

**Recent Progress:**

1. global `g` added for `lmod` and `tg` models.
2. PST flowchart created
3. ML vs OCTAVE speed test
4. 2.x  $\rightarrow$  3.x Models assigned to Dan
5. AGC lit review sent to Dave
6. GitHub updated:  
<https://github.com/thadhaines/MT-Tech-SET0>

**Sandia Action Items:**

- Review lit items related to:
  - multi rate methods
  - variable time step methods
  - power electronics-based models
- Work on converter-based resource models (pwermod, ivmmmod)

**Current Tasks:**

1. Explore/Create v3 example cases
2. Cast individual globals to single structured global
3. Reference Trudnowski code for ‘structured array, functionalized’ approach.
4. Reference Stajcar code for basic transient stability simulation flow.
5. Work on understanding PST operation
6. Document findings of PST functionality
7. Investigate Octave compatibility
8. Continue to work on data plots?

**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  $\approx 340$  globals into 1 structured array with  $\approx 18$  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 & perturbation 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
 

• system model	• Exciter models
• Load flow solver	• converter models
• Network solver	• load modulation models
• Machine models	
• Governor model	• ...
2. PST modeling of transformers?
3. PST modeling of faults:  
Uses alternate Y matrices?  
Creates fault bus?