Modeling Challenges and Opportunities in Transient Simulations for Power Systems with Large Penetration of Converter-Interfaced Generation

Duncan Callaway
Jose Daniel Lara
Rodrigo Henriquez-Auba
University of California, Berkeley
{dcal, jdlara, rhenriquez}@berkeley.edu

PSERC Webinar
November 24, 2020
Next generation grids

Credit: Yashen Lin, NREL
Will they operate reliably?

- Low inertia: Early grid-scale work focused on swing dynamics
  - RoCoF
  - Nadir
  - Steady state

- However higher order dynamics are important, too
  - Small signal stability
  - Transient stability

Lin et al 2017

Undrill et al 2010
Today’s Objective

• Introduce a new simulation tool that supports scientific computing for large-scale dynamics simulations
  • design philosophy,
  • validation,
  • short tutorial
Example: Low Inertia Frequency Dynamics

Tayyebi et al.

- Explore the performance of different grid-forming inverter controls on frequency response metrics
- Customized 9-bus test system model built in Simulink

Markovic et al:

- Explore small signal stability for large-scale systems with high penetrations of power electronic converters
- Constructed a customized small-signal model
- Validated in Matlab Simulink

Example: What is the role of line dynamics?

Henriquez et al., explore the role of:

- line model assumptions
- grid forming control assumptions on region of s.s. stability.
- Leveraged recent Julia libraries to enable fast Jacobian calculations

Henriquez et al. “Grid Forming Inverter Small Signal Stability: Examining Role of Line and Voltage Dynamics” IECON 2020
Example: Teaching

“21st Century Power System Dynamics” (EECS 290O) at UC Berkeley

• Course objective: Equip students with the theoretical knowledge to model power system dynamics with voltage source converters and synchronous machines
• Semester project: Implement one grid-forming inverter model from the literature
• Tremendous effort involved in just getting models to run!
There is a gap in available computing tools

In each example, researchers and students had to build their own simulation models to study emerging questions on dynamics in low-inertia systems

- This takes time (a significant part of someone’s PhD…)
- Hinders reproducibility
- Slows down the review process

We argue that principles of scientific computing are ripe for application in power systems research, and will address the issues above.
Scientific computing principles for power systems research

1. **Data Process**: experiment parameters, test system for the experiment, number of sample sets of confounding variables.

2. **Computing Process**: This enables investigating a range of discrete simulation scenarios

3. **Results and Reporting Process**: Report distribution of results across distribution of confounding variables.

Q: What’s needed to facilitate scientific computing in power systems research?

...Especially if we wish to focus on large-scale systems with high penetration of converter-interfaced generation.

A: Open-source tools for power system analysis

- Built in a fast, interactive, technical language
- Separation between modeling and algorithms
- Modular component descriptions, EMT capabilities
- Seamless capacity for scripting and automating scenario generation and execution
- Parse industry standard data files, validation against industry standard tools.
"As high-level and interactive as Matlab or Python+IPython,
As general-purpose as Python,
As productive for technical work as Matlab or Python+SciPy,
...but as fast as C."

- Steven Johnson, MIT Applied Math
Examples of existing open-source tools

- **Power flow and OPF:**
  - MATPOWER (Matlab Based)
  - PyPower (Python Based)
  - PowerModels.jl (Julia Based)

- **Dynamics:**
  - PSAT (Power flow, dynamics and stability, harmonics) (Matlab Based)
  - ANDES (Python Based)
  - PST (Matlab Based)
  - iTesla Power Systems Library (iPSL) (OpenModelica Based)
  - GridDyn (C++ Based)
PowerSimulationsDynamics.jl (PSID)

PSID is a Julia-based open-source power system modeling and simulation toolbox designed to study system stability in large-scale, low-inertia power systems.

- Data models & modularity facilitate fast model development.
- Julia: provides cutting edge solvers for large, stiff systems

Our vision: enable scientific computing approaches to EMT simulations to study emerging power system dynamics topics. For example,

- Rigorous study of model complexity vs fidelity
- Benchmarking emerging converter control strategies
Software Architecture

**Data model and inputs**
- PSS/e raw and dyr files
- Matpower power flow cases
- PSAT matlab files (upcoming)

**Model Formulation**
- Implicit Model
- Explicit Model
- Small Signal Stability

**Integration Algorithm**
- Implicit:
  - BDF (through Sundials.jl)
  - BDF method pure Julia
- Mass Matrix:
  - Rosenbrock Methods
  - Rosenbrock-W Methods
  - FIRK Methods
  - SDIRK Methods
PowerSystems.jl

- PowerSystems.jl is a package to organize and manipulate data with diverse modeling requirements.
- Provides a generic data model for the simulations
- Implements the metamodels for machines and inverters.
PowerSystems.jl

- PowerSystems.jl currently can parse pss/e dynamic data files.
- Once the system is read, it can be modified and serialized to disk with the modifications.
- Any data additions or modification can be recorded in reproducible scripts.

For additional details, check: https://github.com/NREL-SIIP/PowerSystems.jl
Modeling Strategy

- A key aspect is that PSID is driven by the data model in PowerSystems.jl.
- PSID automatically constructs DAEs from data model.
- Can use metamodels for dynamic devices → allows PSID to construct mathematical models with different levels of stiffness.
Modular Inverter "Meta" Model

Electric Power Grid

$V_{ri}, I_{ri}$

Reference Frame Conversion

$V_{dq}, I_{dq}$

Filter
AC Voltage and Current Dynamics

$V_{dq}^{\text{conv}}, I_{dq}^{\text{conv}}$

Converter
PWM Dynamics

$V_{dc}$

$p = \text{Re}[V_{dq}I_{dq}^{\text{*}}]$ (DC Source)

Energy Source and DC-side Dynamics

$P_{ref}$

Inner Loop Control
Voltage Control, Current Control and Virtual Impedance Dynamics

$p = \text{Re}[V_{dq}I_{dq}^{\text{*}}]$ (Frequency Estimator)

PLL Dynamics

Outer Loop Control
Active and Reactive Power Control Dynamics

$V_{ref}, \omega_{ref}, Q_{ref}$

$p = \text{Re}[V_{dq}I_{dq}^{\text{*}}], q = \text{Im}[V_{dq}I_{dq}^{\text{*}}]$
Running a simulation

• The simulation specification is based on methods defined to model each dynamic component.
• Reads directly the information from the data in the system.
• Define the perturbations into the system:
  • NetworkSwitch: Used for large network reconfigurations
  • BranchTrip: Trip a Line or Transformer in the system
  • ControlReferenceChange: Change the reference points on a device
• Define the timespan
Initialization Procedure

Initialization for each device

Generator:

- Power Flow Solution
- Synch. Machine + Shaft
- Turbine Governor
- AVR
- PSS

Diagram:

- $V, \theta$ from Power Flow Solution
- $p, q$ from Power Flow Solution
- $V_f$ from Synch. Machine + Shaft
- $\tau_m, \omega$ from Turbine Governor
- $V_{ref}$ from AVR
- $p_e, \omega$ from PSS
Initialization Procedure

Initialization for each device

**Inverter:**

- **Power Flow Solution**
  - $V, \theta$
  - $p, q$

- **Filter**
  - $V_r, \theta$
  - $V_r, \theta$

- **DC Side**
  - $V_{dc}$

- **PLL**
  - $\delta_{pll}$
  - $\omega_{pll}$

- **Outer Loop**
  - $\delta_{olc}$

- **PWM**
  - $m_{dq}$

- **Inner Loop**

Variables:
- $p_{control}$
- $\theta_{control}$
- $V_{ri}$, $I_{ri}$, $I_{ri}$
- $V_{control}$, $I_{control}$, $I_{control}$
- $q_{control}$
Software comparison

- We rely on computational tools to accurately represent power systems.
- Any new software must be validated against industry accepted software tools.
- Multiple options: PSS/E, PSLF, PSCAD, DIgSILENT, PowerWorld, EUROSTAG.
- Parser for PSS/E already available in PowerSystems.jl

T. Overbye, M. Venkatsubramanian “Validation and Accreditation of Transient Stability Results”
PSERC Publication 11-08, September 2011.
Validation PSID vs PSS/E

- PSID implementation based on DAE.
- PSS/E models implementation: block diagrams into differential equations.
- Different options for implementing saturation functions and anti-windup.
- Solver algorithms and tolerances will affect results
- Validation of each model via:
  - Operating point (steady-state initialization)
  - Transient simulation under a disturbance

- 3-bus test case:
Validation PSID vs PSS/E

Validation GENROU + AC1A + TGOV1: 3 buses with IB
Validation PSID vs PSS/E

Validation: Voltages IEEE 14 Bus - GENROU - GAST
Validation PSID vs PSS/E

Validation: Speed IEEE 14 Bus - GENROU + GAST
# Current Generator Modeling Capabilities

<table>
<thead>
<tr>
<th>Machine</th>
<th>Shaft</th>
<th>Turbine Governor</th>
<th>AVR</th>
<th>PSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic (GENCLS)</td>
<td>Single-Shaft</td>
<td>TGType I (PSAT)</td>
<td>Type II (PSAT)</td>
<td>SimplePSS (PSAT)</td>
</tr>
<tr>
<td>GENROU/E</td>
<td>Five-Mass-Shaft (PSAT)</td>
<td>TGType II (PSAT)</td>
<td>Type II (PSAT)</td>
<td></td>
</tr>
<tr>
<td>GENSAL/E</td>
<td></td>
<td>TGOV1 (PSS/E)</td>
<td>ESAC1A (PSS/E)</td>
<td></td>
</tr>
<tr>
<td>One d- One q-machine (PSAT)</td>
<td></td>
<td>GAST (PSS/E)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anderson-Fouad Simplified (PSAT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anderson-Fouad (PSAT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marconato Simplified (PSAT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marconato (PSAT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Current CIG Modeling Capabilities

<table>
<thead>
<tr>
<th>Filter</th>
<th>LCL (6-states)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converter (PWM)</td>
<td>AverageDynamics</td>
</tr>
<tr>
<td>Inner Loop Control</td>
<td>Voltage/Current PI Controller (4-states)</td>
</tr>
<tr>
<td>Outer Loop Control</td>
<td>Virtual Inertia + QV droop (3-states)</td>
</tr>
<tr>
<td>Frequency Estimator</td>
<td>Kaura PLL (4-states)</td>
</tr>
<tr>
<td>DC Source</td>
<td>Fixed-DC Source</td>
</tr>
</tbody>
</table>

Inverter validation:

Demonstration with 14-Bus System

- **Gen1:**
  - GENROU,
  - GAST,
  - ESAC1A

- **Gens 2 - 5:**
  - GENROU,
  - ESAC1A
  - Fixed Turbine Governor output

Follow along in https://bit.ly/3mckKUI
Demonstration with 14-Bus System

Remove Generator 4 and substitute with storage using a Virtual Synchronous Machine in the same bus.

To an example of code use ….
How to get involved

• PSID is functional, but still in development!
• We encourage you to:
  • use the tool and flag areas for improvement
  • take part in open source development
  • Join our slack channel (NREL-SIIP)
What’s next?

● Home-turf research:
  ○ DOE-sponsored collaboration on computing tools for accelerating simulation and learning regions of transient stability
  ○ Exploring standards for dynamic simulation with CIG

● Continued development of component libraries
● Expanded research community involvement
Acknowledgements to:
NREL SIIP LDRD, Led by Clayton Barrows

https://github.com/NREL-SIIP/PowerSimulationsDynamics.jl