Modeling Intelligent Energy Systems

Peter Palensky
TU Delft
Smart Nord 10.2.2015
Peter Palensky

- TU Delft, Netherlands
  - Faculty for Electrical Engineering, Computer Science and Mathematics
  - Department for Electrical Sustainable Energy
  - Prof "Intelligent electric power grids"

- Worked in Russia, Germany, Korea, South Africa, California, Austria

- Methods/Theory for hybrid energy systems
  - Modeling, Simulation
  - Optimization

- Applications for hybrid energy systems
  - Controls, Stability
  - Integration
The future power system

- Expectations
  - Increased share of renewable energy sources
  - Host new applications like electric vehicles or cooperative loads
  - Optimized, resilient, flexible, robust, globalized, etc.

- That leads to an increment in:
  - Distributed structure
  - Control and management
  - New energy technologies & markets
  - Links to other “systems”

- Complex (hybrid / cyber-physical) power systems
Example Project / Motivation

- New energy market design and implementation
- Model-predictive load shed/shift
- Interoperability of equipment
- Information security
- System integration
  - PowerMatcher, DEMS, grid plausibility, market platform, CellController, etc.
  - OpenADR
- Fine grained distribution grid model parameters
- Intelligent demand side

DEMS: Decentralized Energy Management System (Siemens)
OpenADR: Open Automated Demand Response
Intelligent Loads

- Refrigerators as regulation power providers
- Frequency-dependent setpoint adjustment
- Distributed droop control
Intelligent Loads

- “GridFriendly” (PNNL)
- KNIVES (Japan)
- California
  - ORB
  - Smart AC
  - PCT
- “50.2 Hz problem” with 10 GW PV inverters in Germany 2011?

PNNL: Pacific Northwest National Laboratory
AC: Air Conditioning
PCT: Programmable Communicating Thermostat
PV: Photovoltaics
Intelligent Loads (OpenADR, bidding)
Research on future power systems

- Usually: Experimental
- Wanted: Model-based

- Four fundamental types of hybrid system elements
  - **Continuous**: energy technology, infrastructure, physics
  - **Discrete**: ICT, software, controls, communication
  - **Game Theory**: markets, market players, roles, agents
  - **Stochastic**: weather, people, aggregated/not-modeled behavior, statistics

- Scalability
  - Large (interconnected grids) <-> Small (microgrids)
  - Quick (frequency balance, harmonics) <-> Slow (weather, fuel price, demographics)
Use Case 1: Simple Hybrid System

- Thermal domain
- Discrete controller
- Agents/Market
- Stochastic events

- Describe via bond graph
- Analyze interplay of continuous domain and asynchronous events
- Scalability of platforms
Use Case 2: el. power station

- Physical parts not isolated
- Plus: Electrical domain
  - Ideal grid
  - Non-ideal power station
- Plus: Mechanical domain

- Further use cases
  - 3: Thermal grid
  - 4: Non-trivial market
  - 5: Communication network
  - 6: non-ideal grid
  - 7: EV-charging

EV: electric vehicle
Two types of Modeling Paradigms

- **Agent-oriented**
  - Autonomous modules
  - Components determine synchronization points
  - Examples: GridLAB-D, Omnet++

- **Monolithic**
  - Equation-based model of physics -> ODE-> code
  - Solver integrates and tries to find zero crossings
  - Examples: Modelica, Simscape

ODE: Ordinary Differential Equations
Monolithic Modeling

- E.g.: Simscape, Modelica
- PRO
  - Convenient
  - Multi-domain physics
  - Strong syntax
  - Good docu
- CON
  - Low Performance
  - Closed platforms?

\[
\text{Energy} \quad E_a + E_b + E_c = 0 \\
\text{Potential} \quad P_a = P_b = P_c
\]

External World of Component A

\[
\begin{align*}
E_a & \quad \text{Energy of Component A} \\
P_a & \quad \text{Potential of Component A} \\
E_b & \quad \text{Energy of Component B} \\
P_b & \quad \text{Potential of Component B} \\
E_c & \quad \text{Energy of Component C} \\
P_c & \quad \text{Potential of Component C}
\end{align*}
\]

Model

- flattening

Flat Model

- sorting

Sorted Equations

- Index reduction

Index Reduced Equations

Executable

- ODE

Challenge the future

12
Modelica / Simscape code example

```modelica
package Energy

package Interfaces

partial connector HeatPort

"Thermal port for 1-dim. Heat transfer"

Types.Temperature T;
flow Types.HeatFlowRate Q_flow;
end HeatPort;

end Interfaces;

package Components

model House4

"House lumped thermal heat"

Types.Temperature T(start=20 + 273.15,
displayUnit="degC") "Temperature of element";

parameter Energy.Types.ThermalCapacity Cth = 430.578 "Heat capacity of element";

parameter Types.Density rho = 1.2041;

parameter Types.Volume volume = 200;

Interfaces.HeatPort_a port_a;

equation

T = port_a.T;

rho*volume*Cth*der(T) = port_a.Q_flow;

end House4;

model Heater

... end Heater;

end Components;

end Energy;
```

```modelica
component ElHeater <

foundation.electrical.branch

nodes

M = foundation.thermal.thermal; % B:right

end

inputs

Level = { 0.50, '1' }; % :left

end

parameters

R = { 40, 'Ohm' }; % Resistance

end

variables

Q = { 0, 'J/s' }; end

function setup

through( Q, [], M.Q );

end

equations

v == R*i*Level;

Q == v*i;

end
```

Challenge the future 13
Multi physics, multi-everything...
Use Case 1 monolithic results

- Good for components!
- Events?
- Scalability?
- Libraries?
Agent oriented: GridLAB-D

Simulation environment **specifically designed** for the analysis of modern power systems
- open source, developed by PNNL

Main features:
- command line tool written in C/C++
  - runs under Windows (MSVC, MinGW, Cygwin) and Unices
- flexible agent-based simulator
  - can model the behavior of many objects over time
- comprises a modular design
  - buildings (residential, commercial, industrial)
  - electric network (generation, transmission, distribution, controllers, reliability)
  - markets (retail double auctions, transaction journals)
  - climate
- implements a modeling language
  - parametric syntax for dynamic model generation
- provides various simulation utilities
  - debugging, profiling, plotting, histogramming, write to file etc.
GridLAB-D: Results

- **Sophisticated time synchronization** of objects
  - each object has to **update** its **current state**
  - each object can tell when it wants to **update next**

  Good for systems...
  - Physics?, Libraries?
GridLAB-D: Results

- Sophisticated time synchronization of objects
  - each object has to update its current state
  - each object can tell when it wants to update next

Good for systems...
- Physics?, Libraries?

Diagram:
- Time
- Objects: Obj 1, Obj 2, Obj 3, Obj n
- Order of execution
GridLAB-D: Results

- Sophisticated time synchronization of objects
  - each object has to update its current state
  - each object can tell when it wants to update next

Good for systems...

Physics?, Libraries?

```
<table>
<thead>
<tr>
<th>Obj 1</th>
<th>Obj 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj 2</td>
<td>Obj 2</td>
</tr>
<tr>
<td>Obj 3</td>
<td>Obj 3</td>
</tr>
<tr>
<td>Obj n</td>
<td>Obj n</td>
</tr>
</tbody>
</table>
```

(order of execution)
Co-simulation environment

- Many possible ways to couple simulations
  - **Direct** coupling of tools
    - small overhead
    - typically not reusable
    - complicated for complex scenarios
  - **Generic** coupling of tools
    - introduces overhead
    - reusable
    - more flexible

**PowerFactory, OMNet++, 4DIAC, SCADA**

- **Power System Analysis**
- **Communication**
  - electrical signals, power, voltages
  - control signals
- **Components**
  - *Matlab/Simulink*
  - *ScadaBR / openSCADA*
Physics: causal vs. acausal modeling

- **Block diagrams** are very suitable for modeling of controls/signal processing
  - each block represents a *set of equations*
    - typically ordinary differential equations (ODE)
  - connections define **uni-directional** (causal) relations *between* blocks
    - i.e. between input signals, output signals and state variables
  - use **numerical solver** (ODE integrator) to compute successive states of system

- **Bi-directional** (acausal) connections are more intuitive for physics modeling
FMI for Model Exchange/Co-Simulation

- Functional Mock-Up Unit
  - model *interface* (shared library)
  - model *description* (XML file)

- Executable according to C API
  - *low-level* approach
  - most *fundamental* functionalities only
  - tool/platform *independent*

- FMI provides only well-defined access to the model
  - *master algorithm* definition left out on purpose!
Co-Simulation: thermal system simulation

- **domain-specific** co-simulation components
  - developed by domain experts

- **generic coupling** via Ptolemy II & FMI++

- mixed use of FMUs for *Model Exchange* and *Co-Simulation*
  - make use of what is available

- mixed used of *dynamic* and *fixed step* simulation
  - handled automatically by the environment

FMU: functional mockup unit
Co-Simulation: Power System, Communication & Controls

- Coupling of *event-based* and *continuous* simulation
- *Real-time* simulation
  - coupling with physical components (C-HIL)

Use Case: LV Network, OLTC control and PLC communication
Co-Simulation of hybrid systems

- UseCase 7: Flexible EV Charging as real-time demand response
- Co-Simulation
  - Gridlab-D -> Middleware
  - OpenModelica -> Components (Batteries, etc.)
  - PowerFactory -> el. grid
- Standardized Interface: Functional Mockup Interface (FMI)
- Combination of highly accurate physical models and large-scale system

EV: Electric Vehicle
Use Case 7: MATSim, EVSim, PowerFactory

Simulation Environment
Co-Simulation Results

SOC: State of Charge
Optimization of complex systems

- Uncertainties, Constraints, Risk, etc.

**Medium Voltage Substation Network**

**Quadratic Optimization problem**
- Constraints: 1587
- Scalar variables: 330
- Integer variables: 120
- Solver Used: MOSEK
- Platform: Windows/64-X86

**Objectives:** Cost Minimization

**Constraints:**
- Generator specific: Ramp rate, Minimum time for ON/OFF, power limits
- Network Specific: Power flow constraints, bus voltage limit
Hot Topics 2015++

- **Fundamentals**: Modeling and Simulation of (complex) power grids
  - Co-simulation, Modelica for power, power and controller HIL, etc.

- **Application**: Distributed, network-based controls of power grids
  - Stability, scalability, structure, resilience, self-organization, etc.

- **Interdisciplinary teams**
  - Mathematics, Computer Science, Physics, Electrical Engineering
  - Policy making, Markets, Socio-economic phenomena

- **International network**
  - UC Berkeley/Berkeley National Lab (us), NREL (us)
  - TU Delft (nl), AIT (at), DTU (dk), OFFIS (de)
Thank you!

Peter Palensky
Department of Electrical Sustainable Energy
Delft University of Technology
P.Palensky@tudelft.nl