# Notes on modeling solar systems (or any power injection):

* Nc\_load.m is a function where the loads are formed (all ZIP). This is where I add in the modulation. Eg, lmod is added on line 120 via lmod\_st = states of lmod. This is where the power flow eqns are conducted.
* lmod\_st are added as constant impedance. That is, load modulation is actually modeled as impedance modulation.
* To add a modulation as a const P, look at the variable curr\_load. This is the current load at a given time step (actual S power).
* Plan
  + Add a control matrix to the data file called pwrmod\_con which tells which buses will have a power modulation and all the settings.
  + Build a function called pwrmod\_p.m that has the state eqns for modulating real power. Variables:
    - pwrmod\_p\_sig = commanded modulated real powers. pwrmod\_p\_sig(i,k) corresponds to row i of pwrmod\_con at time t(k).
    - pwrmod\_p\_st = modulated real powers to be added to curr\_load in nc\_load.m. pwrmod\_p\_st (i,k) corresponds to row i of pwrmod\_con at time t(k)
    - dpwrmod\_p\_st = derivative of pwrmod\_p\_st.
  + Build a function called pwrmod\_q.m that has the state eqns for modulating reac power. Variables:
    - pwrmod\_q\_sig = commanded modulated reac powers. pwrmod\_q\_sig(i,k) corresponds to row i of pwrmod\_con at time t(k).
    - pwrmod\_q\_st = modulated reac powers to be added to curr\_load in nc\_load.m. pwrmod\_q\_st (i,k) corresponds to row i of pwrmod\_con at time t(k)
    - dpwrmod\_q\_st = derivative of pwrmod\_q\_st.
  + Build a function called pwrmod\_dyn.m to implement the equations that construct pwrmod\_p\_sig and pwrmod\_q\_sig at time t(k). This can include dynamics.
    - P\_states{i} = real-power modulation states for bus pwrmod\_con(i,:).
    - dP\_states = derivative of P\_states
    - P = commanded real-power modulation = pwrmod\_p\_sig(:,k).
    - Q\_states{i} = reac-power modulation states for bus pwrmod\_con(i,:).
    - dQ\_states = derivative of Q\_states
    - Q = commanded reac-power modulation = pwrmod\_q\_sig(:,k)
    - pwrmod\_data = general variable for storing user data.

# Code Changes:

* Created control matrix for the data file:
  + pwrmod\_con = n\_pwrmod x 7 matrix of power modulation information. Row k is for the kth injection bus and is
    - [nBus, Tp, Pmax, Pmin, Tq, Qmax, Qmin]
      * nBus = injection bus number
      * Tp = time constant for real-power injection
      * Pmax = max real-power (or current) modulation on system base
      * Pmin = min real-power (or current) modulation on system base
      * Tq = time constant for reac-power injection
      * Qmax = max reac-power (or current) modulation on system base
      * Qmin = min reac-power (or current) modulation on system base
    - Note, this bus MUST be declared as a constant-power P/Q or constant-current P/Q load bus in the load\_con.
* Modified s\_simu\_Batch.m
  + Added adjustment to bus min/max Q to match pwrmod\_con min/max Q
  + Added the call to pwrmod\_indx near line 129
  + Initialize pwrmod variables near line 324
  + Initialize power modulation controls near line 481
  + Added derivative calculations for predictor step near line 743
  + Added predictor steps near line 825
  + Added derivative calculations for corrector step near line 1015
  + Added corrector steps near line 1102
* Modified pst\_var.m to set up the following as global
  + pwrmod\_con
  + n\_pwrmod = number of rows of pwrmod\_con.
  + pwrmod\_idx = indexing variable. pwrmod\_idx(k) = row of load\_con corresponding to pwrmod\_con(k,1).
  + pwrmod\_p\_st = n\_pwrmodx1 vector of modulated real powers as defined above.
  + dpwrmod\_p\_st = derivative of pwrmod\_p\_st.
  + pwrmod\_q = n\_pwrmodx1 vector of modulated reac powers as defined above.
  + dpwrmod\_q\_st = derivative of pwrmod\_q\_st
  + pwrmod\_p\_sig = n\_pwrmodx1 vector of commanded modulated real powers as defined above
  + pwrmod\_q\_sig = n\_pwrmodx1 vector of commanded modulated reac powers as defined above
  + pwrmod\_data = general variable for storage by the user of pwrmod\_dyn.m
* Created new function pwrmod\_indx.m that
  + sets up n\_pwrmod and pwrmod\_idx. Set up the similar to lmod variables.
* Create new functions pwrmod\_p.m and pwrmod\_q.m
  + Initializes, network interfaces, or derivative calculations.
* Create new function pwrmod\_dyn.m
  + Equations to build pwrmod\_p\_sig and pwrmod\_q\_sig
* Modified nc\_load.m
  + Modulated P and Q loads near line 134
  + Changed all load\_pot(\*,1) to load\_pot1 after 134 to implement the modulation.
* In red\_ybus.m
  + Adjusted bus\_sol to account that pwrmod bus is a PV bus near line 108.
* In svm\_mgen.m
  + Identify n\_pwrmod and pwrmod\_idx near line 273
  + Add n\_pwrmod to state and max\_state lines near 360
  + Initialized pwrmod\_p\_sig and pwrmod\_q\_sig near line 414
  + Initialize states and state rates near line 503
  + Initialize pwrmod states near line 615
  + Set states near line 701
  + Set sigs near line 724
  + Build b matrices near line 804
* In p\_file.m (for svm\_mgen.m)
  + Added state calculations near line 109
  + dvector addition near line 193
  + reset pwrmod variables near line 461
  + build b\_pwrmod\_p matrix near line 285
* Created function pwrm\_indx.m (for svm\_mgen.m). same as pwrmod\_indx.m
  + sets up n\_pwrmod and pwrmod\_idx. Set up the similar to lmod variables
* In ns\_file.m (svm\_mgen.m)
  + added pwrmod states near line 195
* In p\_m\_file.m (svm\_mgen.m)
  + Added p\_mat near line 271
* In p\_cont.m
  + Add pertabations near line 296