



# Verify & Validate + Calibrate Tool (V<sup>2</sup>Cal) for Automated Model Parameters Tuning

P. Etingov, S. Biswas



PNNL is operated by Battelle for the U.S. Department of Energy

PNNL-SA-221608

## MOD-026-2 (new approved NERC standard)

- **MOD-026-2 replaces MOD-026-1 and MOD-027-1**
  - **MOD-026-1:** validation of excitation / Volt-Var control models
  - **MOD-027-1:** validation of turbine-governor / frequency-power control models
- MOD-026-2 expands validation to **modern BPS resources**, including **IBRs, FACTS, HVDC, and synchronous condensers**
- **EMT simulation is important** for large disturbances because it captures fast controls, nonlinear behavior, and protection actions that **positive-sequence models may miss**
- **Positive-sequence models** are still used for bulk planning studies, but they should be checked against **EMT model response** for severe events
- **Model validation is essential** to ensure study models match actual field behavior and produce credible planning results

<https://www.nerc.com/standards/reliability-standards/mod/mod-026-2>

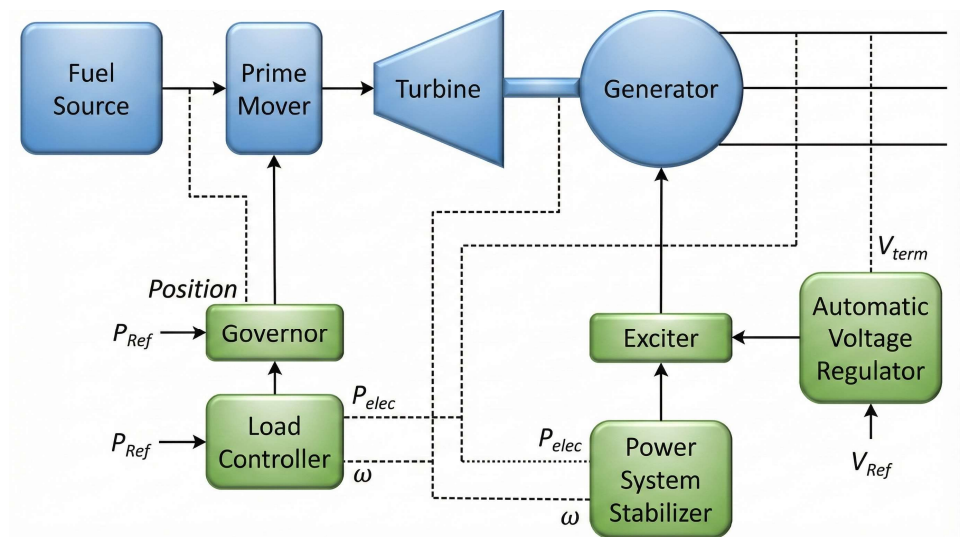
## V<sup>2</sup>Cal: Verify & Validate + Calibrate Tool

- V<sup>2</sup>Cal is a Python-orchestrated workflow that automates end-to-end play-in studies.
  - Supports PSS®E with extensibility to PSLF/PowerWorld and EMT solvers
  - Ingests PMU/DFR (SCADA optional), prepares channels, builds cases
  - Runs simulations, sensitivity analysis, and Bayesian calibration
  - Produces interactive, shareable HTML reports
  - Replaces manual data prep and tuning, reducing analyst burden and streamlining measurement-based verification
  - Supports NERC MOD studies by streamlining measurement-based model verification and validation workflows
- Will be released on GitHub (pending DOE approval):  
<https://github.com/pnnl/v2cal>

# Power Plant Model Verification

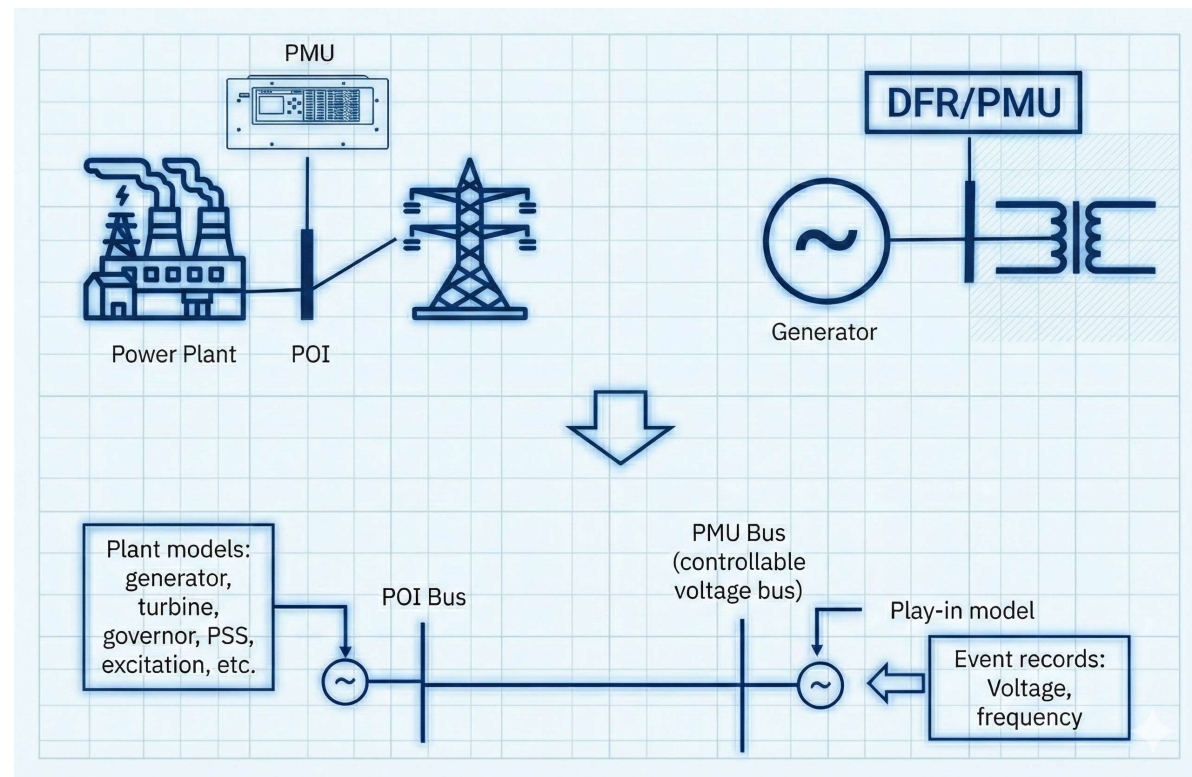
- Verify model structure
- Verify model parameters
- Scope typically includes:
  - Generator (synchronous machine)
  - Turbine – Governor
  - Excitation system/AVR
  - Power system stabilizer (PSS)
  - If applicable: limiters, turbine load controller, protection

## Conventional Power Plant Structure



## Play-in Model Validation

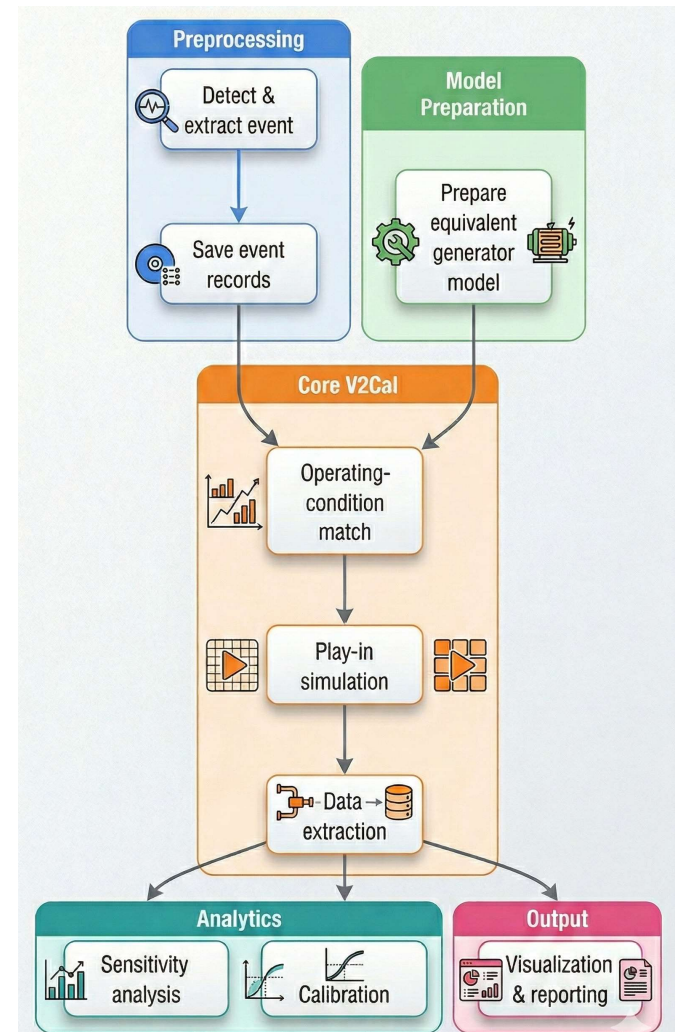
- **Data Requirement:** High-fidelity PMU/DFR data at the POI or Generator.
- **Method:** "Play-in" recorded Voltage/Frequency as a controllable source during simulation.
- **Benefit:** Enables validation of specific plant components (e.g., Governor, Exciter) without simulating the full network.





## V<sup>2</sup>Cal Workflow Overview

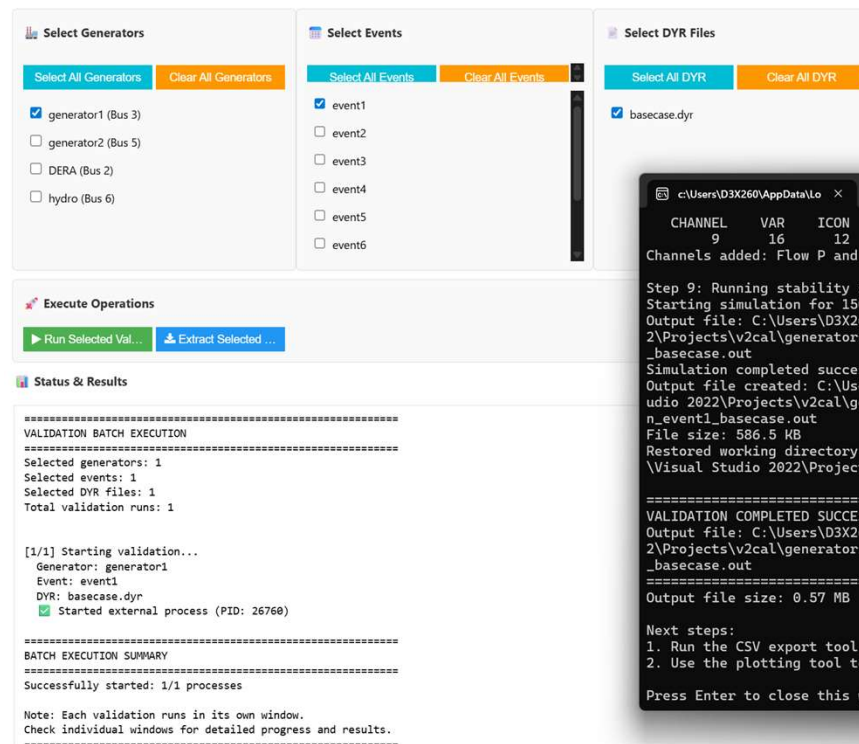
- Initial data preprocessing
  - Detect and extract disturbance event
  - Save event records in PSS®E-compatible CSV format
- Model preparation - Build/prepare the equivalent generator model for play-in studies
- Core V<sup>2</sup>Cal modules
  - Operating-condition matching between power-flow case and measurements
  - Play-in simulation (main validation step) using measured V/F as inputs
  - Data extraction from binary channel files
- Visualization & reporting - Generate interactive, shareable HTML reports
- Additional analytics
  - Sensitivity analysis to identify impactful parameters
  - Calibration of model parameters using Bayesian optimization



## V<sup>2</sup>Cal Architecture

- Pure Python application
- GUI implemented in Jupyter Notebook
- Easy to install, setup and run
- Communicates with PSS®E through native Python API
- Transparent, easy-to-modify codebase compared to previously developed PPMV tool

### PSS/E Model Validation Tool



**Select Generators**

Select All Generators Clear All Generators

☒ generator1 (Bus 3)  
☐ generator2 (Bus 5)  
☐ DERA (Bus 2)  
☐ hydro (Bus 6)

**Select Events**

Select All Events Clear All Events

☒ event1  
☐ event2  
☐ event3  
☐ event4  
☐ event5  
☐ event6

**Select Dyr Files**

Select All Dyr Clear All Dyr

☒ basecase.dyr

**Execute Operations**

Run Selected Val... Extract Selected ...

**Status & Results**

```

=====
VALIDATION BATCH EXECUTION
=====
Selected generators: 1
Selected events: 1
Selected Dyr files: 1
Total validation runs: 1

[1/1] Starting validation...
Generator: generator1
Event: event1
Dyr: basecase.dyr
Started external process (PID: 26760)

=====
BATCH EXECUTION SUMMARY
=====
Successfully started: 1/1 processes

Note: Each validation runs in its own window.
Check individual windows for detailed progress and results.
=====

```

```

c:\Users\D3X260\AppData\Lo  X + -
CHANNEL  VAR  ICON
9        16  12
Channels added: Flow P and Q (category 16)

Step 9: Running stability simulation...
Starting simulation for 150.0 seconds...
Output file: C:\Users\D3X260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\work\generator1_validation_event1_basecase.out
Simulation completed successfully
Output file created: C:\Users\D3X260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\work\generator1_validation_event1_basecase.out
File size: 586.5 KB
Restored working directory to: C:\Users\D3X260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators

=====
VALIDATION COMPLETED SUCCESSFULLY!
=====
Output file: C:\Users\D3X260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\work\generator1_validation_event1_basecase.out
Output file size: 0.57 MB

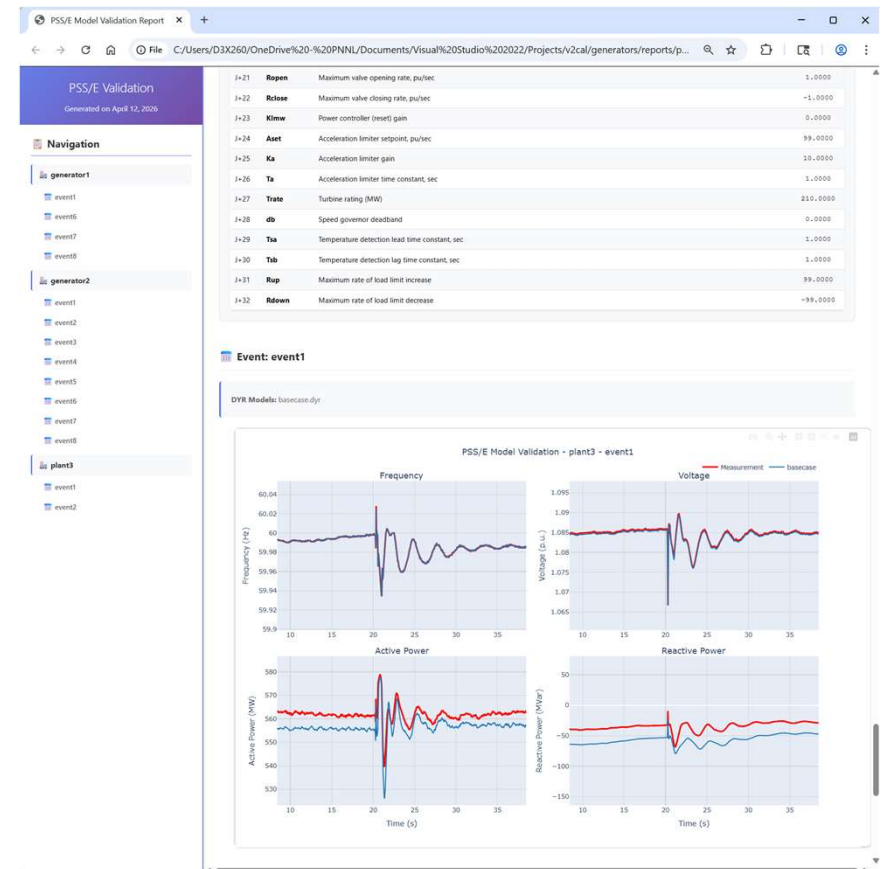
Next steps:
1. Run the CSV export tool to convert .out to .csv
2. Use the plotting tool to compare results

Press Enter to close this window...

```

## Reporting & Visualization

- Generates interactive HTML-based report
- Highly configurable (plots, tables, navigation structure, content selection)
- Supports interactive plots for deeper exploration of system behavior
- Provides tables with model parameters – no need to check dyr files
- Enables easy comparison of model responses — different parameter sets, before/after calibration





# Sensitivity Analysis

- Quantifies which model parameters most strongly influence active or reactive power control behavior
- Helps identify the most impactful parameters for pre-screening prior to calibration
- Supports understanding of model dynamics and control-loop behavior

Parameter: ★ K1 - 1/per unit regulation ▼

**Base Value from DYR:**

Base Value:

---

**Parameter Values Input:**

Input Method:

☐ Range (min/max/points)

☒ Manual values

Values:

Tip: Enter values like: 0.5, 1.0, 1.5, 2.0 or use scientific notation: 1e-3, 5e-3, 1e-2

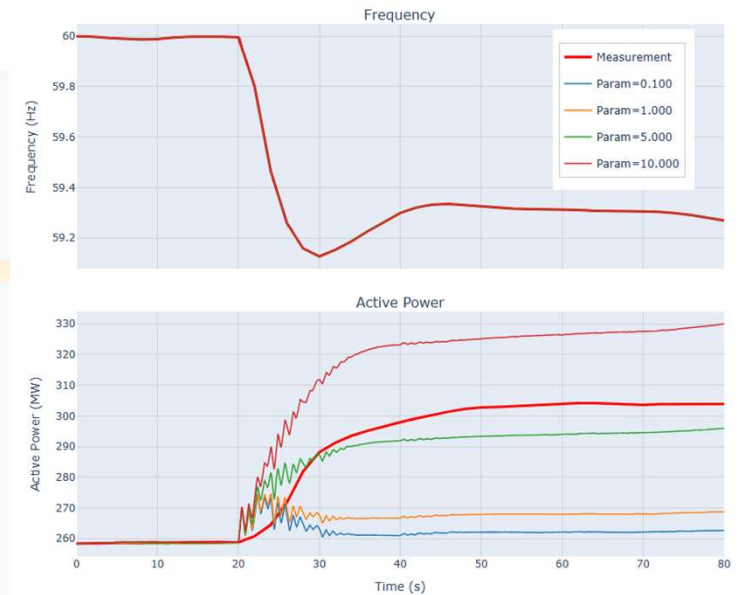
## Analysis Options

Sim Time (s):  ☒ Calculate Error Metrics

## Execute Analysis

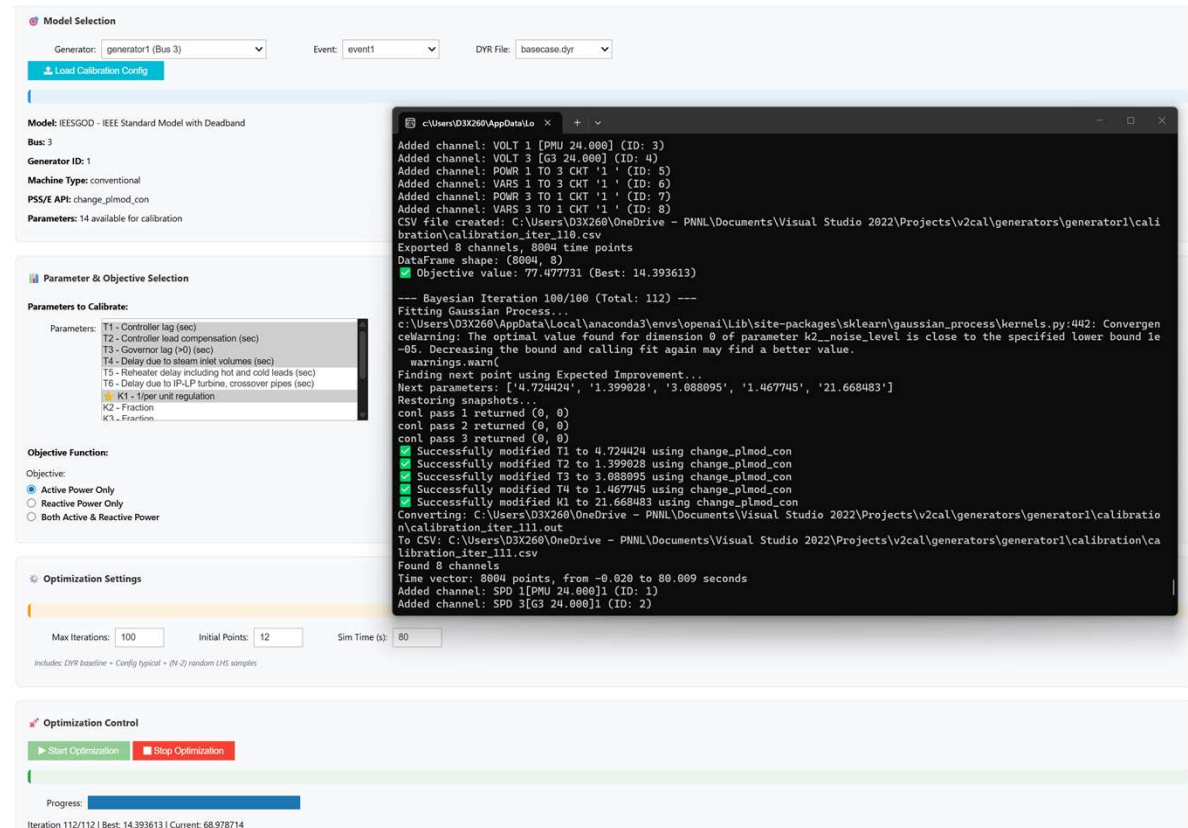
[▶ Run Sensitivity A...](#)

Results will be saved in the generator's sensitivity folder



# Model Calibration

- Bayesian optimization automatically tunes model parameters to minimize mismatch between measured and simulated responses
- User selects which parameters to calibrate
- Flexible objective setup: optimize for active power (P), reactive power (Q), or combined P&Q performance



**Model Selection**

Generator: generator1 (Bus 3) Event: event1 DFR File: basecase.dfr

[Load Calibration Config](#)

Model: IEESGOOD - IEEE Standard Model with Deadband  
Bus: 3  
Generator ID: 1  
Machine Type: conventional  
PSS/E API: change\_plmod\_con  
Parameters: 14 available for calibration

**Parameter & Objective Selection**

Parameters to Calibrate:

- T1 - Controller lag (sec)
- T2 - Controller lead compensation (sec)
- T3 - Governor lag (+0) (sec)
- T4 - Delay due to steam inlet volumes (sec)
- T5 - Reheater delay including hot and cold leads (sec)
- T6 - Delay due to IP-LP turbine crossover pipes (sec)
- K1 - 1-per unit regulation
- K2 - Fraction
- K3 - Fraction

Objective Function:

Objective:

- ☒ Active Power Only
- ☐ Reactive Power Only
- ☐ Both Active & Reactive Power

**Optimization Settings**

Max Iterations: 100 Initial Points: 12 Sim Time (s): 80

Includes: DFR baseline + Config typical + (N-2) random LHS samples

**Optimization Control**

[Start Optimization](#) [Stop Optimization](#)

Progress:

Iteration 112/112 | Best: 14.393613 | Current: 68.978714

```

c:\Users\DJ260\AppData\Local\Microsoft\Windows\Terminal\
Added channel: VOLT 1 [PMU 24.000] (ID: 3)
Added channel: VOLT 3 [G3 24.000] (ID: 4)
Added channel: POWR 1 TO 3 CKT '1 ' (ID: 5)
Added channel: VARS 1 TO 3 CKT '1 ' (ID: 6)
Added channel: POWR 3 TO 1 CKT '1 ' (ID: 7)
Added channel: VARS 3 TO 1 CKT '1 ' (ID: 8)
CSV file created: C:\Users\DJ260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\calibration\calibration_iter_110.csv
Exported 8 channels, 8004 time points
DataFrame shape: (8004, 8)
Objective value: 77.477731 (Best: 14.393613)

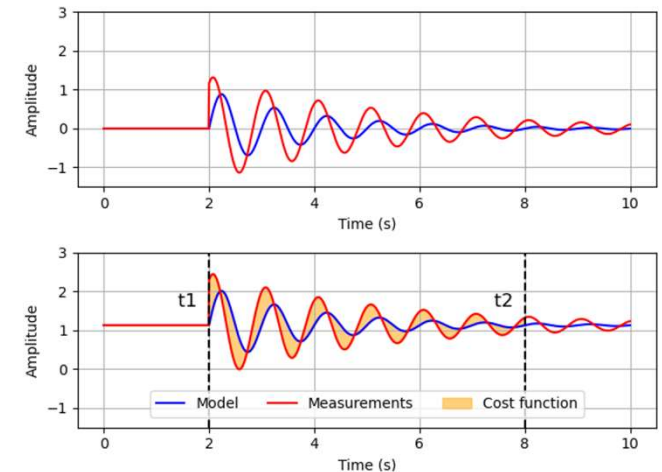
--- Bayesian Iteration 100/100 (Total: 112) ---
Fitting Gaussian Process...
c:\Users\DJ260\AppData\Local\anaconda3\envs\openai\lib\site-packages\sklearn\gaussian_process\kernels.py:442: Convergen
ceWarning: The optimal value found for dimension 0 of parameter k2_noise_level is close to the specified lower bound ie
-85. Decreasing the bound and calling fit again may find a better value.
warnings.warn(
Finding next point using Expected Improvement...
Next parameters: ['4.724424', '1.399028', '3.088095', '1.467745', '21.668483']
Restoring snapshots...
conl pass 1 returned (0, 0)
conl pass 2 returned (0, 0)
conl pass 3 returned (0, 0)
Successfully modified T1 to 4.724424 using change_plmod_con
Successfully modified T2 to 1.399028 using change_plmod_con
Successfully modified T3 to 3.088095 using change_plmod_con
Successfully modified T4 to 1.467745 using change_plmod_con
Successfully modified K1 to 21.668483 using change_plmod_con
Converting: C:\Users\DJ260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\calibratio
n\calibration_iter_111.out
To CSV: C:\Users\DJ260\OneDrive - PNNL\Documents\Visual Studio 2022\Projects\v2cal\generators\generator1\calibration\ca
libration_iter_111.csv
Found 8 channels
Time vector: 8004 points, from -0.020 to 80.009 seconds
Added channel: SPD 1[PMU 24.000]1 (ID: 1)
Added channel: SPD 3[G3 24.000]1 (ID: 2)
  
```

## Error metric

$$err(\mathbf{x})_{t_1}^{t_2} = \left| \int_{t_1}^{t_2} [M_1(\mathbf{x}) - \min(M_1(\mathbf{x})_{t_1}^{t_2}, M_2(\mathbf{x})_{t_1}^{t_2})] dt - \int_{t_1}^{t_2} [M_2(\mathbf{x}) - \min(M_1(\mathbf{x})_{t_1}^{t_2}, M_2(\mathbf{x})_{t_1}^{t_2})] dt \right|,$$

where  $err(x)$  expresses the mismatch between two time series  $M_1$ (observed response) and  $M_2$ (model-based response) in the time range  $(t_1, t_2)$ .

- This metric allows comparing signals with different temporal resolution and captures system response characteristics around a disturbance (e.g. rate of change, over/undershoot, settling values) without focusing solely on the sample-to-sample mismatch.



# Calibration based on Bayesian optimization approach

- **Bayesian Optimization (BO)** is a constrained, derivative-free optimization method.
  - Ideal for cost functions that are computationally expensive or lack analytical forms.
  - Uses a **Gaussian process surrogate model** to approximate the cost function.
  - Strategically selects new evaluation points based on prior observations to efficiently find global optima.
  - Bayesian Optimization package was used: <https://github.com/bayesian-optimization/BayesianOptimization>
- **Parameter Grouping Strategy:**
  - Parameters divided into groups based on their influence on DER behaviors.
    - ✓ Active power-frequency response parameters.
    - ✓ Reactive power-voltage response parameters.
- **Iterative Optimization Process:**
  - Calibrate parameters in one group using Bayesian Optimization, holding other groups constant.
  - Optimized parameters from one iteration inform the next iteration.
  - Process continues iteratively until no significant improvement in cost function is achieved.
- **Benefits of this Method:**
  - Reduces the number of costly simulations.
  - Efficiently identifies optimal parameter sets.
  - Systematically captures interactions between parameter groups for improved model accuracy.

---

## Algorithm 1 Iterative Parameter Calibration

---

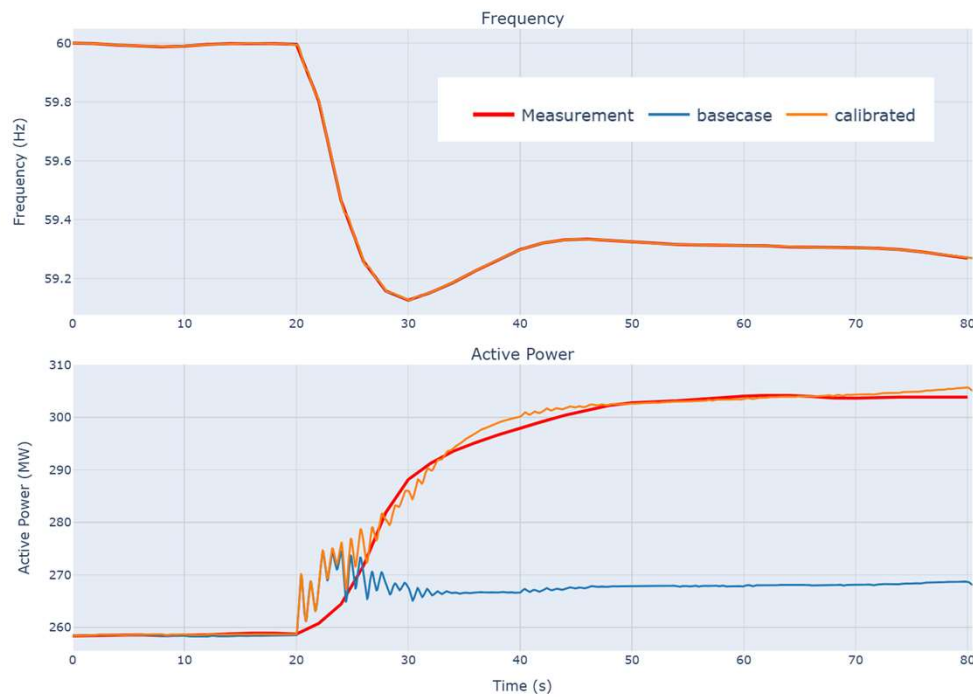
```

1: initialize  $n_g$ : no. of parameter groups,  $n$ : iteration counter,
    $N$ : max. no. of iterations,  $\mathbf{x}_0$ : initial parameter estimates,
    $\sigma$ : tolerance threshold
2:  $n \leftarrow 1$ ,  $\mathbf{x}^* \leftarrow \mathbf{x}_0$ 
3: while  $n \leq N$  do
4:   for  $i = 1$  to  $n_g$  step 1 do
5:     Calibrate parameters in group  $i$  using BO keeping
     all other parameters fixed to obtain parameters  $\mathbf{x}_n$ 
6:   end for
7:   Evaluate cost function  $f(\mathbf{x}_n)$ 
8:   if  $n > 1$  then
9:     if  $f(\mathbf{x}_{n-1}) - f(\mathbf{x}_n) \leq \sigma$  then
10:      break
11:    else  $f^* \leftarrow f(\mathbf{x}_n)$ ,  $\mathbf{x}^* \leftarrow \mathbf{x}_n$ ,  $n \leftarrow n + 1$ 
12:    end if
13:  end if
14: end while
15: return  $\mathbf{x}^*$ 

```

---

# Calibration Results

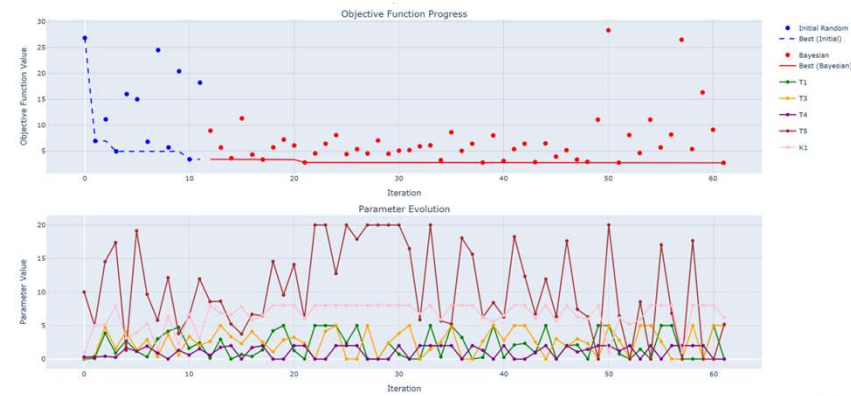


## Speed Governor Model Parameters

Model: IIESGOD - IEEE Standard Model with Deadband

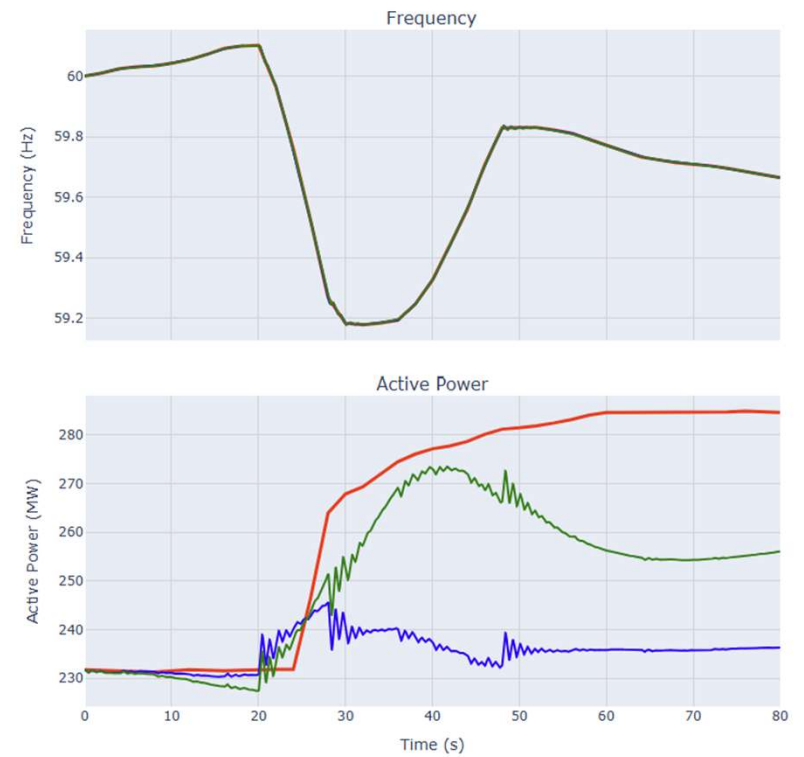
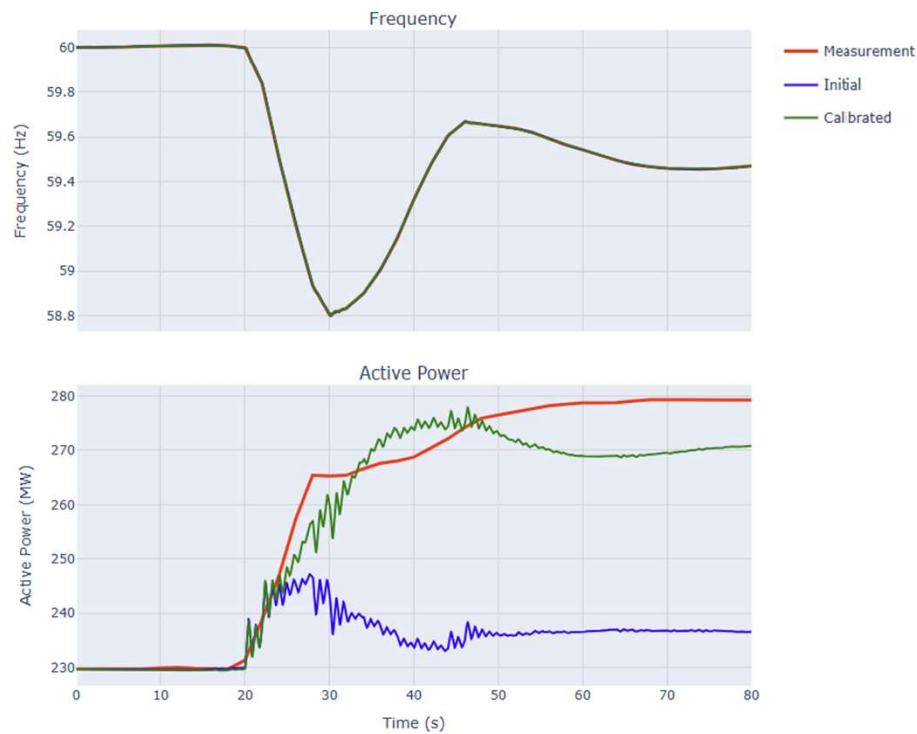
Bus: 3

Index	Parameter	Description	basecase	calibrated
J	T1	Controller lag (sec)	0.0000	1.0000
J+1	T2	Controller lead compensation (sec)	0.0000	0.0000
J+2	T3	Governor lag (>0) (sec)	0.1000	1.0000
J+3	T4	Delay due to steam inlet volumes (sec)	0.3000	1.0000
J+4	T5	Reheater delay including hot and cold leads (sec)	10.0000	10.0000
J+5	T6	Delay due to IP-LP turbine, crossover pipes (sec)	0.4000	1.0000
J+6	K1	1/per unit regulation	1.000000	6.500000
J+7	K2	Fraction	0.7140	0.7140
J+8	K3	Fraction	0.6000	0.6000
J+9	PMAX	Upper power limit	1.0000	1.0000
J+10	PMIN	Lower power limit	0.0000	0.0000
J+11	DBH	Deadband for overspeed, p.u.	0.0000	0.0000
J+12	DBL	Deadband for underspeed, p.u.	0.0000	0.0000
J+13	Trate	Turbine rating, if zero, then MBASE used	0.0000	0.0000





# Calibration results validation



## Future Work / Work in Progress

- Enable V<sup>2</sup>Cal for agentic workflows
  - Expose V<sup>2</sup>Cal capabilities through an MCP-compatible interface
  - Make the tool agent-friendly for automated orchestration
  - Support natural-language prompts for
    - ✓ Understanding model structure and parameters
    - ✓ Generating reports
    - ✓ Executing commands and analysis steps
- Extended simulation capabilities
  - Add support for EMT-level simulation tools (e.g., PSCAD)
  - Perform EMT-based model validation
  - Benchmark and compare RMS vs. EMT model behavior for consistency and fidelity

## Conclusion

- V<sup>2</sup>Cal streamlines end-to-end measurement-based model validation through a fully automated workflow
- Integrates data ingestion, play-in case construction, simulation, sensitivity analysis, and parameter calibration
- Bayesian optimization and targeted parameter selection improve model fidelity while minimizing analyst effort
- Interactive HTML reports enhance transparency, traceability, and collaboration
- Overall, V<sup>2</sup>Cal accelerates verification workflows and supports more reliable active and reactive power control modeling



**Thank you**

