Recent Progress:

1. Global g, linear & non-linear functionality for:
   - pwrmod - using gen bus
2. Explored ODE45 viability.
3. Found note about PSS gain
4. Wrote up changes to pstSETO Version.
5. Fixed default machine model in 2.3 and reran mini WECC version comparisons
6. created MATLAB plot functions to compare PST data
7. GitHub updated: `https://github.com/thadhaines/MT-Tech-SETO`

Sandia Action Items:

- Continue development of current injection and voltage injection converter models and their implementation in PST.
- Decide on PST base version
- Plan for variable time step methods
- Investigate power electronics-based models

Current Tasks:

1. Continue to cast globals to structured g
2. Get code/cases from Ryan
3. Test models in both non-linear and linear simulations
4. Think about using standard ODE solvers
5. Investigate PSS differences between v2.3 and 3.1
6. think about cleaning up or flowcharting svm_mgen_Batch
7. Explore/Create v3 example cases
8. Think about AGC implementation.
9. Work on understanding PST operation
10. Document findings of PST functionality
11. Investigate Octave compatibility

Possible Future Tasks:

1. Investigate Sandia integrator stability methods. See if the modified PST used by Sandia in 2015 paper exists for an example of how they implemented different integration routines / stability calculations. (Contact Ryan?)

Coding Thoughts:

1. Condense ≈340 globals into 1 structured array with ≈18 fields based on category that contain PST arrays used for vector calculations. ex: `g.lmod.lmod_con`, `g.sys.t`
2. Enable 'objects' (structure of arrays), but include functions to interact with condensed globals so vectorized operations are still possible.
   This requires more conceptual modeling to understand what needs to be passed/references/changed for each 'object'. Would enable addition of area definitions to models.
3. Separate total system calculation of derivatives into scripts/functions to allow for easier changing of integration method. Possibly employ `feval` for a more dynamic calculation routine.
4. Rework how switching & perturbance events are handled into a more flexible and general format. (use flags)
5. Generate something similar to unit test cases to verify code changes don't break everything during refactor.
6. Generate comparison scripts to verify simulated results match after code changes.

Current Questions:

1. Requirements for variable step methods
2. PQ bus for pwrmod not working?
3. PST modeling of transformers?