Intro

The purpose of this document is to record changes of note made to PST over the course of the SETO work that may be worth not forgetting about. It should be noted that PST SETO is based on PST version 3.

PSS model

There was a correction to the washout time constant in the PSS model between PST version 2.x and 3. To accommodate for this, the SETO version has two pss files named `pss2` and `pss3` which mimic the computation of each PST version respectively. The idea is to enable a user to speify which model the pss settings use in a particular case. The current usage is similar to:

```
copyfile([PSTpath 'pss2.m'],[PSTpath 'pss.m']); % use version 2 model of PSS
```

Alternatively, a `pssGainFix` variable may be set to 1, or true, which will adjust the version 2 data from a `d_` file to work the same way with the version 3 model. This is accomplished by executing: `pss_con(:,3) = pss_con(:,3)./pss_con(:,4);` While this works, it's kind of confusing and may be removed.

Sub-transient Machine model

There are two versions of the `mac_sub` model available. The `_ORIG` model is the standard PST model based on the R. P. Schulz, "Synchronous machine modeling" algorithm. The `_NEW` model is based on the PSLF 'genrou' model by John Undrill. Either may be copied to the `mac_sub` file for use.

exc_dc12

In 2015 there were 'errors' corrected in the saturation block that create differences between version 2 and 3 of this model. Effects are noticeable, but a solution hasn't been investigated yet.

exc_st3

Corrected theta index to `n_bus` from `n` per Ryan Elliott. Corrected simple `*` to `.*` int the `if ~isempty(nst3_sub)` section.

mac_tra

Commented out code that prevented the setting equal of the transient reactances.

pwrmod

This is the power or current injection model Dan created for version 2.3. It's meant to model the 'grid following' type of converters. It is included in both the non-linear and linear simulation modes of PST SETO.

ivmmod

This is the voltage behind an impedance model Dan created. It's meant to model a 'grid forming' converter where voltage and angle are manipulatable. While there are questions about the reality of such operations, the model exists and appears to work in the non-linear simulation of PST SETO.

Global Variables

To enable easier manipulation of PST - it was decided to create a global structure that contains all system globals. While this may or may not have been a good idea - it happened. Initial results show a speed up of over 2 times. In other words, it could be assumed previous versions of PST spend half of their computation time loading globals…

Inside the global variable `g` are fields that corresponds to models, or groups, of other globals. As of this writing, compiled on July 9, 2020, the following subparagraphs describe the globals contained in each field.

lmod

```
global lmod_con % defined by user
global n_lmod lmod_idx % initialized and created in lm_indx
global lmod_sig lmod_st dlmod_st % initialized in s_simu
global lmod_pot  % created/initialized in lmod.m
global lmod_data % added by Trudnowski - doesn't appear to be used?
```

tg

```
%% turbine-governor variables
global  tg_con tg_pot
global  tg1 tg2 tg3 tg4 tg5 dtg1 dtg2 dtg3 dtg4 dtg5
global  tg_idx  n_tg tg_sig tgh_idx n_tgh
```

It should be noted that the hydro governor model `tgh` has not been modified as no examples seemed to use it.

rlmod

```
global  rlmod_con n_rlmod rlmod_idx
global  rlmod_pot rlmod_st drlmod_st
global  rlmod_sig
```

exc

```
global  exc_con exc_pot n_exc
global  Efd V_R V_A V_As R_f V_FB V_TR V_B
global  dEfd dV_R dV_As dR_f dV_TR
global  exc_sig % pm_sig n_pm % not related to exciters?
global smp_idx n_smp dc_idx n_dc  dc2_idx n_dc2 st3_idx n_st3;
global smppi_idx n_smppi smppi_TR smppi_TR_idx smppi_no_TR_idx ;
global smp_TA smp_TA_idx smp_noTA_idx smp_TB smp_TB_idx smp_noTB_idx;
```

```
global smp_TR smp_TR_idx smp_no_TR_idx ;
global dc_TA dc_TA_idx dc_noTR_idx dc_TB dc_TB_idx dc_noTB_idx;
global dc_TE  dc_TE_idx dc_noTE_idx;
global dc_TF dc_TF_idx dc_TR dc_TR_idx
global st3_TA st3_TA_idx st3_noTA_idx st3_TB st3_TB_idx st3_noTB_idx;
global st3_TR st3_TR_idx st3_noTR_idx;
```

mac

```
global  mac_con mac_pot mac_int ibus_con
global  mac_ang mac_spd eqprime edprime psikd psikq
global  curd curq curdg curqg fldcur
global  psidpp psiqpp vex eterm theta ed eq
global  pmech pelect qelect
global  dmac_ang dmac_spd deqprime dedprime dpsikd dpsikq
global  n_mac n_em n_tra n_sub n_ib
global  mac_em_idx mac_tra_idx mac_sub_idx mac_ib_idx not_ib_idx
global  mac_ib_em mac_ib_tra mac_ib_sub n_ib_em n_ib_tra n_ib_sub
global pm_sig n_pm
```

pss

```
global  pss_con pss_pot pss_mb_idx pss_exc_idx
global  pss1 pss2 pss3 dpss1 dpss2 dpss3 pss_out
global  pss_idx n_pss pss_sp_idx pss_p_idx;
global  pss_T  pss_T2 pss_T4 pss_T4_idx
global pss_noT4_idx % misspelled in pss_indx as pss_noT4
```

Despite the renaming of the `pss_noT4_idx`, it doesn't seem to actually be used anywhere.

pwr

```
global  pwrmod_con n_pwrmod pwrmod_idx
global  pwrmod_p_st dpwrmod_p_st
global  pwrmod_q_st dpwrmod_q_st
global  pwrmod_p_sig pwrmod_q_sig
global  pwrmod_data
```

There are some cells that contain user defined derivatives that aren't included yet.

ncl

```
global  load_con load_pot nload
```

sys

```
global  basmva basrad syn_ref mach_ref sys_freq
global  bus_v bus_ang psi_re psi_im cur_re cur_im bus_int
global  lmon_con % lmon_con not used in non-linear sim...
global theta % moved from g.mac as it is a system variable
```

Despite the user manual listing `cur_re`,`cur_im`,`psi_re`, and `psi_im` as system variables, they appear to only correspond to machines.

svc

```
global  svc_con n_svc svc_idx svc_pot svcll_idx
global  svc_sig
% svc user defined damping controls
global n_dcud dcud_idx svc_dsig
global svc_dc % user damping controls?
global dxsvc_dc xsvc_dc
%states
global B_cv B_con
%dstates
global dB_cv dB_con
```

There seems to be some code related to user defined damping control of SVC, but it is not it the user manual.