STEVE SOMMERVILLE

POWER SYSTEMS · AI · RESEARCH

Technical Note 003: Validating and Parameterising a CCGT plant

N-003

This is preliminary research and has not been validated or peer-reviewed. It could therefore be totally wrong.

Key Concepts

- PSCAD is a great tool, but it has several limitations, and it can get challenging on large systems with lots of data points and when trying to set specific MW and MVAr loading points.

- Modelling large synchronous machines with multiple controllers with different parameters is a slow and tedious process and very easy to make mistakes.

- Using AI (Claude) with an MCP interface and an appropriate toolkit it is possible to directly interface to PSCAD and obtain results and validate the model.

  - Inspect and interrogate the system prior to running any simulations to make sure the network is modelled correctly and all parameters are correct.

  - Estimate and refine the synchronous machine voltage and angle to get specific MW and MVAr loading cases – avoiding the need to build a parallel model in DIgSILENT or PSS/E and manually tuning.

The Problem

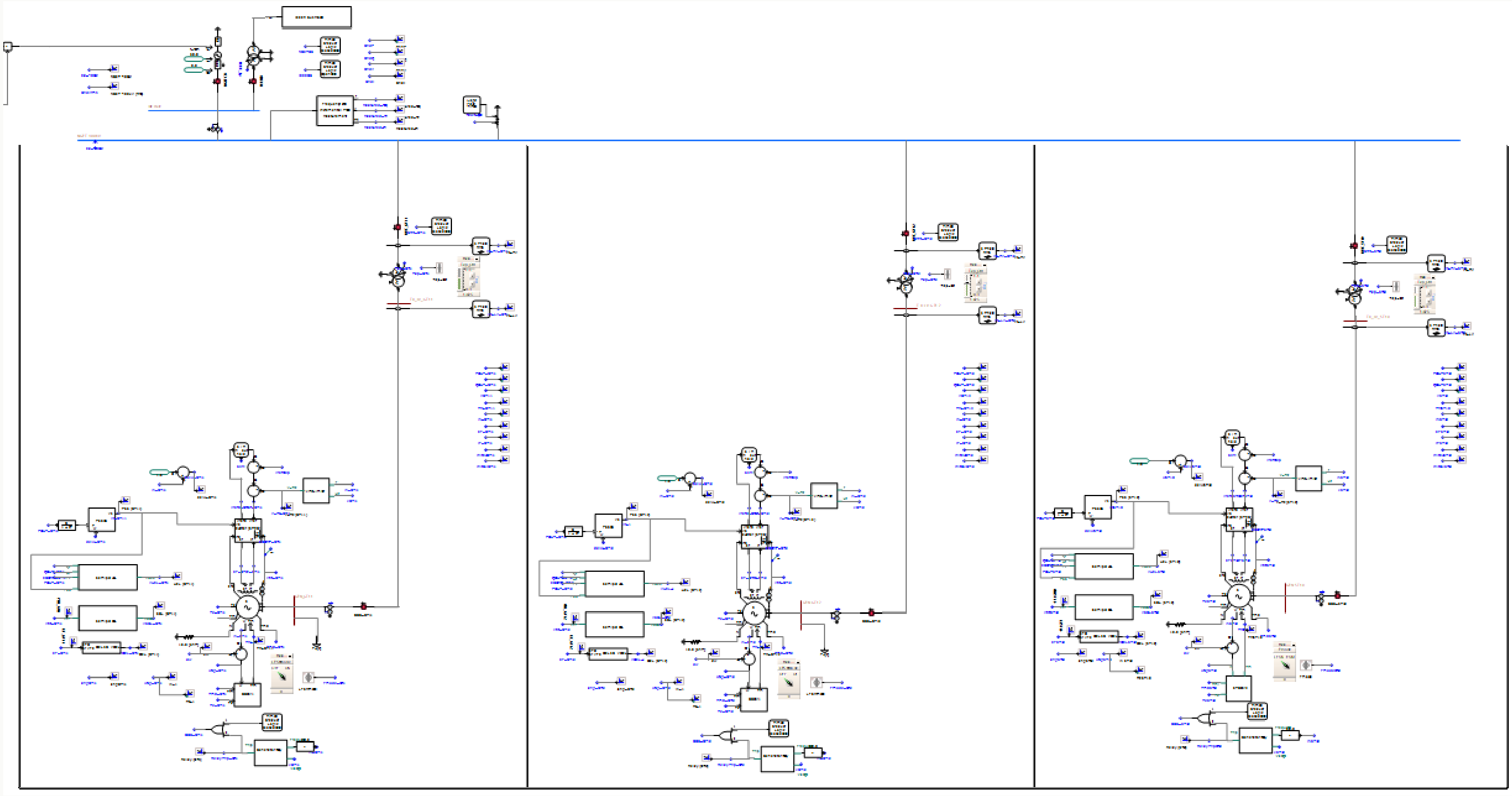
- The exercise was to build a detailed model of a 1000MW CCGT plant with 2x identical Gas Turbine Generators and 1x Steam Turbine Generator, and then run a series of simulation cases with it.

- With three very similar synchronous generators and a lot of study cases to run it is important to make sure all the machine parameters and controller settings aligned with the provided information. Data entry and setting of variables is very easy to get wrong and can lead to a lot of rework.

- A number of different MW and MVAr loading cases were necessary for different scenarios. This would normally mean manually setting each machine voltage and angle to define the required power flow. This is incredibly tedious and time consuming.

PSCAD Network Model

- 2x Identical GTGs and 1x STG, connected to a common 400kV busbar with an equivalent Grid model.



Validation and Parameter Check

- Each synchronous machine has a full set of DQ axis frame parameters that were obtained from an old PDF datasheet and manually entered the model. The Claude MCP server was used to scan the PDF and then cross-checked:
  - Most of the parameters were right, but a couple of typos
  - It is also checked and converted the leakage reactance and Potier reactance to an Air Gap Factor, without me asking it to.
  - It picked up that the default mechanical windage and friction losses were too high at 0.05pu and suggested a much lower value.

- A full set of AVR datasheets was also fed into Claude and checked:
  - AVR parameters were checked and all ok.
  - The OEL, UEL, SCL, V/Hz and PSS settings had a couple of typos spotted
  - It picked up some interesting inconsistence on which ambient air temperature to use for the SCL and OEL and a 40°C or 20 °C ambient
  - It spotted that the machine inertia didn't match the value used in the PSS implementation (KS2 = T7/2H). I had used just the alternator inertia rather than the inertia of the full alternator+turbine.

Validation and Parameter Check

Check	GT1	GT2	ST	Result / notes
Machine base (MVA, Vt)	!	!	✓	machine data & PSS scaling self-consistent
Inertia H vs PSS KS2 = T7/2H	✓	✓	✓	machine data & PSS scaling self-consistent
Converter (KAS, TAS, VBmax, KC)	✓	✓	✓	all per OEM
AVR ST9C core (KA, TA, KU, TCD/TBD/ZA, VR limits)	✓	✓	✓	correctly different per own doc; VRmax/min all units
Limiter topology & signal wiring	✓	✓	✓	UEL take-over, OEL summation, SCLu summation (= OEM VSCLsum), SCLo none; per-machine labels, no collisions
PSS parameters (PSS2B section)	✓	✓	✓	All consistent – washout setting Tw4 set OOS. Inertia is inconsistent with setting
PSS structure (PSS2C 4th stage T12/T13)	✗	✗	✓	exact
PSS on/off hysteresis (PPSSon/off)	✗	✗	✗	Not in master:pss — PSS permanently in service; only matters below 0.15 pu MW
OEL thermal (OEL3C) + OELmax	✓	✓	✓	40 °C basis
UEL (KQB, gains, Qtref curve)	!	!	✓	all per OEM
SCL (structure, KQ, Qmin, TAlt, ITLmax)	✓	✓	✓	all per OEM
SCL setpoint temperature basis	!	!	!	all per OEM
V/Hz limiter	✓	✓	✓	all per OEM
GT1 ≡ GT2 (all blocks incl. governor, GSU)	✓ identical			15 differences total — every one a per-machine signal name; zero parameter drift

Calculation of Machine Settings for MW and MVAr

- After validating the model, it was time to calculate the MW and MVAr. This is part of a process that is very tedious.

- If you cannot remember the math, the (simplified) formulas are below:
  - $P = \frac{V_1 V_2}{X} sin\delta$
  - $Q = \frac{V_1 V_2}{X} cos\delta - \frac{V_2}{X}$

- This was set as a problem for Claude, and I was not entirely sure how it was going to tackle the problem. I did tell it to control the voltage through the transformer tap controllers rather than the machine voltage though. It produced a clever idea:
  1. **Step 1 — Extract the network from the model itself.** Rather than running through the GUI with an MCP interface, the .pscx project XML was parsed directly. MCP was used to cross check.
  2. **Step 2 — Build a first-principles phasor model.** It created a ~100-line positive-sequence solver in python
  3. **Step 3 — Calibration**. It ran a number of flat simulations with no events to check the machine output and the overall output matched expectations to <0.1%.

Summary

- The use of AI was a huge help in this and is very much playing to the AI strengths, of mathematical modelling and a direct MCP plug-in to PSCAD.

- Inspecting a model against a set of PDF datasheets is exactly the sort of things that AI is great at. It takes minutes and is accurate and saves a huge amount of time, before committing to the run.

- Using Claude to build a simple first principles phasor model of the system to estimate the machine voltage and angle.

- The really interesting point was the fact that Claude managed to build a phasor domain model to solve and estimate the parameters from first principles. The obvious issue here is that it doesn't scale up well onto more difficult and complex systems

- For future work I am going to explore using PSCAD and Pandapower in parallel to see what we can do for more complex loadflow interface and solvers. Initial trials look promising!