch tool*
- A *modular* co-simulation platform which allows the *integration of several novel tools developed at LBNL* in existing commercial simulation and control software

Thank you!

schatzivasileiadis@lbl.gov