Co-Simulation of Intelligent Energy Systems

Peter Palensky
TU Delft
“Grid-friendly” loads

- Thermal inertia, matter transport
- Multi-agent system
- Local load model
- Scalability?
“Grid-friendly”, smart buildings

- Building Agent
- CHP, heat pump, building controls, etc.
- Complex
  - Communication,
  - Controller,
  - Process,
  - People,...
What are we dealing with?

- **physical world**
  - continuous models
    - energy generation, transport, distribution, consumption, etc.

- **information technology**
  - discrete models
    - controllers, communication infrastructure, software, etc.

- **roles/individual behavior**
  - game theory models
    - agents acting on behalf of a customer, market players, etc.

- **statistical elements**
  - probabilistic models
    - weather, aggregated individuals, etc.

**cyber-physical energy system**
Use Case 1: Thermal System with Market

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

- Describe via bond graph
- Analyze interplay of continuous domain and asynchronous events
- Scalability of methods
Use Case 2: UC1 + el. power station

- Plus: Electrical domain
  - Ideal grid
  - Non-ideal power station
- Plus: Mechanical domain
- Tightly coupled elements!

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

EV: electric vehicle
Options for modeling a hybrid system

• (0) Squeeze submodels into one tool
  – Language, solver, method
  – n-1 submodels in wrong language
    • Tedious
    • Error prone
    • Brutal simplifications
• (1) Universal tool
  – Universal language, solver, etc.
• (2) Co-simulation
  – Combine specialized languages, solvers, etc.
(1) “Universal tool” approach

- (a) Agent-oriented / modular
  - Autonomous objects
  - GridLAB-D, Omnet++
- (b) Monolithic
  - Equation-based model
  - Modelica, Simscape
- Both
  - Use object oriented languages
  - Are compiled into executable
(1a) Agent-oriented models

- Model split into (large) number of autonomous objects
  - Communicate variables with each other
  - Determine synchronization points
- Discrete event scheduler coordinates
- Good scalability
(1a) Time synchronization

- Objects update their states
- Objects predict “personal” sync time
(1a) Time synchronization

- Objects update their states
- Objects predict “personal” sync time
(1a) Time synchronization

- Objects update their states
- Objects predict “personal” sync time
(1b) Monolithic models

- Simscape, Modelica
- Equations compiled into executable
- Numerical solver finds zero crossings
- Convenient, multi-domain physics
- Strong syntax, good docu
(1) “One Tool” bottom line...

- Both options not ideal...
- Missed events?
- Physics in agents?
- Performance?
- Parallel computing?

![Graph showing simulation time and number of houses](image)

Simulation time in seconds

Number of Houses

3 days model time

M xxx: Monolithic, A xxx: Agent based
(2) What is co-simulation?

• “normal simulation”
  - Model → ODEs/DAEs
  - One modeling tool, one solver (e.g. Euler, RK45, DASSL, ODE15s, etc.)

• Co-Simulation
  - Multiple solvers, multiple tools!
  - Coupled models

\[
\dot{x} = f(t, x) \\
x_{n+1} = x_n + h\Phi(t_n, x_n, h)
\]

- ODE: ordinary differential equation
- DAE: differential-algebraic equation
Co(upled) Simulation

- Multiple simulators
- Multiple models
- How to link?
- Scenario Handling?
- Interface?
Reality and models

Real system

Simulated system

Energy Market Simulator

Car Usage Simulator

Communication Network Simulator

Battery Simulator

Power Electronics/Controls Simulator

Distribution Grid Simulator

IEEE PES

17
Master Algorithm

- Initialize simulators
- Exchange variables
- Sync time
- User interaction
Synchronization

- Sync points = Macro steps
- Exchange variables

Diagram:
- Simulator 1
- Simulator 2
- Macro time step
- Micro time step
- Data exchange
FMI: Functional Mockup Interface

- Executable (C API)
  - low-level approach
  - tool/platform independent
- Good for physical models
- Growing support by tools
- Two version
  - FMI for co-simulation (contains solver)
  - FMI for model exchange
Proof of concept: Use Case 7 “dynamic demand-response”

- household load profiles taken from measurement campaign
- small scale distribution grid: medium/low voltage network with consumers
- realistic battery model
- charging control algorithm: distributed charging power regulation
- stochastic driving patterns derived from real-world car sharing data
Use Case 7 implementation (commercial)

- PowerFactory
  - Power system
  - Loads, generation
- Dymola/Modelica
  - Batteries
- GridLAB-D
  - Charging controls
  - Charging agents
  - Vehicle driving
Use Case 7 implementation (Open Source)

- PSAT/Octave
  - Power System
- Open Modelica
  - Batteries
- GridLAB-D
  - Vehicles, charging
- 4DIAC
  - Grid controls
Results: dynamic step size control
Results: details, details, details

• Detailed models
  – Unbalanced grid
  – Chemical battery
  – Emulated controls
    • Real-time simulation

• Multi-focus
  – Battery ageing
  – Grid limits
  – Agent convergence
  – Economic optimum
Results: ICT & Physics

- Impact of communication latency, packet loss, etc.
- Various communication channels (power line, etc.)
Splitting models

- Voltage and current source coupled
- Shared node/bus
Co-Simulation test model

Transmission (HV)

Distribution (MV)
Transient stability results (interface bus)

Benchmark

Distribution side

Voltage at interface bus

Current through interface bus
EMT results (interface bus)

Benchmark

Distribution side
- Sub-models can be encapsulated in composite actors
- Opaque composite actor
  - composite actor with its own director
  - not necessarily of the same MoC as at the upper level hierarchy
  - appear as „black box“ to the outside
Ptolemy II: superdense time

- In **superdense** time a **timestamp** consists of
  - model time \( t \)
  - microstep (also called index) \( n \)
- allows to handle simultaneous causally-related event

High Level Architecture (HLA)

- Standard for distributed computing
- Open source and commercial
Usage of co-model approach

- Smart Grids challenges require new methods
  - Co-model-based design, operations, controls, analytics
  - Integration of disciplines → interfaces

- Co-simulation is the way to go
  - For the time being...

- Trans-disciplinary problem
  → Trans-disciplinary team!
  - The real challenge...
Co-simulation of Smart Grids

- No off-the-shelf tool
- Many academic first steps
- Serious numeric questions
  - Stability, accuracy, multi-rate
- Biggest tasks are monkey jobs
  - Work-flow, standards, interfaces, user interface
- Needed: modular, open source platform
IEEE Workshop MSCPES

• Modeling and Simulation of Cyber-Physical Energy Systems
  – 2013 Berkeley
  – 2014 Berlin
  – 2015 Seattle
  – 2016 Vienna
  – 2017 Pittsburgh

• Part of CPS Week

• Join forces!
Be part of Smart Grids!
It is fun!