Power System Toolbox
Version 3.0

Graham Rogers
RR#5 Colborne
Ontario K0K 1S0
Canada

phone & fax: (905)349-2485
email: cherry@eagle.ca

Table of Contents

1 Load Flow

1.1 Introduction

In power systems, a load flow study is performed to obtain a set of feasible steady state system conditions which obey certain system constraints. It requires that the system structure is specified together with the generators' real powers and the system's active and reactive power loads. System bus voltage magnitudes and angles are then calculated by solving the nonlinear algebraic network equations so that the specified loads are supplied.
Although load flow studies are important in their own right, they are also required to act as starting points for dynamic simulation.

1.2 Data Requirements

The system structure is specified, in PST, by two matrices, **bus** and **line**. The format for these two specification matrices is given in **Function: loadflow**. The example given in that function description is used as a basis for this tutorial.

1.3 Load Flow Example Data

The bus and line data of a 4 generator, 2 area system [1] are

```
bus = […
1   1.03     18.5   7.00  1.61  0.00  0.00  0.00  0.00  1  99.0  -99.0  22.0  1.1  .9;
2   1.01     8.80   7.00  1.76  0.00  0.00  0.00  0.00  2  5.0   -2.0   22.0  1.1  .9;
3   0.9781  -6.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5;
4   0.95    -10     0.00  0.00  9.76  1.00  0.00  0.00  3  0.0   0.0   115.0 1.05  .95;
10  1.0103  12.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5;
11  1.03    -6.8    7.16  1.49  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0  1.1  .9;
12  1.01    -16.9   7.00  1.39  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0  1.1  .9;
13  0.9899  -31.8   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5;
14  0.95    -38     0.00  0.00  17.67 1.00  0.00  0.00  3  0.0   0.0   115.0 1.05  .95;
20  0.9876   2.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5;
101 1.05    -19.3   0.00  8.00  0.00  0.00  0.00  0.00  2  99.0  -99.0 230.0  1.5  .5;
110 1.0125  -13.4   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5;
120 0.9938  -23.6   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0  1.5  .5 ];

line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.05;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.005    0.00    1.0  0. 1.2 0.8 0.05;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];
```

The single line diagram of the test system is shown in Fig. 1. The system consists of two identical areas interconnected by two long transmission lines. In each area, there are two generators, at buses 1 and 2 in area 1, and at buses 11 and 12 in area 2. The loads are at bus 4 in area 1, and at bus 14 in area 2. Bus 1 acts as the swing bus. Bus 101 is considered to be a generator in the load flow. It has zero real power generation and acts as a reactive power source to hold the voltage at the center of the interconnecting transmission lines. When we come to do dynamic simulations, this bus will be the site of a static VAR compensator, and the reactive generation will give the initial susceptance of the SVC.

There are step down under-load tap changing transformers between bus 3 and bus 4, and bus 13 and bus 14. The tap settings are changed during a load flow solution so that the load bus voltages are maintained between the limits set in columns 14 and 15 of the **bus** matrix .

The generators at buses 2, 11, and 12 have reactive power limits set to -2pu to 5pu. The swing bus generator and the reactive power source at bus 101 has limits -99pu to 99pu.

The rated voltage (kV) for each bus is specified in column 13 of **bus**. This is not used in an ac power flow, but we will see later, that in a dc power flow the information is necessary, since the dc system is modelled in natural units rather than in per unit.

1.4 Load Flow

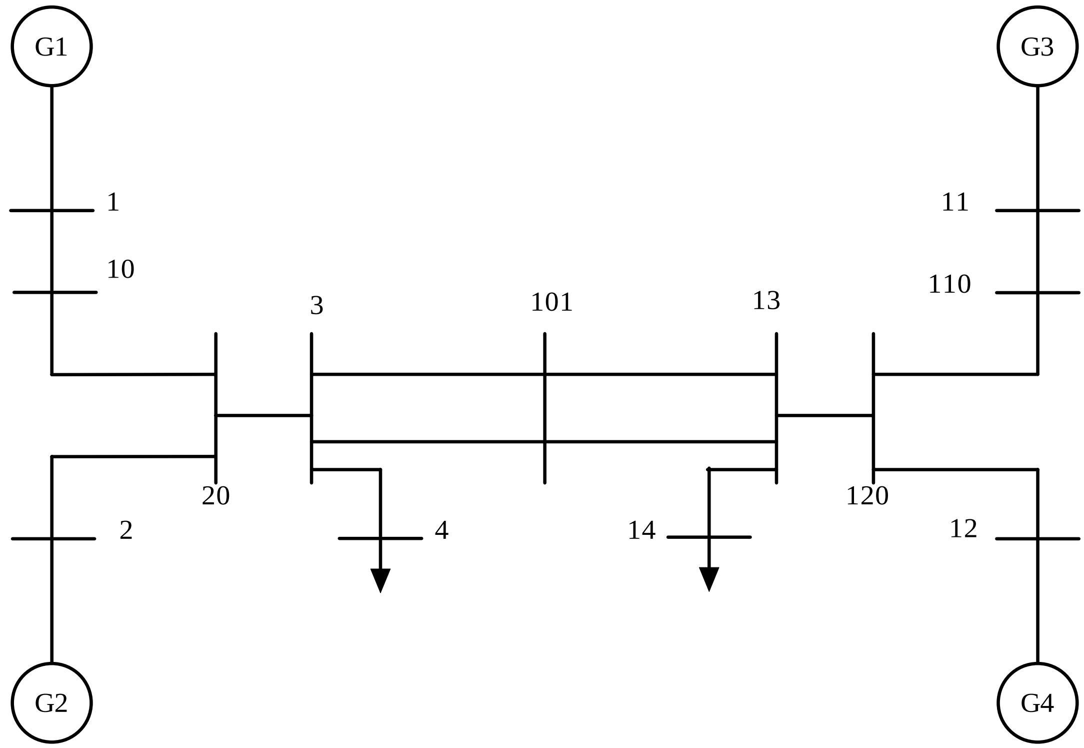


Figure 1 Single Line Diagram Two Area System

The script file **lfdemo** is an ac load flow driver. When **lfdemo** is typed at the MATLAB command , you are asked to choose a data file which contains the bus and line load flow specification files. In our example case, these are specified in **data2a.m**. If your choice of file contains valid load flow data, you will be asked whether you wish to have a load flow report. Entering **'y'** opens a diary file in the current MATLAB directory with the name **lf_report.txt**. type **'n'** or press **enter** if you do not want a report. As the solution progresses (a Newton_Raphson algorithm performed by **loadflow**) the voltages at the load buses are found to be out-of-limits. The corresponding transformer taps are adjusted to bring the load voltage back in range. At the end of the solution, either the solution has converged, or the number of allowed iterations has been exceeded. In either case, the user is given a list of solution viewing options.

For the example case, the solution progress is as follows:

```
lfdemo
loadflow demo program
0.5 constant current load, svc at bus 101
Do you need a load-flow solution report?  [y/n]n >>
inner ac load flow failed to converge after 10 iterations
at tap iteration number 1
voltage low changing tap on line
     3
taps reset to
tap =
         0.95
voltage low changing tap on line
    10
taps reset to
tap =
         0.95
inner ac load flow failed to converge after 10 iterations
at tap iteration number 2
voltage low changing tap on line
    10
taps reset to
tap =
          0.9
inner load flow iterations
     6
 tap iterations
     3
Elapsed time is 0.190000 seconds.
You can examine the system data
Type 1 to see initial bus data
     2 to see modified line data
     3 to see solved load flow bus solution
     4 to see line flow
     5 to see bus voltage magnitude profile
     6 to see bus voltage phase profile
     0 to quit
enter selection >> 3
                                    Solved Bus Data
                                      GENERATION             LOAD
      BUS     VOLTS     ANGLE      REAL  REACTIVE      REAL  REACTIVE
      1         1.03        18.5       7.2138       2.0926           0           0
      2         1.01      8.1584            7       2.8023           0           0
      3      0.94845     -7.3757            0            0  1.0293e-014  5.9826e-014
      4      0.99212       -10.2            0            0         9.76           1
      10       1.0029      11.803            0            0 -6.5395e-016 -1.3665e-016
      11         1.03     -6.9696         7.16       2.5494            0            0
      12         1.01     -17.315            7       3.9297            0            0
      13      0.91512     -33.347            0            0 -1.2581e-014 -7.2892e-015
      14       1.0081     -38.292            0            0        17.67            1
      20      0.97059      1.3096            0            0  8.6183e-016  1.9702e-017
      101        1.05     -21.022 -5.5511e-015       5.0029            0            0
      110     0.99546     -13.667            0            0 -5.1076e-015 -6.0373e-015
      120     0.95208     -24.298            0            0  4.8129e-014  7.9611e-015
paused: press any key to continue
Type 1 to see initial bus data
     2 to see modified line data
     3 to see solved load flow bus solution
     4 to see line flow
     5 to see bus voltage magnitude profile
     6 to see bus voltage phase profile
     0 to quit
enter selection >> 5
```

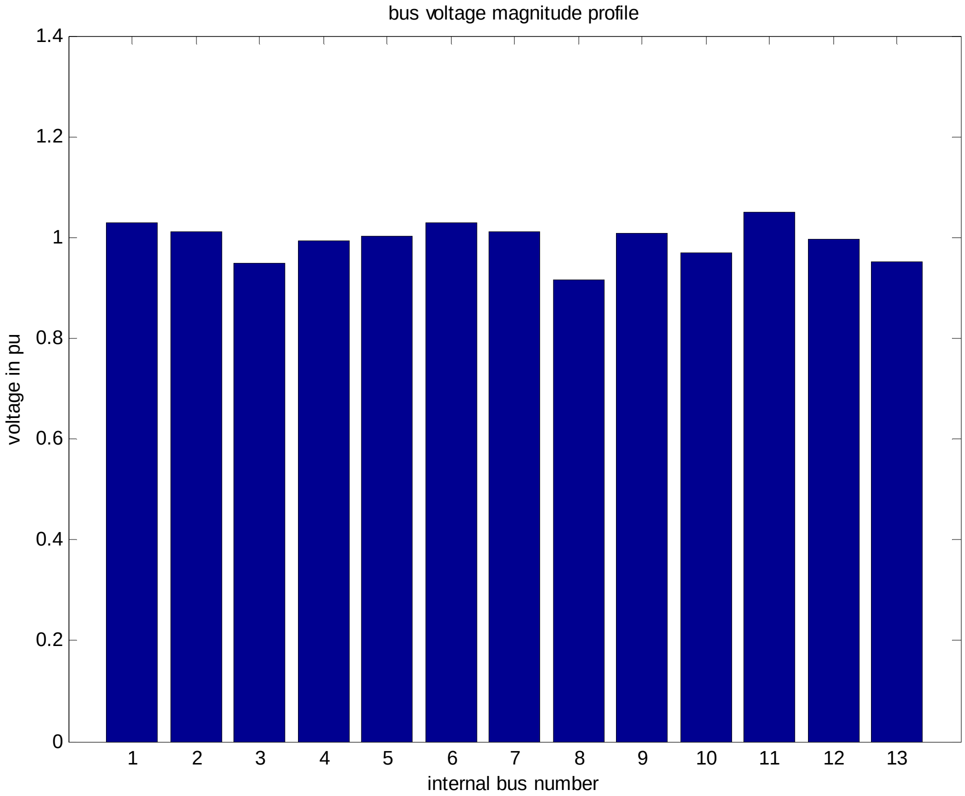


Figure 2 Bar chart of bus voltage magnitude

1.5 Voltage Stability

The script file **vsdemo** is a driver for steady state voltage stability analysis. The ac load flow program **loadflow** is used in this demo, so as it stands it cannot be used to examine voltage stability in systems having HVDC links.
The demonstration allows the total active and reactive power loads to be increased in steps by a ratio of the original bus loads. A load flow is performed at each step, and if required the inverse eigenvalues of the load flow Jacobian ($\frac{\partial Q}{\partial V}$)can be found. The maximum eigenvalue and the maximum element of the corresponding eigenvector are displayed. The critical eigenvalue may be plotted if desired.
Normally, the as the load increases, the load flow will take longer to converge. Close to voltage stability, it will likely fail to converge. Consequently, the user is given the option of starting the next load flow from the previous load flow solution, or from the original load flow data.
On output, a history of the loads is contained in **load_p** and **load_q**, and that of the system voltages in **v_mag**. These may be plotted to show the system **V/P** characteristics.

1.5.1 Voltage Stability Example

The following data represents a 3 generator, 9 bus system

```
bus = [   1 1.04    0.00   0.00   0.00  0.00  0.00  0.00  0.00 1;
         2 1.02533 0.00   1.63   0.00  0.00  0.00  0.00  0.00 2;
         3 1.02536 0.00   0.85   0.00  0.00  0.00  0.00  0.00 2;
         4 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
         5 1.00    0.00   0.00   0.00  0.90  0.30  0.00  0.00 3;
         6 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
```

```
          7 1.00    0.00   0.00   0.00  1.00  0.35  0.00  0.00 3;
          8 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
          9 1.00    0.00   0.00   0.00  1.25  0.50  0.00  0.00 3];

line = [  1 4 0.0    0.0576 0.     1. 0. ;
          4 5 0.017  0.092  0.158  1. 0. ;
          5 6 0.039  0.17   0.358  1. 0. ;
          3 6 0.0    0.0586 0.     1. 0. ;
          6 7 0.0119 0.1008 0.209  1. 0. ;
          7 8 0.0085 0.072  0.149  1. 0. ;
          8 2 0.0    0.0625 0.     1. 0. ;
          8 9 0.032  0.161  0.306  1. 0. ;
          9 4 0.01   0.085  0.176  1. 0. ];
```

The following results were obtained using vsdemo:
With a load increase of 2.05 times the statring load
The dominant eigenvalue `0.33375`
The maximum eigenvector entry is `0.63776`
The corresponding bus number is `9`

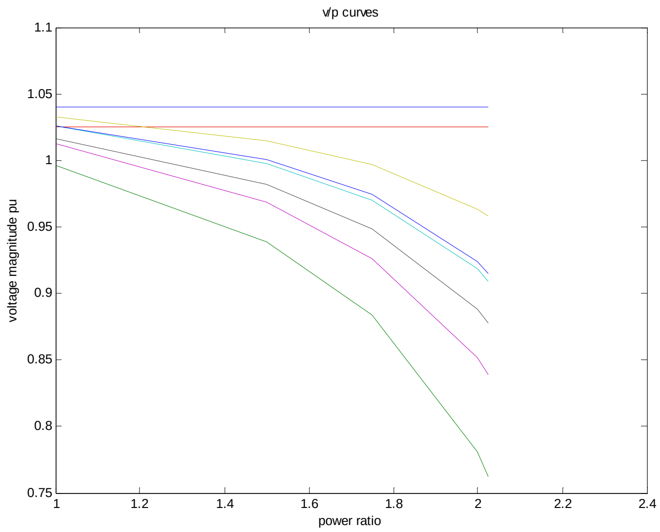


Figure 3 Voltage magnitudes as load increases

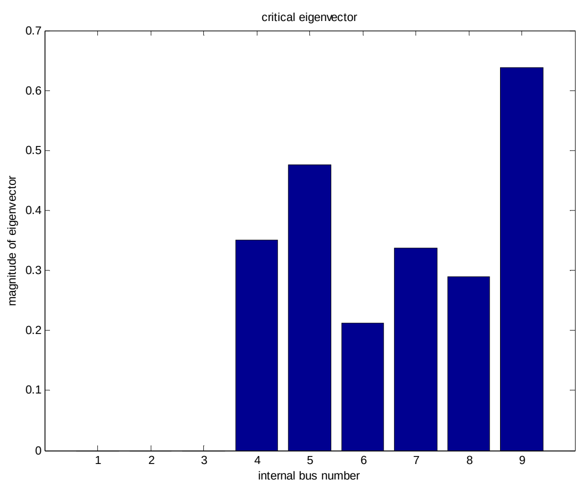


Figure 4 Critical Eigenvector

1.6 HVDC

The script file **lfdc** is a load flow driver for systems having HVDC links. In addition to ac load flow data, dc data must be supplied in the form of the dc converter specification matrix (**dcsp_con**) and the dc line specification matrix (**dcl_con**). A complete set of data (**dtestdc.m)** for the two area system having an HVDC link between ac bus 5 and ac bus 15 is

```
bus = [ …
1   1.03 18.5  7.00 1.61   0.00  0.00  0.00  0.00  1  99.0 -99.0   22.0  1.1  .9;
2   1.01 8.80  7.00 5.76   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1  .9;
3   1.0  -6.1  0.00 0.00   0.00  0.00  0.00  6.00  2   0.0   0.0  500.0  1.5  .5;
4   0.95 -10   0.00 0.00   9.76  1.00  0.00  0.00  3   0.0   0.0  115.0  1.05 .95;
5   1.0  -10   0.00 0.00   10.7  2.8   0.00  0.00  3  99.0 -99.0  115.0  1.2  .8;
10  1.01 12.1  0.00 0.00   0.00  0.00  0.00  0.00  3   0.0   0.0  230.0  1.5  .5;
11  1.03 -6.8  7.16 1.49   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1  .9;
12  1.01 -16.9 7.00 1.39   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1  .9;
13  0.99 -31.8 0.00 0.00   0.00  0.00  0.00  0.00  2   0.0   0.0  500.0  1.5  .5;
14  0.95 -38   0.00 0.00   17.7  1.00  0.00  0.00  3   0.0   0.0  115.0  1.05 .95;
15  1.0  -14   0.00 0.00  -10.4  2.7   0.00  6.00  3  99.0  -99.0 115.0  1.2  .8;
20  0.99  2.1  0.00 0.00   0.00  0.00  0.00  0.00  3   0.0   0.0  230.0  1.5  .5;
101 1.05 -19.3 0.00 2.00   0.00  0.00  0.00  0.00  3  99.0  -99.0  500.0  1.5  .5;
110 1.01 -13.4 0.00 0.00   0.00  0.00  0.00  0.00  3   0.0    0.0  230.0  1.5  .5;
120 0.99 -23.6 0.00 0.00   0.00  0.00  0.00  0.00  3   0.0    0.0  230.0  1.5  .5 ];
```

```
line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005    0.00    1.0  0. 1.5 0.5 0.05;
3    5  0.0     0.007    0.00    1.0 0. 2.0 0.5 0.005;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.007    0.00    1.0  0. 1.5 0.5 0.05;
13   15 0.0     0.01     0.00    1.0 0. 2.0 0.5 0.005;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];
```

```
dcsp_con = [...
1  5  1  500  6    4   5   30;
2  15 2  500  6    4  18   25];
```

```
dcl_con = [...
1  2  20  0  0  0  0  1000  15];
```

The ac buses 5 and 15 are the LT buses of the HVDC converter transformers. The corresponding HT buses are buses 3 and 13 respectively.

The dc load flow is performed by a sequence of ac load flows, followed by dc load flows which reset the dc controls and the loads at the LT buses. When both the ac and dc load flows have converged, the overall HVDC load flow is taken to be converged.

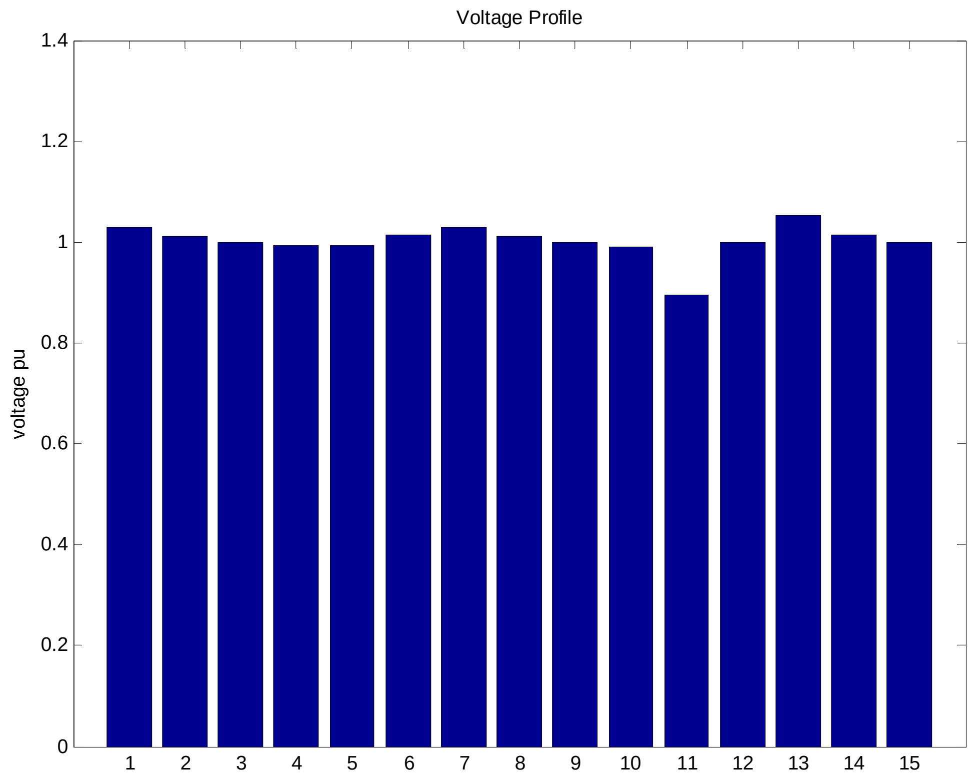


Figure 5 Voltage Profile of Solved Loadflow

The converter parameters are:
Rectifier
alpha in deg
27.373
dc voltage in kV
524.2666
Power in MW
104.8533
Inverter
gamma in degrees
18
dc voltage in kV
504.2666
power in MW
100.8533
line current in kA
0.2Rectifier

It should be noted that the LT bus voltage magnitudes are low. This is the result of the dc load flow solution adjusting the taps to get the required dc voltage at the inverter with the minimum extinction angle of 18^{o}.

2 Dynamic Simulation

2.1 Introduction

The Power System Toolbox provides models of machines and control systems for performing transient stability simulations of a power system, and for building state variable models for small signal analysis and damping controller design. These dynamic models are coded as MATLAB functions.
MATLAB m-files are provided which enable a user to perform transient and small signal stability analysis without adding any new models. However, since the complete code is in the form of MATLAB m-files, by following a set of rules, the user can assemble customized models and applications.
In this tutorial, the model conventions, structure and data requirements, and the method of interconnecting the models to form power system simulation models is discussed.

2.1.1 The Power System Structure

The power system structure is defined by the **bus** and **line** specification matrices used in load flow calculations. A **solved** load flow case is required to set the operating conditions used to initialize the dynamic device models. Load flow data which represents an unsolved case will lead to dynamic models which are not at equilibrium when initialized.

2.1.2 Dynamic Data

Generators are defined in the specification matrix **mac_con**. There are three types of generator model

1. the electromechanical (em) , or classical model (**mac_em**)
2. the transient model (**mac_tra**)
3. the subtransient model (**mac_sub**)

All use the same fields for data, but only the subtransient model uses every field. Thus all generator models are specified using a single specification matrix.
Depending on requirements additional data must be specified for the

- generator controls
  - exciters - **exc_con**
  - power system stabilizers - **pss_con**
  - turbine-generators - **tg_con**
- induction motors - **ind_con** and **mld_con**
- induction generators – **igen_con**
- non-conforming loads - **load_con**
- static VAR compensators - **svc_con**
- Thyrister controlled series capacitors **– tcsc_con**
- HVDC lines
  - converters - **dcsp_con**
  - lines - **dcl_con**
  - controls - **dcc_con**

In small signal stability simulation, generators may be specified as infinite buses using **ibus_con**

2.1.3 Simulation Control Data

For transient stability simulation, some method for instructing the simulation program to apply a fault is required. The script file **y_switch** is an example of a simulation organization file. It uses the data specification file **sw_con** .

2.2 Dynamic Model Functions

The models available in this version of PST include:

1. Generator models
    - (a) **mac_em** -- electromechanical (classical) model
    - (b) **mac_tra** -- model including transient effect
    - (c) **mac_sub** -- model including subtransient effect [1]
    - (d) **mac_ib** -- a generator as infinite bus model (used only in small signal stability simulation)
2. Excitation system models
    - (a) **smpexc** -- simplified exciter model
    - (b) **exc_dc12** -- IEEE type DC1 and DC2 models [2]
    - (c) **exc_st3** -- IEEE type ST3 model [2]
3. Power system stabilizer model ... **pss**
4. Simplified thermal turbine-governor model ... **tg**
5. Simplified hydro turbine-governor model ... **tgh**
6. Induction Motor Model...**mac_ind**
7. Induction Generator Model ... **mac_igen**
8. Static VAR compensator model -- **svc** [3]
9. Thyristor Controlled Seried Compensator - **tcsc**
10. Load Modulation Control ... **lmod**
11. HVDC line model – **dc_line, dc_cont**
12. Non-conforming load model -- **nc_load**
13. Line flow function -- **line_pq**
14. Utility functions -- **pss_des** (power system stabilizer design), **statef** (frequency response from state space), **step_res** (step response from state space system models).

2.3 Standard Dynamic Drivers

Driving functions are provided for transient stability (**s_simu**)and small signal stability (**svm_mgen**). These functions provide an environment which requires only the system data to be specified and act much like stand-alone transient and small signal stability programs. Details are given in the function descriptions section which follows this tutorial.

2.4 Expanding the Capabilities of PST

Since the source code for all functions is provided, a user may expand PST to meet special modelling or simulation requirements. The following indicates the preferred form of dynamic models.

2.4.1 Model Structure

Each model function consists of 3 parts

1. initialization of the state variables - **flag = 0**
2. network interface computation - **flag = 1**
3. calculation of the rates of change of state variables - **flag = 2**

In general, there are 4 input variables to a function, namely, **i** (the device number), **k** (time step), **bus** and **flag**. A convention used in all the supplied models is that if **i** is zero, the model calculations are made using vector methods. Additional variables are normally required for dynamic models. In PST these variables are normally specified as global. The m_file **pst_var** declares all global variables used in PST. For consistency new global variables should be added to **pst_var**.

Most models require an interface mode, but some, such as the induction motor do not. If the mode does not exist, it is good practice to have a null section defined, see **mac_ind.m** for an example. In the case of the non-conforming load model, there are no state variables and hence no action is taken when this function is called with **flag = 2**. New models should be coded so that they exit without error if the corresponding index or data specification matrix does not exist. In this way a single driver program, which calls all possible models, will not fail when the driver is run for a data set which does not contain the new model.

2.4.2 Vector Computation

In MATLAB, it is important to use vector computation whenever possible, and avoid loops in the computation process. In this version of PST, index functions are used to store data about the different types of similar models, e.g., generators. For example, if a new exciter model is added, the index function **exc_indx.m** must be modified to include the appropriate indexes which are passed on to the new exciter model as global variables.

2.4.3 Use of Templates

New dynamic models are most easily and efficiently formed by modifying the existing models. The data input format for the model should follow the same conventions as that of the existing models. The state variables should have meanings in new models similar to those in existing models. If there is no confusion, states already defined in **pst_var** should be used.

2.5 Transient Stability Simulation

A power system transient stability simulation model consists of a set of differential equations determined by the dynamic models and a set of algebraic equations determined by the power system network.

In PST, the dynamic generator models, with **flag = 1**, calculate the generator internal node voltages, i.e., the voltage behind transient impedance for the electromechanical generator, transient generator, and the voltage behind subtransient impedance for the subtransient generator. In the induction motor model the internal voltages behind transient impedance are the states vdprime and vqprime. These internal voltages are used with a system admittance matrix reduced to the internal nodes and the non-conforming load bus nodes to compute the current injections into the generators and motors. When there is an HVDC link in the model, the reduced Y matrix has additional rows and columns associated with the equivalent HT terminals of the HVDC links. The current injections are then used in the generator and motor models, and the non-conforming load voltages are used in the SVC and HVDC link models with **flag = 2**, to calculate the rates of change of their state variables.

All models should detect for the existence if valid model data, e.g., if the required data is not supplied, the model function exits with no changes. In this way, the driver can contain all existing models and rely on the data set to define those necessary for the required simulation.

Rather than build a new simulation driver from scratch for every additional simulation model, it is recommended that new models be added to the general transient stability driver **s_simu**. The structure is quite straightforward and well documented within the code.

2.5.1 Example

The two-area system with subtransient generators, static exciters, thermal turbine governors and power system stabilizers is defined in the data file d2asbegp

```
% Two Area Test Case
% sub transient generators with static exciters, turbine/governors
% 50% constant current active loads
% load modulation
% with power system stabilizers
disp('Two-area test case with subtransient generator models')
disp('Static exciters and power system stabilizers')
disp('turbine/governors')
% bus data format
% bus:
% col1 number
% col2 voltage magnitude(pu)
% col3 voltage angle(degree)
% col4 p_gen(pu)
% col5 q_gen(pu),
% col6 p_load(pu)
% col7 q_load(pu)
% col8 G shunt(pu)
```

```
% col9 B shunt(pu)
% col10 bus_type
%       bus_type - 1, swing bus
%                - 2, generator bus (PV bus)
%                - 3, load bus (PQ bus)
% col11 q_gen_max(pu)
% col12 q_gen_min(pu)
% col13 v_rated (kV)
% col14 v_max  pu
% col15 v_min  pu
bus = [...
    1  1.03    18.5   7.00   1.61  0.00  0.00  0.00  0.00 1  5.0  -1.0  22.0  1.1  .9;
    2  1.01    8.80   7.00   1.76  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0  1.1  .9;
    3  0.9781  -6.1   0.00   0.00  0.00  0.00  0.00  3.00 3  0.0   0.0  230.0  1.5  .5;
    4  0.95   -10     0.00   0.00  9.76  1.00  0.00  0.00 3  0.0   0.0  115.0   1.05 .95;
   10 1.0103  12.1   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5;
   11 1.03    -6.8   7.16   1.49  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0   1.1  .9;
   12 1.01    -16.9  7.00   1.39  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0   1.1  .9;
   13 0.9899  -31.8  0.00   0.00  0.00  0.00  0.00  5.00 3  0.0   0.0  230.0  1.5  .5;
   14 0.95    -35    0.00   0.00  17.65 1.00  0.00  0.00 3  0.0   0.0  115.0  1.05  .95;
   20 0.9876   2.1   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5;
  101 1.00    -19.3  0.00   0.00  0.00  0.00  0.00  2.00 3  2.0   0.0  500.0  1.5  .5;
  110 1.0125  -13.4  0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5;
  120 0.9938  -23.6  0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5 ;
];
% line data format
% line:
%      col1     from bus
%      col2     to bus
%      col3     resistance(pu)
%      col4     reactance(pu)
%      col5     line charging(pu)
%      col6     tap ratio
%      col7     tap phase
%      col8     tapmax
%      col9     tapmin
%      col10    tapsize
line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.02;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13   14  0.0    0.005    0.00    1.0  0. 1.2 0.8 0.02;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
];
% Machine data format
% Machine data format
%       1. machine number,
%       2. bus number,
%       3. base mva,
%       4. leakage reactance x_l(pu),
%       5. resistance r_a(pu),
%       6. d-axis sychronous reactance x_d(pu),
%       7. d-axis transient reactance x'_d(pu),
%       8. d-axis subtransient reactance x"_d(pu),
%       9. d-axis open-circuit time constant T'_do(sec),
%      10. d-axis open-circuit subtransient time constant
%                T"_do(sec),
%      11. q-axis sychronous reactance x_q(pu),
%      12. q-axis transient reactance x'_q(pu),
%      13. q-axis subtransient reactance x"_q(pu),
%      14. q-axis open-circuit time constant T'_qo(sec),
%      15. q-axis open circuit subtransient time constant
%                T"_qo(sec),
```

```
%       16. inertia constant H(sec),
%       17. damping coefficient d_o(pu),
%       18. dampling coefficient d_1(pu),
%       19. bus number
%
% note: all the following machines use sub-transient model
mac_con = [ ...

1 1 900 0.200  0.00    1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.24 0.4   0.05...
                       6.5  0  0  1  0.0654  0.5743;
2 2 900 0.200  0.00    1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.25 0.4   0.05...
                       6.5  0  0  2  0.0654  0.5743;
3 11 900 0.200  0.00   1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.24 0.4   0.05...
                       6.5  0  0  3  0.0654  0.5743;
4 12 900 0.200  0.00   1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.25 0.4   0.05...
                       6.5  0  0  4  0.0654  0.5743;
];
% simple exciter model, type 0; there are three exciter models
exc_con = [...
0 1 0.01 200.0  0.05     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0    0;
0 2 0.01 200.0  0.05     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0    0;
0 3 0.01 200.0  0.05     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0    0;
0 4 0.01 200.0  0.05     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0    0;
];
% power system stabilizer model
%   col1    type 1 speed input; 2 power input
%   col2    generator number
%   col3    pssgain*washout time constant
%   col4    washout time constant
%   col5    first lead time constant
%   col6    first lag time constant
%   col7    second lead time constant
%   col8    second lag time constant
%   col9    maximum output limit
%   col10   minimum output limit
pss_con = [...
     1  1  100  10 0.05 0.015 0.08 0.01 0.2 -0.05;
     1  2  100  10 0.05 0.015 0.08 0.01 0.2 -0.05;
     1  3  100  10 0.05 0.015 0.08 0.01 0.2 -0.05;
     1  4  100  10 0.05 0.015 0.08 0.01 0.2 -0.05;
 ];
% governor model
% tg_con matrix format
%column        data          unit
%  1    turbine model number (=1)
%  2    machine number
%  3    speed set point   wf        pu
%  4    steady state gain 1/R       pu
%  5    maximum power order  Tmax   pu on generator base
%  6    servo time constant   Ts    sec
%  7    governor time constant  Tc  sec
%  8    transient gain time constant T3 sec
%  9    HP section time constant   T4   sec
% 10    reheater time constant    T5    sec

tg_con = [...
1  1  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  2  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  3  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  4  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
];
```

```
% induction motor data
% 1. Motor Number
% 2. Bus Number
% 3. Motor MVA Base
% 4. rs pu
% 5. xs pu - stator leakage reactance
% 6. Xm pu - magnetizing reactance
% 7. rr pu
% 8. xr pu - rotor leakage reactance
% 9. H  s  - motor plus load inertia constant
% 10. rr1 pu - second cage resistance
% 11. xr1 pu - intercage reactance
% 12. dbf    - deepbar factor
% 13. isat pu - saturation current
% 15. fraction of bus power drawn by motor ( if zero motor statrts at t=0)
ind_con = [];
% Motor Load Data
% format for motor load data - mld_con
% 1 motor number
% 2 bus number
% 3 stiction load pu on motor base (f1)
% 4 stiction load coefficient (i1)
% 5 external load  pu on motor base(f2)
% 6 external load coefficient (i2)
% load has the form
% tload = f1*slip^i1 + f2*(1-slip)^i2
mld_con = [];
%   col1    bus number
%   col2    proportion of constant active power load
%   col3    proportion of constant reactive power load
%   col4    proportion of constant active current load
%   col5    proportion of constant reactive current load
load_con = [4  0  0  0  0;%constant impedance
            14 0  0  0  0;
];
disp('50% constant current load')
%disp('load modulation')
%active and reactive load modulation enabled
%   col1    load number
%   col2    bus number
%   col3    MVA rating
%   col4    maximum output limit pu
%   col4    minimum output limit pu
%   col6    Time constant
lmod_con = [];
rlmod_con = [];
%Switching file defines the simulation control
% row 1 col1  simulation start time (s) (cols 2 to 6 zeros)
%       col7  initial time step (s)
% row 2 col1  fault application time (s)
%       col2  bus number at which fault is applied
%       col3  bus number defining far end of faulted line
%       col4  zero sequence impedance in pu on system base
%       col5  negative sequence impedance in pu on system base
%       col6  type of fault  - 0 three phase
%                            - 1 line to ground
%                            - 2 line-to-line to ground
%                            - 3 line-to-line
%                            - 4 loss of line with no fault
%                            - 5 loss of load at bus
%                            - 6 no action
%       col7  time step for fault period (s)
% row 3 col1  near end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for second part of fault (s)
% row 4 col1  far end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for fault cleared simulation (s)
% row 5 col1  time to change step length (s)
%       col7  time step (s)
%
%
%
% row n col1 finishing time (s)  (n indicates that intermediate rows may be inserted)
```

```
sw_con = [...
0    0    0    0    0    0    0.01;%sets intitial time step
0.1  3    101  0    0    0    0.01; % 3 ph fault at bus 3
0.15 0    0    0    0    0    0.01; %clear near end
0.20 0    0    0    0    0    0.01; %clear remote end
%0.50 0    0    0    0    0    0.01; % increase time step
%1.0  0    0    0    0    0    0.01; % increase time step
5.0  0    0    0    0    0    0]; % end simulation
%fpos=60;
%ibus_con = [0 1 1 1];% sets generators 2, 3 and 4 to be infinite buses
%                       behind source impedance in small signal stability model
```

Running s_simu and choosing the file d2asbegp, simulates a three-phase fault at bus 3 on the first line from bus 3 to bus 101. The fault is cleared at bus 3 0.01s after the fault is applied, and at bus 10 0.02 s after the fault is applied.

```
s_simu
non-linear simulation
Two-area test case with subtransient generator models
Static exciters
power system stabilizers
50% constant current load
load modulation
enter the base system frequency in Hz - [60]
enter system base MVA - [100]
Do you want to solve loadflow > (y/n)[y]
inner load flow iterations
     4
 tap iterations
     1

Performing simulation.
constructing reduced y matrices
initializing motor,induction generator, svc and dc control models
initializing other models
generators
xqpp made equal to xdpp at generators
     1     3
generator controls
non-linear loads
elapsed time = 31.4838s
You can examine the system response
Type 1 to see all machine angles in 3D
     2 to see all machine speed deviation in 3D
     3 to see all machine turbine powers
     4 to see all machine electrical powers
     5 to see all field voltages
     6 to see all bus voltage magnitude in 3D
     7 to see the line power flows
     0 to quit and plot your own curves
enter selection >>
```

As the simulation progresses, the voltage at the fault bus (bus 3) is plotted. The final response is shown in Figure 6.

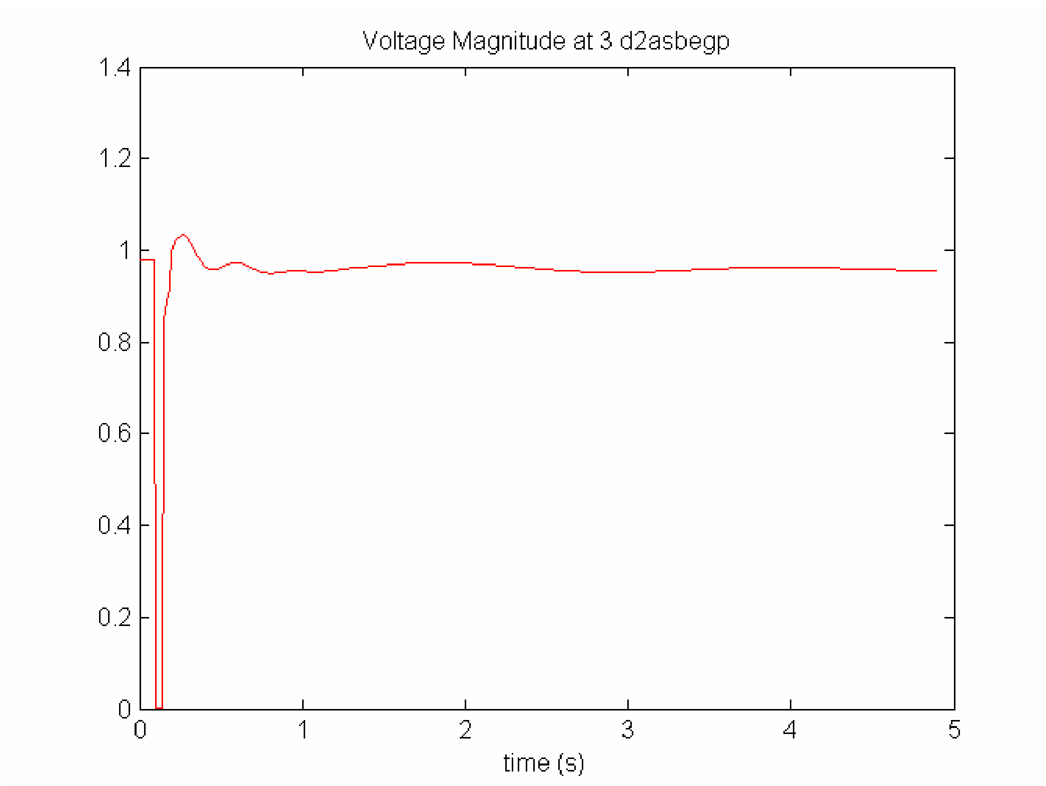


Figure 6 Bus 3 voltage magnitude response to three-phase fault

Other plots may be chosen from the menu which is displayed after the simulation is complete. The available values are shown in the MATLAB Workspace.

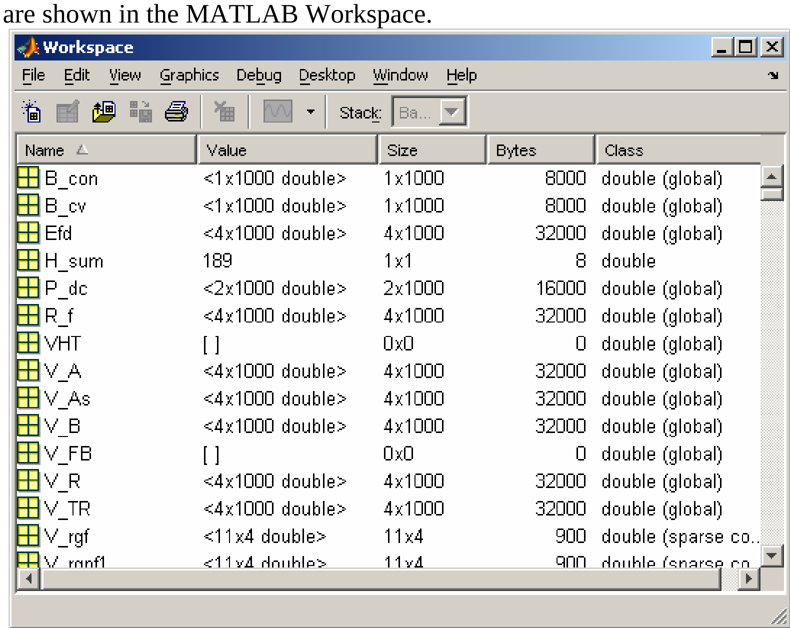


Figure 7 The MATLAB Workspace following the completion of s_simu

Thus to plot the generator field voltages

```
plot(t,Efd)
```

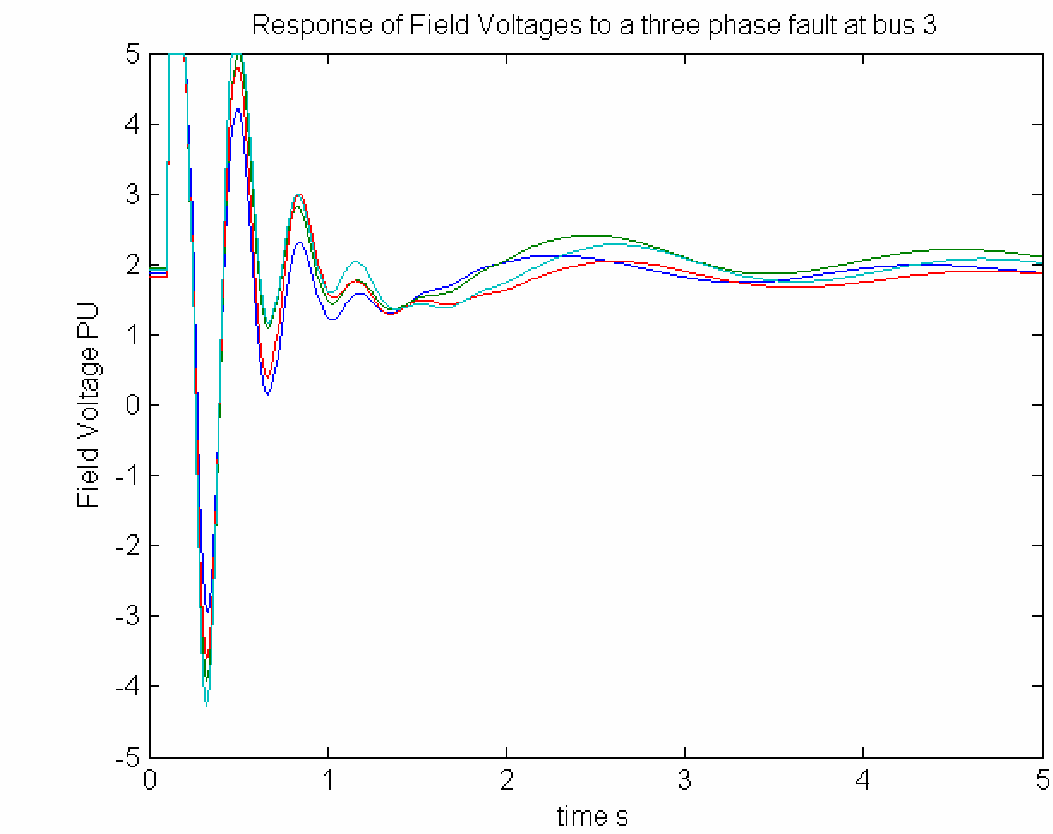


Figure 8 Response of generator field voltage to three phase fault at bus 3

2.5.2 Applying disturbances to Control Reference Inputs

Each control has a reference input. Initially, the reference inputs are set to give the required steady state output from the control, e.g., the exciter reference voltage is set to give the value of generator field voltage determined from the initialization of the generator. The reference inputs may be modulated in the transient simulation using special modulation m_files. These files are normally set to give zero change in the reference input, but they may be modified by a user so as to apply any function of time. The modulation functions are:

- mexc_sig - modulates the exciter voltage reference
- mpm_sig - modulates the generator shaft torque
- mtg_sig - modulates the governor power reference
- msvc_sig - modulates the svc reference
- ml_sig - modulates the active load at a bus
- rml_sig - modulates the reactive load at a bus
- mdc_sig - modulates dc reference inputs

The construction of the modulation functions is similar. The following is the mexc_sig m-file, set to produce no modulation output.

```
function f = mexc_sig(t,k)
% Syntax: f = mexc_sig(t,k)
% 1:20 PM 15/08/97
% defines modulation signal for exciter control
global exc_sig n_exc
f=0; %dummy variable
if n_exc~=0
%  exc_sig(:,k)=zeros(n_exc,1);
%  exc_sig(1,k)=0.1;
%end
 if t<=0
     exc_sig(:,k) = zeros(n_exc,1);
 else
    exc_sig(:,k) = zeros(n_exc,1);
    %exc_sig(1,k) = 0.05;
 end
end
return
```

To see the effect of a step input of 0.05 in the exciter reference of G1 in the two area system, the switching file should be set for no-fault (6 in column 6 of line 2)

```
sw_con = [...
0    0    0    0    0    0    0.01;%sets intitial time step
0.1  3    101  0    0    6    0.01; % no fault
0.15 0    0    0    0    0    0.01; %clear near end
0.20 0    0    0    0    0    0.01; %clear remote end
%0.50 0    0    0    0    0    0.01; % increase time step
%1.0  0    0    0    0    0    0.01; % increase time step
5.0  0    0    0    0    0    0]; % end simulation
```

Modify mexc_sig as follows, and save the file

```
function f = mexc_sig(t,k)
% Syntax: f = mexc_sig(t,k)
% 1:20 PM 15/08/97
% defines modulation signal for exciter control
global exc_sig n_exc
f=0; %dummy variable
if n_exc~=0
%  exc_sig(:,k)=zeros(n_exc,1);
%  exc_sig(1,k)=0.1;
%end
 if t<=0
     exc_sig(:,k) = zeros(n_exc,1);
 else
    %exc_sig(:,k) = zeros(n_exc,1);
    exc_sig(1,k) = 0.05;
 end
end
return
```

Run s_simu. The voltage at bus 3 is displayed as the simulation progresses. It is shown in Figure 9.

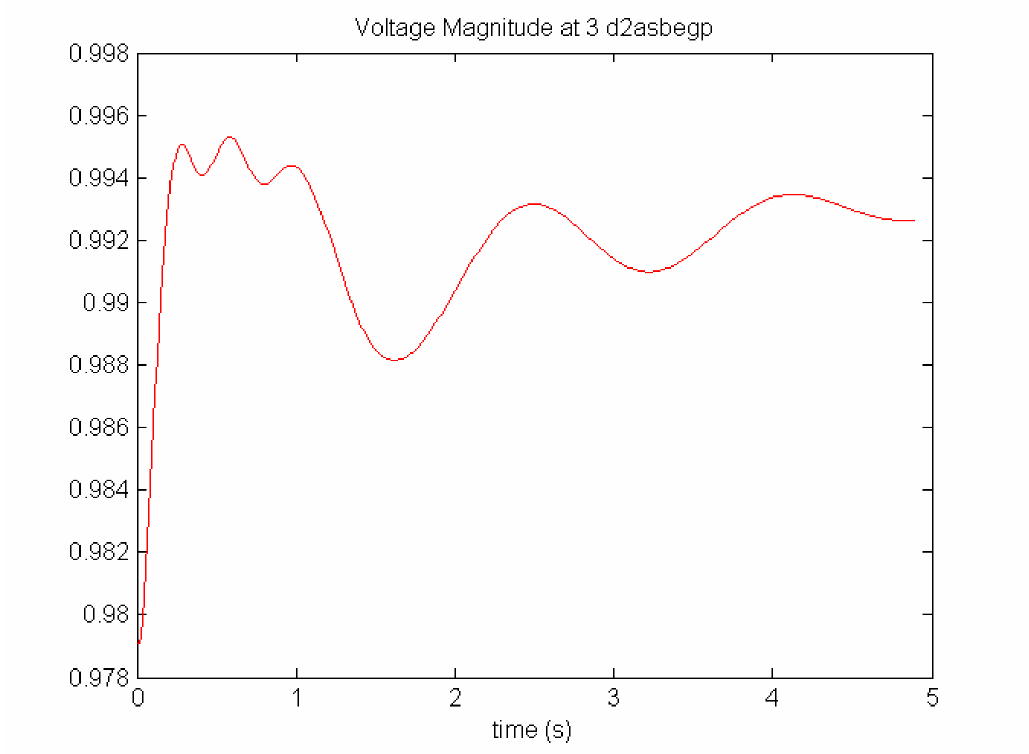


Figure 9 Response of voltage magnitude at bus 3 to a 0.05 step change in Vref at generator 1

The response of the field voltage at generator 1 is shown in Figure 10. Even with this small input, the response is nonlinear – it is limited by the maximum limit of the exciter.

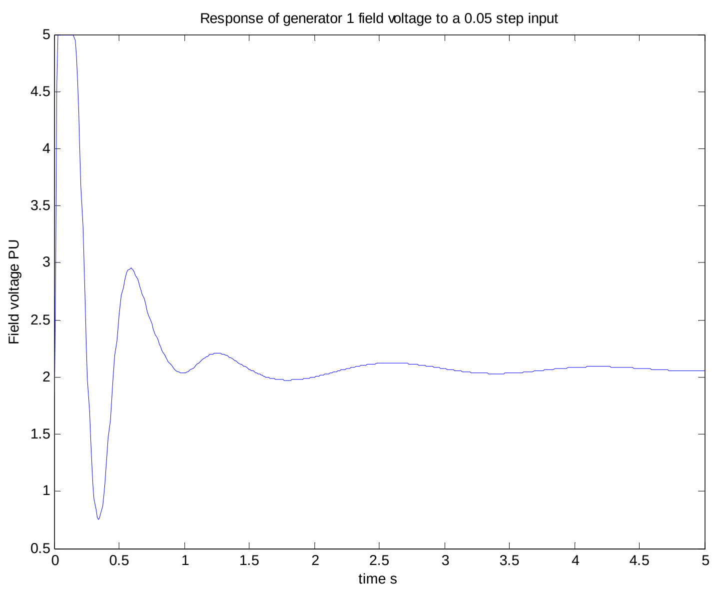


Figure 10 Response of the field voltage at generator 1 to a 0.05 step change in Vref

As a second example, consider the modulation of the load at bus 4. To do this active load modulation must be enabled at the bus. First return mexc_sig to its original form which applies no change to the exciter reference. Add active and reactive load modulation.

```
%active and reactive load modulation enabled
%   col1    load number
%   col2    bus number
%   col3    MVA rating
%   col4    maximum output limit pu
%   col4    minimum output limit pu
%   col6    Time constant
lmod_con = [...
1 4  100 1 -1 1 0.05;
2 14 100 1 -1 1 0.05;
];

rlmod_con = [...
1 4   100 1  -1  1  0.05;
2 14  100 1  -1  1  0.05;
];
```

The loads at these buses must be declared as non-conforming loads in order for them to be modulated. In this system model they are declared as non-conforming loads, and are 50/50 constant current/constant impedance.

```
function f = ml_sig(t,k)
% Syntax: f = ml_sig(t,k)
%4:40 PM 15/08/97
% defines modulation signal for lmod control
global lmod_sig n_lmod
f=0; %dummy variable
% you modify the following to do what you want with the load
% lmod_con must be specified in the data file
% and the load bus must be in the nonconforming load list.
if n_lmod~=0
  if t<=0
     lmod_sig(:,k)= zeros(n_lmod,1);
  else
     %lmod_sig(:,k) = zeros(n_lmod,1);
     lmod_sig(1,k) = 0.5;
  end
end
return
```

The above causes the active load at bus 4 to be increased by 0.5 PU, and running s_simu gives

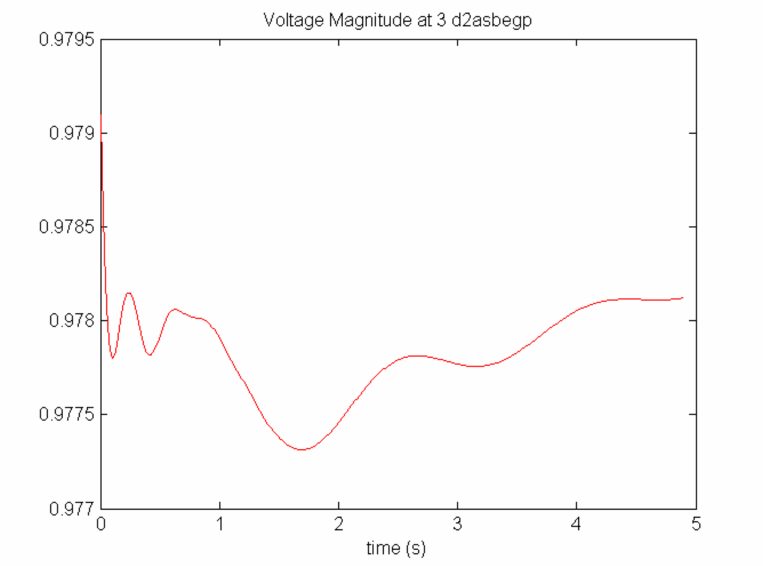


Figure 11 Response of the voltage magnitude at bus 3 to a 0.5 PU step change in active load at bus 4

The response of bus 4 voltage magnitude is shown in Figure 12.

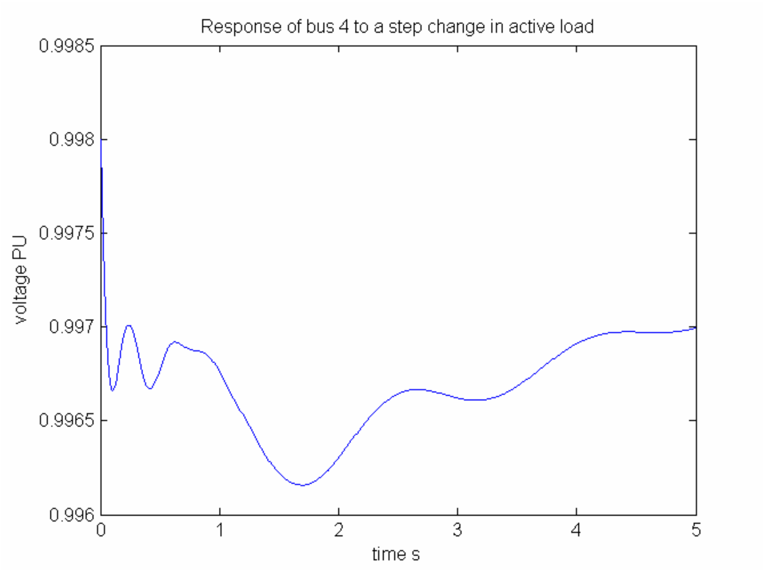


Figure 12 Response of bus 4 to a 0.5 step change in active load

The response to a step change in the reactive load may be obtained by modifying rlm_sig. The function lm_sig must be returned to the zero disturbance form.

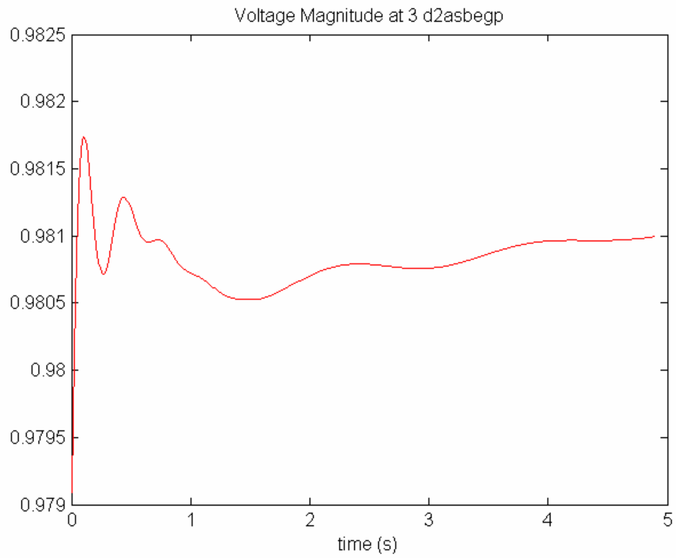


Figure 13 Response of bus 3 voltage magnitude to a 0.1 step change in reactive load at bus 4

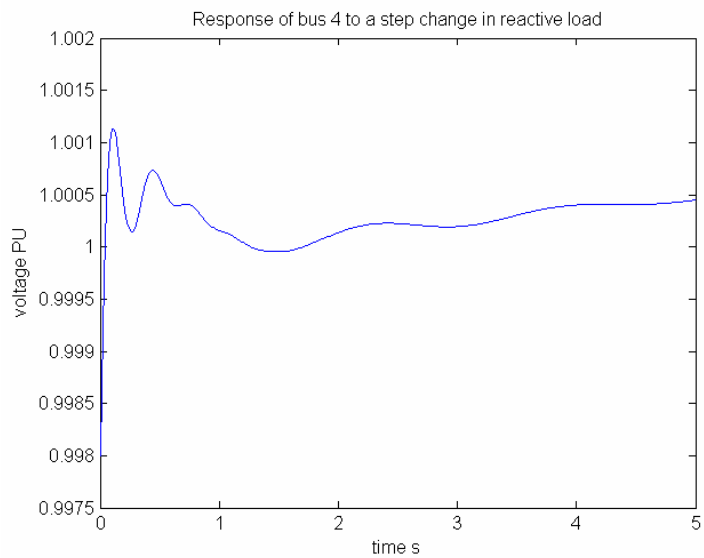


Figure 14 Response of bus 4 terminal voltage magnitude to a 0.1 step change in reactive load

2.6 Small Signal Stability

The stability of the operating point of a dynamic system to small disturbances is termed small signal stability. To test for small signal stability the system's dynamic equations are linearized about a steady state operating point to get a linear set of state equations

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

where A is the state matrix; B is the input matrix; C is the output matrix ; D is the feed forward matrix; x is the state vector and u is the input.

In some small signal stability programs (e.g. MatNetEig), the state matrices are calculated analytically from the Jacobians of the non-linear state equations. In the Power System Toolbox, on the other hand, the linearization is performed by calculating the Jacobian numerically. This has the advantage of using identical dynamic models for transient and small signal stability. However, there is some loss of accuracy, particularly in the zero eigenvalue which is characteristic of most inter-connected power systems.

In PST, starting from the states determined from model initialization, a small perturbation is applied to each state in turn. The change in the rates of change of all the states divided by the magnitude of the perturbation gives a column of the state matrix corresponding to the disturbed state. A permutation matrix **p_mat** is used to arrange the states in a logical order. Following each rate of change of state calculation, the perturbed state is returned to its equilibrium value and the intermediate variable values are reset to there initial values. Each step in this process is similar to a single step in a simulation program. The input matrix B, the output matrix C and the feed forward matrix D can be determined in a similar manner.

A single driver, **svm_mgen**, for small signal stability is provided. It is organized similarly to the transient stability simulation driver **s_simu**. New models should be designed to work satisfactorily in either driver. Generally, if a model is satisfactory in **s_simu**, it will be satisfactory in **svm_mgen**.

2.6.1 Example

Using the same file as in the previous example, with active and reactive load modulation specified at buses 4 and 14, running svm_mgen gives

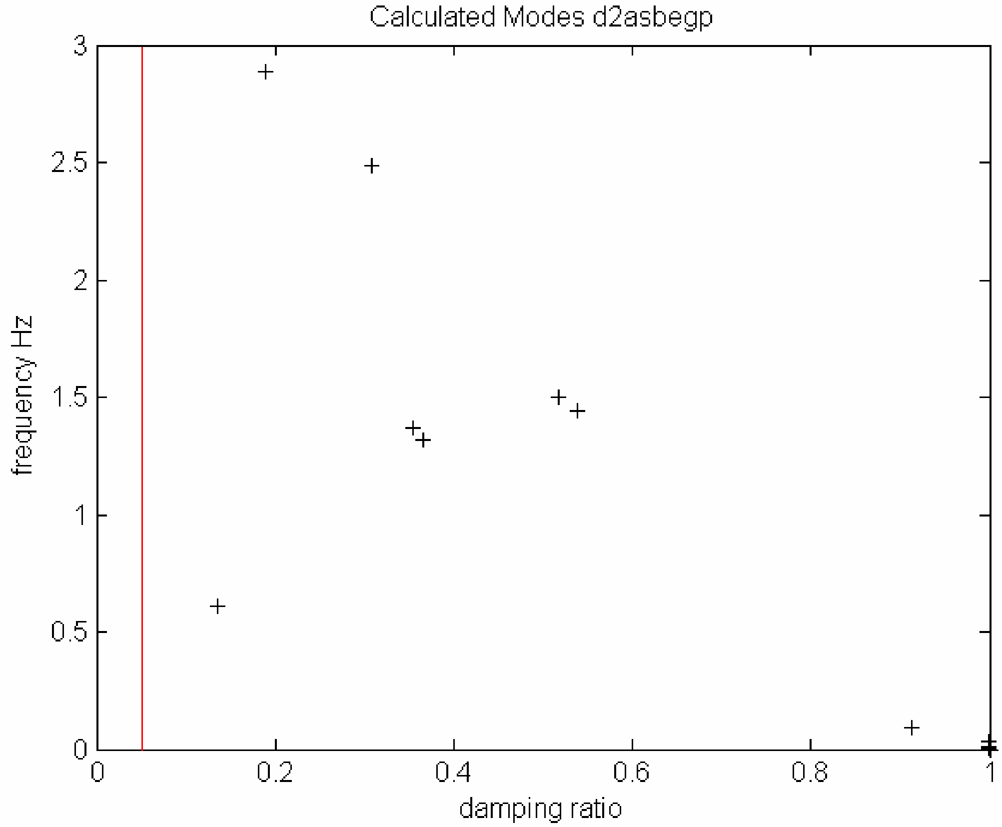
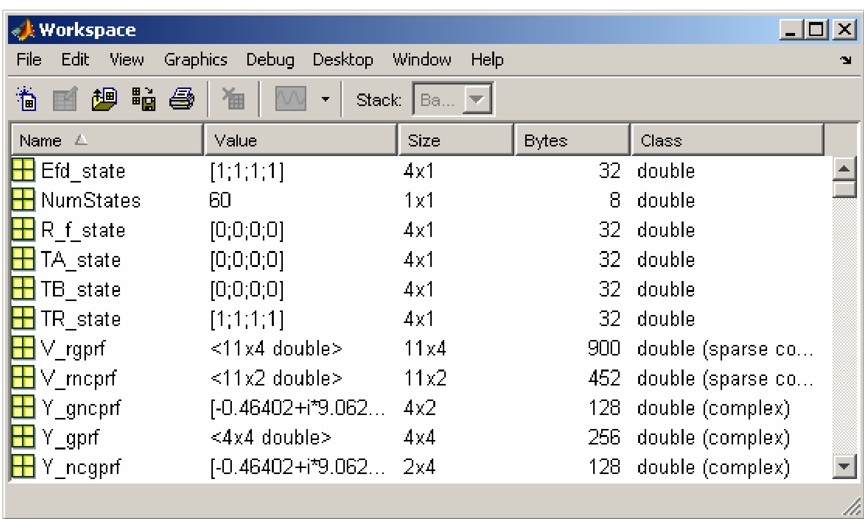


Figure 15 Calculated modes d2asbegp

The MATLAB Workspace after running svm_mgen is shown in Figure 16.



Name	Value	Size	Bytes	Class
Efd_state	[1;1;1;1]	4x1	32	double
NumStates	60	1x1	8	double
R_f_state	[0;0;0;0]	4x1	32	double
TA_state	[0;0;0;0]	4x1	32	double
TB_state	[0;0;0;0]	4x1	32	double
TR_state	[1;1;1;1]	4x1	32	double
V_rgprf	<11x4 double>	11x4	900	double (sparse co...
V_rncprf	<11x2 double>	11x2	452	double (sparse co...
Y_gncprf	[-0.46402+i*9.062...	4x2	128	double (complex)
Y_gprf	<4x4 double>	4x4	256	double (complex)
Y_ncgprf	[-0.46402+i*9.062...	2x4	128	double (complex)

Figure 16 The MATLAB Workspace following a run of svm_mgen

The number of dynamic states in this model is 60 (NumStates); 56 for the generators and their controls and 4 for the active and reactive load modulation. The eigenvalues damping ratios and frequencies of the modes may be obtained using

```
[l damp freq]

ans =

1.1981e-005                      1                          0
    -0.074232                      1                          0
     -0.10079                      1                          0
     -0.10091                      1                          0
     -0.10098                      1                          0
     -0.19541                      1                          0
     -0.19798                      1                          0
     -0.19801                      1                          0
     -0.49299                      1                          0
      -1.2731 -    0.57246i      0.91203                    0.09111
      -1.2731 +    0.57246i      0.91203                    0.09111
      -1.9122 -  0.0034404i            1                 0.00054756
      -1.9122 +  0.0034404i            1                 0.00054756
      -1.9146                      1                          0
      -3.1846                      1                          0
      -3.2735                      1                          0
      -3.6349 -   0.056365i      0.99988                  0.0089708
      -3.6349 +   0.056365i      0.99988                  0.0089708
     -0.52484 -     3.8483i      0.13513                    0.61248
     -0.52484 +     3.8483i      0.13513                    0.61248
      -3.2505 -     8.2795i      0.36545                     1.3177
      -3.2505 +     8.2795i      0.36545                     1.3177
       -3.248 -     8.5995i      0.35333                     1.3687
       -3.248 +     8.5995i      0.35333                     1.3687
      -10.065                      1                          0
      -10.066                      1                          0
      -10.098                      1                          0
      -10.114                      1                          0
      -5.7661 -     9.0385i      0.53783                     1.4385
      -5.7661 +     9.0385i      0.53783                     1.4385
      -5.7078 -     9.4463i      0.51716                     1.5034
      -5.7078 +     9.4463i      0.51716                     1.5034
      -5.0489 -     15.634i      0.30732                     2.4882
      -5.0489 +     15.634i      0.30732                     2.4882
      -3.4997 -     18.135i      0.18949                     2.8862
      -3.4997 +     18.135i      0.18949                     2.8862
          -20                      1                          0
          -20                      1                          0
          -20                      1                          0
          -20                      1                          0
      -30.705                      1                          0
      -31.178                      1                          0
      -36.048                      1                          0
        -36.2                      1                          0
      -41.102                      1                          0
      -41.147                      1                          0
      -41.625 -   0.070921i            1                   0.011287
      -41.625 +   0.070921i            1                   0.011287
      -94.588                      1                          0
      -94.633                      1                          0
      -96.049 -    0.22277i            1                   0.035455
      -96.049 +    0.22277i            1                   0.035455
         -100                      1                          0
         -100                      1                          0
         -100                      1                          0
         -100                      1                          0
      -105.29 -    0.21481i            1                   0.034189
      -105.29 +    0.21481i            1                   0.034189
      -106.14                      1                          0
      -106.22                      1                          0
```

All eigenvalues, apart from the theoretically zero eigenvalue, have negative real parts, i.e, the system is stable. In the plot of frequency against damping ratio, the damping ratio of the effectively zero eigenvalue is taken as one, if its magnitude is less than 10^{-4}.

The nature of each mode may be identified from the corresponding eigenvector. The states in the small signal model are ordered so that those associated with each generator are grouped in order.
The generator states are first. They are grouped as follows
rotor angle
rotor speed
Efd'
ψd"
ψq'
ψq"
up to 5 exciter states
up to 3 power system stabilizer states
up to 3 turbine/governor states
up to three induction motor states
up to 3 induction generator states
up to 2 svc states
1 active load modulation state
1 reactive load modulation state

The generator state numbers may be obtained from mac_state. The first column gives the mode number; the second the type of state; and the third the generator number. Numbers in column 2 from 1 to 6 represent the generator states, numbers 7 to 11 represent the exciter states, numbers 12 to14 represent power system stabilizer states, and 15 to 17 represent the governor states.

```
mac_state =

     1     1     1
     2     2     1
     3     3     1
     4     4     1
     5     5     1
     6     6     1
     7     7     1
     8    10     1
     9    12     1
    10    13     1
    11    14     1
    12    21     1
    13    22     1
    14    23     1
    15     1     2
    16     2     2
    17     3     2
    18     4     2
    19     5     2
    20     6     2
    21     7     2
    22    10     2
    23    12     2
    24    13     2
    25    14     2
    26    21     2
    27    22     2
    28    23     2
    29     1     3
    30     2     3
    31     3     3
    32     4     3
    33     5     3
    34     6     3
    35     7     3
    36    10     3
    37    12     3
    38    13     3
    39    14     3
    40    21     3
    41    22     3
    42    23     3
    43     1     4
    44     2     4
    45     3     4
    46     4     4
    47     5     4
    48     6     4
    49     7     4
    50    10     4
    51    12     4
    52    13     4
    53    14     4
    54    21     4
    55    22     4
    56    23     4
```

Thus for mode 1 which is the effectively zero eigenvalue, the eigenvector is

```
[(1:NumStates)' u(:,1)]

ans =

           1          0.5
           2  1.5891e-008
           3  3.8309e-006
           4  3.5872e-006
           5  2.2235e-006
           6  2.9003e-006
           7 -4.0544e-008
           8   8.109e-006
           9  1.5889e-008
          10   -7.615e-011
          11 -7.6136e-011
          12 -3.9726e-007
          13 -3.9726e-007
          14 -2.9793e-007
          15          0.5
          16  1.5891e-008
          17   3.998e-006
          18  3.6474e-006
          19   2.454e-006
          20  3.2009e-006
          21 -6.3088e-008
          22   1.262e-005
          23  1.5889e-008
          24 -7.6145e-011
          25 -7.6152e-011
          26 -3.9726e-007
          27 -3.9726e-007
          28 -2.9793e-007
          29          0.5
          30  1.5891e-008
          31  3.4891e-006
          32  3.2739e-006
          33  2.0683e-006
          34  2.6978e-006
          35 -3.5293e-008
          36  7.0618e-006
          37  1.5889e-008
          38 -7.6145e-011
          39 -7.6143e-011
          40 -3.9726e-007
          41 -3.9726e-007
          42 -2.9793e-007
          43          0.5
          44   1.589e-008
          45  3.8583e-006
          46  3.5378e-006
          47  2.3714e-006
          48  3.0931e-006
          49 -5.6499e-008
          50  1.1299e-005
          51  1.5889e-008
          52 -7.6144e-011
          53 -7.6146e-011
          54 -3.9726e-007
          55 -3.9726e-007
          56 -2.9793e-007
          57            0
          58            0
          59            0
          60            0
```

This eigenvector has non-zero entries in the rows associated with the rotor angle states. The eigenvalue would be exactly zero if the load flow was solved to a very low tolerance, and is due to the fact that the system dynamics do not alter if all the bus voltage angles are changed by the same amount.

Eigenvalues 19 and 20 correspond to the inter-area mode.

```
[(1:NumStates)' u(:,19:20)]

ans =
1                  -0.030566 +    0.10935i    -0.030566 -    0.10935i
2                  0.0011588 + 0.00015978i    0.0011588 - 0.00015978i
3                    0.01681 +   0.024459i      0.01681 -   0.024459i
4                   0.014906 +   0.019039i     0.014906 -   0.019039i
5                 0.00097158 +  0.0023634i   0.00097158 -  0.0023634i
6                  0.0022508 +  0.0025479i    0.0022508 -  0.0025479i
7                  0.0084215 + 0.00076658i    0.0084215 - 0.00076658i
8                    0.71582 -    0.49534i      0.71582 +    0.49534i
9               -7.3862e-006 +2.9296e-005i -7.3862e-006 -2.9296e-005i
10                  -0.04662 -  0.0070506i     -0.04662 +  0.0070506i
11                 -0.047011 + 0.00029728i    -0.047011 - 0.00029728i
12                 -0.024775 -   0.014278i    -0.024775 +   0.014278i
13                 0.0021665 -   0.013706i    0.0021665 +   0.013706i
14                0.00052338 + 0.00012863i   0.00052338 - 0.00012863i
15                -0.0048905 +   0.083325i   -0.0048905 -   0.083325i
16                0.00085739 - 6.608e-005i   0.00085739 + 6.608e-005i
17                 0.0015332 +   0.031379i    0.0015332 -   0.031379i
18                -0.0023372 +   0.023349i   -0.0023372 -   0.023349i
19                -0.0078935 +  0.0033738i   -0.0078935 -  0.0033738i
20                -0.0084537 +  0.0076796i   -0.0084537 -  0.0076796i
21                 0.0029782 -  0.0024985i    0.0029782 +  0.0024985i
22                         1                          1
23              -7.3351e-007 +2.2199e-005i -7.3351e-007 -2.2199e-005i
24                 -0.034592 +  0.0022116i    -0.034592 -  0.0022116i
25                 -0.033726 +  0.0075513i    -0.033726 -  0.0075513i
26                 -0.020027 -  0.0063903i    -0.020027 +  0.0063903i
27               -0.00058291 -   0.010184i  -0.00058291 +   0.010184i
28                0.00039607 +1.0712e-005i   0.00039607 -1.0712e-005i
29                 -0.014101 -    0.14862i    -0.014101 +    0.14862i
30                -0.0014975 + 0.00035085i   -0.0014975 - 0.00035085i
31                 -0.024464 +   0.010685i    -0.024464 -   0.010685i
32                 -0.025875 +   0.015371i    -0.025875 -   0.015371i
33                -0.0077239 +   0.010998i   -0.0077239 -   0.010998i
34                -0.0049198 +    0.01719i   -0.0049198 -    0.01719i
35                 -0.015525 +  0.0055899i    -0.015525 -  0.0055899i
36                   0.40959 +    0.62916i      0.40959 -    0.62916i
37               -4.763e-006 -3.9439e-005i  -4.763e-006 +3.9439e-005i
38                  0.060541 -   0.013352i     0.060541 +   0.013352i
39                  0.057553 -   0.022554i     0.057553 +   0.022554i
40                  0.037144 +  0.0058288i     0.037144 -  0.0058288i
41                 0.0038104 +   0.017843i    0.0038104 -   0.017843i
42               -0.00070302 +8.9181e-005i  -0.00070302 -8.9181e-005i
43                -0.0098859 -    0.13092i   -0.0098859 +    0.13092i
44                -0.0013226 + 0.00028318i   -0.0013226 - 0.00028318i
45                 -0.024433 +   0.015799i    -0.024433 -   0.015799i
46                 -0.026588 +   0.018475i    -0.026588 -   0.018475i
47                 -0.010643 +   0.011629i    -0.010643 -   0.011629i
48                 -0.008314 +   0.019255i    -0.008314 -   0.019255i
49                 -0.014864 +  0.0044114i    -0.014864 -  0.0044114i
50                   0.56109 +    0.65147i      0.56109 -    0.65147i
51              -3.5213e-006 -3.4758e-005i -3.5213e-006 +3.4758e-005i
52                  0.053458 -   0.010716i     0.053458 +   0.010716i
53                  0.050987 -   0.018858i     0.050987 +   0.018858i
54                  0.032561 +  0.0057532i     0.032561 -  0.0057532i
55                 0.0030488 +   0.015754i    0.0030488 -   0.015754i
56               -0.00061966 + 6.653e-005i  -0.00061966 - 6.653e-005i
57                         0                          0
58                         0                          0
59                         0                          0
60                         0                          0
```

The rotor angle states may be identified using

```
ang_idx = find(mac_state(:,2)==1)
ang_idx =
```

```
  1
 15
 29
 43
```

A compass plot of the rotor angle state terms of the eigenvector is shown in Figure 17. It was produced using

```
compass(u(ang_idx,20))
```

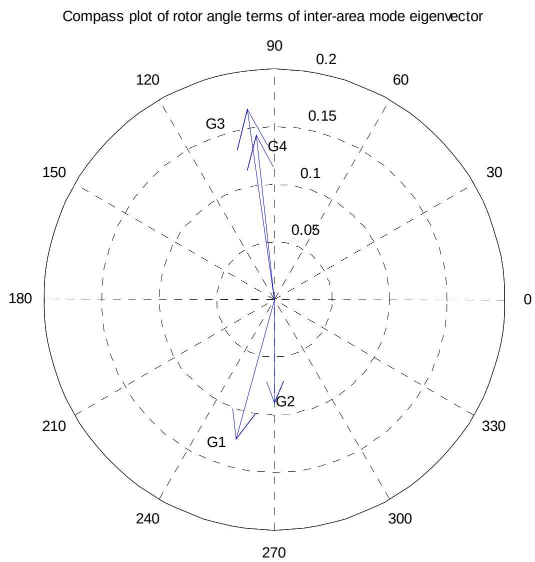


Figure 17 Compass plot of rotor angle terms of inter-area mode eigenvector

The eigenvector associated with a mode indicates the relative changes in the states which would be observed when that mode of oscilation is excited. It enables us to confirm that mode 20 is an inter-area mode, since generators 1 and 2 are oscilating against generators 3 and 4. However, the largest components of the eigenvector are those associated with the second exciter state. This means that the inter-area mode may be most easily observed by monitoring those states. It does not mean that these states are necessarily good for controlling the inter-area mode.

Participation factors are useful measures for indicating the best generator for power system stabilizer placement. They give the sensitivity of an eigenvalue to a change in the diagonal elements of the state matrix. The speed participation factors indicate the sensitivity of a mode to added damping at the shaft of the generators. A bar chart of the real part of speed participation for the inter-area mode is obtained using

```
bar(real(p(ang_idx+1,20)))
```

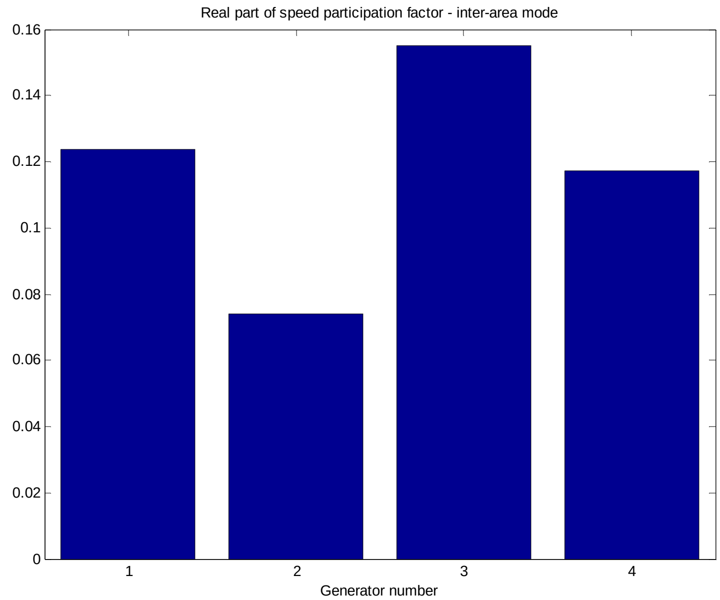


Figure 18 Real part of generator speed participation factors

If the real part of the speed participation is positive, a damping torque at the corresponding generator's shaft will add damping to the mode. In this case, a damping torque at any of the generators will add to the damping of the inter-area mode.

A state space model of the system may be constructed, either using the stsp object available from Graham Rogers, or the MATLAB Control Toolbox. The state matrix following the completion of svm_mgen is stored as a_mat, and b, c and d matrices are available for normal controls.

Thus, to form a state space model using the stsp object for vref input on generator 1 and Efd output from generator 1:

```
svrefd1 = stsp(a_mat,b_vr(:,1),c_Efd(1,:),0);
```

and the response to a step input of magnitude 0.05 is obtained using

```
[r,t]=stepres(svrefd1,0.05,5,0.01);
```

The response is shown in Figure 19. It can be seen to be similar to the non-linear response obtained using s_simu and shown in Figure 10. Since the model is linear, there is no limiting of the response.

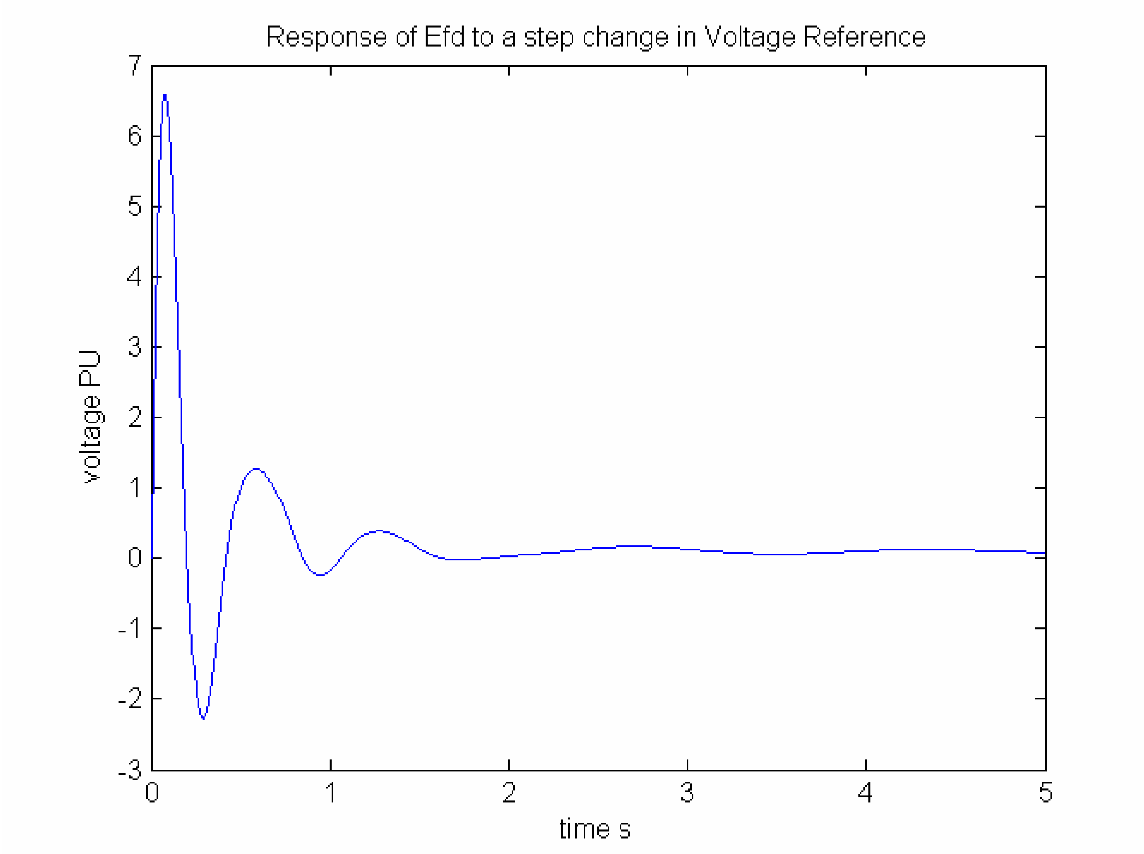


Figure 19 Response of linear model Efd to a 0.05 step change in exciter voltage reference

2.7 Damping Controller Design

For the two-area system with no PSS, the data file is

```
% Two Area Test Case
% sub transient generators with static exciters, turbine/governors
% 50% constant current active loads
% load modulation

disp('Two-area test case with subtransient generator models')
disp('Static exciters')
disp('turbine/governors')
% bus data format
% bus:
% col1 number
% col2 voltage magnitude(pu)
% col3 voltage angle(degree)
% col4 p_gen(pu)
% col5 q_gen(pu),
% col6 p_load(pu)
% col7 q_load(pu)
% col8 G shunt(pu)
% col9 B shunt(pu)
% col10 bus_type
%       bus_type - 1, swing bus
%               - 2, generator bus (PV bus)
%               - 3, load bus (PQ bus)
% col11 q_gen_max(pu)
% col12 q_gen_min(pu)
% col13 v_rated (kV)
% col14 v_max  pu
```

```
% col15 v_min  pu


bus = [...
    1  1.03    18.5   7.00   1.61  0.00  0.00  0.00  0.00 1  5.0  -1.0  22.0  1.1  .9;
    2  1.01     8.80  7.00   1.76  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0  1.1  .9;
    3  0.9781  -6.1   0.00   0.00  0.00  0.00  0.00  3.00 3  0.0   0.0  230.0 1.5  .5;
    4  0.95   -10     0.00   0.00  9.76  1.00  0.00  0.00 3  0.0   0.0  115.0 1.05 .95;
    10 1.0103  12.1   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0 1.5  .5;
    11 1.03    -6.8   7.16   1.49  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0  1.1  .9;
    12 1.01   -16.9   7.00   1.39  0.00  0.00  0.00  0.00 2  5.0  -1.0  22.0  1.1  .9;
    13 0.9899 -31.8   0.00   0.00  0.00  0.00  0.00  5.00 3  0.0   0.0  230.0 1.5  .5;
    14 0.95   -35     0.00   0.00  17.65 1.00  0.00  0.00 3  0.0   0.0  115.0 1.05  .95;
    20 0.9876   2.1   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0 1.5  .5;
   101 1.00   -19.3   0.00   1.09  0.00  0.00  0.00  0.00 2  2.0   0.0  500.0 1.5  .5;
   110 1.0125 -13.4   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0 1.5  .5;
   120 0.9938 -23.6   0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0 1.5  .5 ];

% line data format
% line: from bus, to bus, resistance(pu), reactance(pu),
%       line charging(pu), tap ratio, tap phase, tapmax, tapmin, tapsize

line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.02;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13   14  0.0    0.005    0.00    1.0  0. 1.2 0.8 0.02;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];

% Machine data format
% Machine data format
%       1. machine number,
%       2. bus number,
%       3. base mva,
%       4. leakage reactance x_l(pu),
%       5. resistance r_a(pu),
%       6. d-axis sychronous reactance x_d(pu),
%       7. d-axis transient reactance x'_d(pu),
%       8. d-axis subtransient reactance x"_d(pu),
%       9. d-axis open-circuit time constant T'_do(sec),
%      10. d-axis open-circuit subtransient time constant
%                T"_do(sec),
%      11. q-axis sychronous reactance x_q(pu),
%      12. q-axis transient reactance x'_q(pu),
%      13. q-axis subtransient reactance x"_q(pu),
%      14. q-axis open-circuit time constant T'_qo(sec),
%      15. q-axis open circuit subtransient time constant
%                T"_qo(sec),
%      16. inertia constant H(sec),
%      17. damping coefficient d_o(pu),
%      18. dampling coefficient d_1(pu),
%      19. bus number
%
% note: all the following machines use sub-transient model
mac_con = [ ...

1 1 900 0.200  0.00    1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.24 0.4   0.05...
                       6.5  0  0  3  0.0654  0.5743;
2 2 900 0.200  0.00    1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.25 0.4   0.05...
                       6.5  0  0  3  0.0654  0.5743;
3 11 900 0.200  0.00   1.8  0.30  0.25 8.00  0.03...
```

```
                       1.7  0.55  0.24 0.4   0.05...
                       6.5  0  0  3  0.0654  0.5743;
4 12 900 0.200  0.00   1.8  0.30  0.25 8.00  0.03...
                       1.7  0.55  0.25 0.4   0.05...
                       6.5  0  0  3  0.0654  0.5743];

exc_con = [...
0 1 0.01 200.0  0.05     0     0     5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 2 0.01 200.0  0.05     0     0     5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 3 0.01 200.0  0.05     0     0     5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 4 0.01 200.0  0.05     0     0     5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0];

pss_con = [];

% governor model
% tg_con matrix format
%column        data         unit
%  1    turbine model number (=1)
%  2    machine number
%  3    speed set point   wf        pu
%  4    steady state gain 1/R       pu
%  5    maximum power order  Tmax   pu on generator base
%  6    servo time constant    Ts   sec
%  7    governor time constant  Tc  sec
%  8    transient gain time constant T3 sec
%  9    HP section time constant   T4   sec
% 10    reheater time constant    T5    se
tg_con = [...
1  1  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  2  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  3  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  4  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
];
load_con = [4  0  0  .5  0;
            14 0  0  .5  0];
disp('50% constant current load')
%disp('load modulation')
%active and reactive load modulation enabled
lmod_con = [...
%1 4  100 1 -1 1 0.05;
%2 14 100 1 -1 1 0.05
];

rlmod_con = [...
%1 4   100 1  -1  1  0.05;
%2 14  100 1  -1  1  0.05
];
%Switching file defines the simulation control
% row 1 col1  simulation start time (s) (cols 2 to 6 zeros)
%       col7  initial time step (s)
% row 2 col1  fault application time (s)
%       col2  bus number at which fault is applied
%       col3  bus number defining far end of faulted line
%       col4  zero sequence impedance in pu on system base
%       col5  negative sequence impedance in pu on system base
%       col6  type of fault  - 0 three phase
%                            - 1 line to ground
%                            - 2 line-to-line to ground
%                            - 3 line-to-line
%                            - 4 loss of line with no fault
%                            - 5 loss of load at bus
%                            - 6 no action
%       col7  time step for fault period (s)
% row 3 col1  near end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for second part of fault (s)
% row 4 col1  far end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for fault cleared simulation (s)
% row 5 col1  time to change step length (s)
```

```
%       col7  time step (s)
%
%
%
% row n col1 finishing time (s)  (n indicates that intermediate rows may be inserted)
sw_con = [...
0    0    0    0    0    0    0.01;%sets intitial time step
0.1  3    101  0    0    0    0.01; %3 ph to ground fault
0.15 0    0    0    0    0    0.01; %clear near end
0.20 0    0    0    0    0    0.01; %clear remote end
%0.50 0    0    0    0    0    0.01; % increase time step
%1.0  0    0    0    0    0    0.01; % increase time step
10.0  0    0    0    0    0    0]; % end simulation

%ibus_con = [0 1 1 1];
```

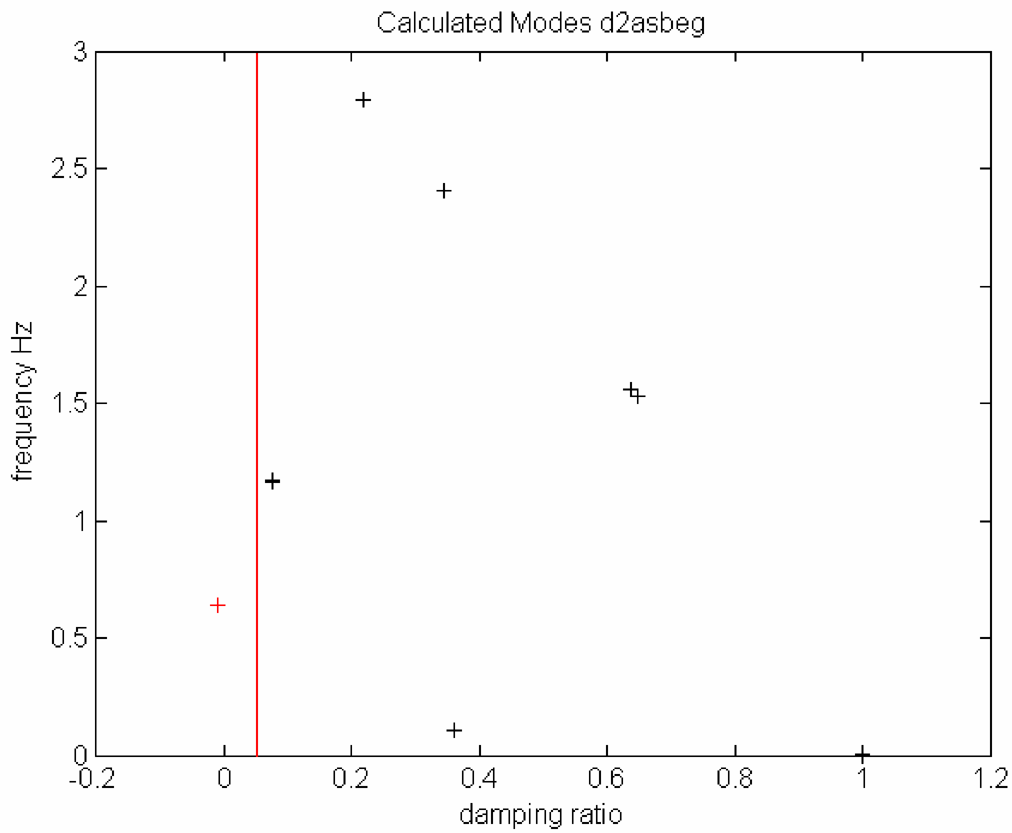


Figure 20 Modes of two-area system with exciters and governors on all units

The inter-area mode is unstable +.

To design a power system stabilizer, a system model with the generator rotor states removed is required. The input to the system is the voltage reference of the generator at which the power system stabilizer is to be placed. The output is the generator electrical power.

```
a=a_mat; b = b_vr(:,1); c=c_p(1,:); d=0;
ang_idx = find(mac_state(:,2)==1)
ang_idx =
     1
    12
    23
    34
spd_idx = ang_idx+1;
rot_idx = sort([ang_idx;spd_idx])
rot_idx =
     1
     2
    12
    13
    23
    24
    34
    35
a(rot_idx,:)=[];a(:,rot_idx)=[];
b(rot_idx)=[];
c(rot_idx)=[];
spssd = stsp(a,b,c,d);
```

The ideal power system stabilizer phase lead is given by the negative of the response of spssd. This is obtained using

```
f = linspace(.1, 2,100);
[f,ympd,yapd]=fr_stsp(spssd,f);
plot(f,-yapd)
```

The plot is shown in Figure 21.

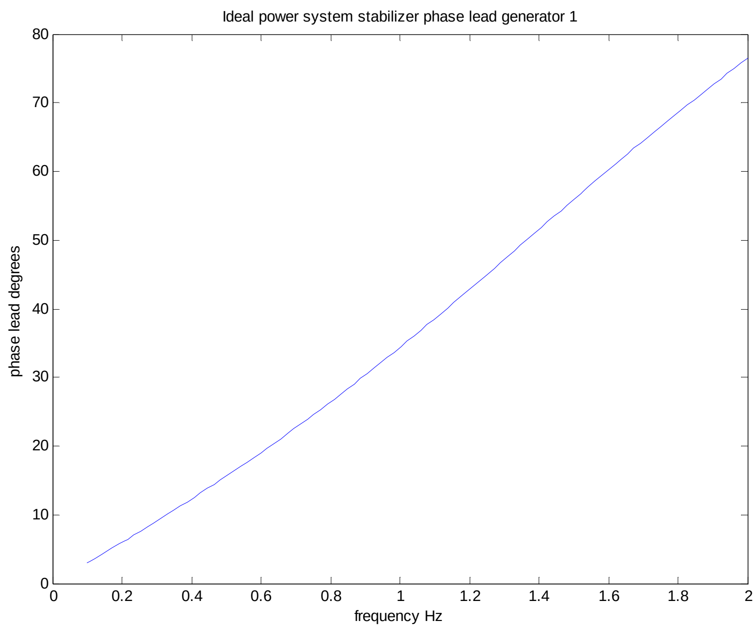


Figure 21 Ideal power system stabilizer phase lead

The power system stabilizer has the form

$$spss = \frac{sT_{wo}}{1+sT_{wo}}\left(\frac{1+sT_1}{1+sT_2}\right)^2$$

A state space model may be otained using
```
spss1 = wo_stsp(10).*ldlg_stsp(1,.02,.07).*ldlg_stsp(1,.02,.07);
```

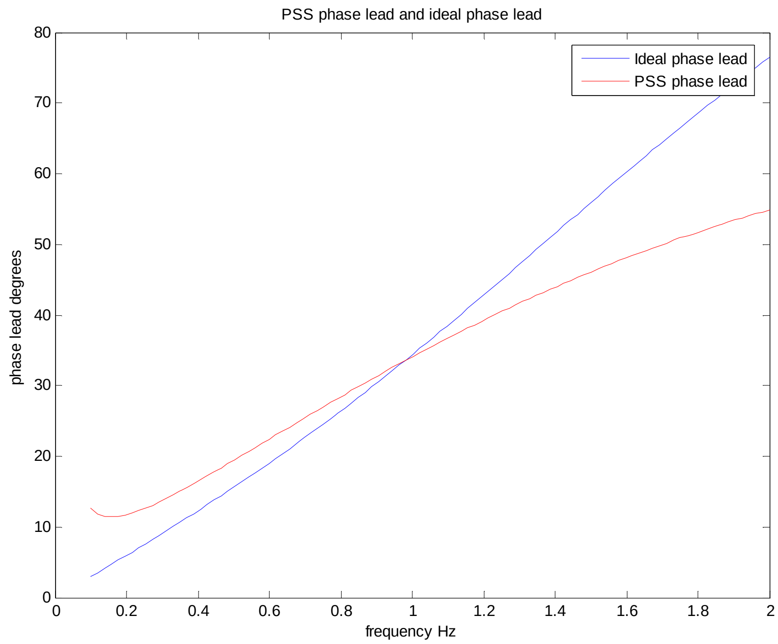


Figure 22 Ideal and PSS phase lead

Figure 22 shows that the power system stabilizer phase lead and the ideal phase lead are sufficiently close.

A root locus with gain of the PSS is obtained using
```
figure
plot(l,'k+')
hold
Current plot held
plot(lz,'ko')
plot(rlpss,'k.')
axis([-10 1 0 20])
grid
plot(l,'k+')
plot(rlpss,'k.')
dr_plot(0,20,0.05,'k');
grid
plot(rlpss(:,10),'r*')
```

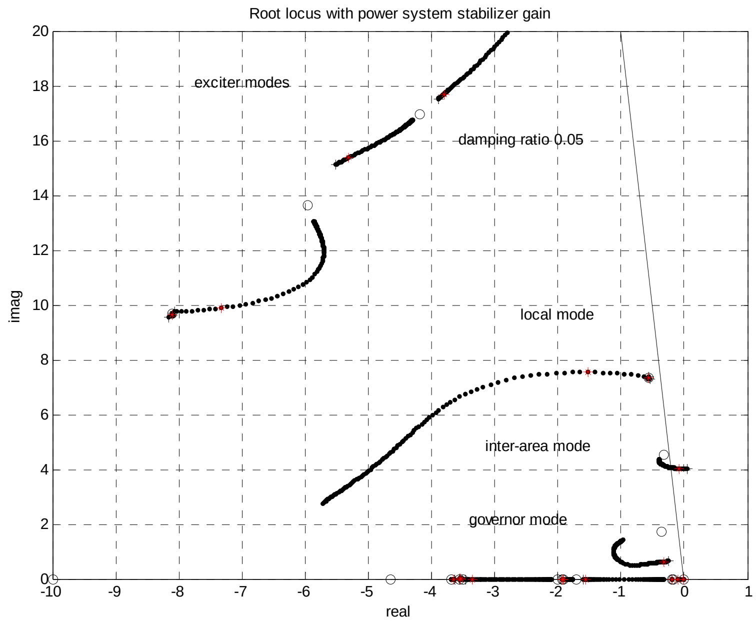


Figure 23 Root locus with PSS gain

Note: Any linear stabilizer design, should be checked for robustness using a transient stability simulation under a wide range of operating conditions. It is normal to set the PSS output limits so that the stabilizer has no adverse effects on a generator's response to a fault. Generally, the lower the negative output limit, the more effect the PSS has on the terminal voltage recovery following a fault.

2.8 References

1. R.P. Schulz, "Synchronous Machine Modeling," presented at the *Symposium "Adequacy and Philosophy of Modeling: System Dynamic Performance*," San Francisco, July 1972.
2. IEEE Committee Report, "Excitation System Models for Power System Stability Studies," *IEEE Transactions of Power Apparatus and Systems*, vol. PAS-100, pp. 494-509, 1981.
3. E.V. Larsen and J. H. Chow, "SVC Control Concepts for System Dynamic Performance," in *Application of Static VAR Systems for System Dynamic Performance*, IEEE Publications 87TH0187-5-PWR, 1987.
4. W.L. Brogan, *Modern Control Theory*, Quantum Publishers, New York, 1974.
5. J.H. Chow, editor, *Time-Scale Modeling of Dynamic Networks with Applications to Power Systems*, Springer-Verlag, Berlin, 1982.
6. V. Vittal, "Transient Stability Test Systems for Direct Stability Methods," IEEE Committee Report, IEEE Winter Power Meeting, Paper 91 WM 224-6 PWRS, 1991.
7. Graham Rogers and Joe Chow, " Hands-On Teaching of Power System Dynamics" *IEEE Computer Applications in Power*, January 1995, pp 12-16.
8. Graham Rogers, ‘Power System Oscillations’, Kluwer Academic Press, Boston, 1999.

3 Function Descriptions

The following contains descriptions of all of each function in the Power System Toolbox.

3.1 calc

3.1.1 Purpose:

Calculates the load flow power mismatch and checks convergence. Determines updated values of active and reactive power.

3.1.2 Syntax:

```
[delP,delQ,P,Q,conv_flag] = ...
         calc(nbus,V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol)
```

3.2 cdps

3.2.1 Purpose:

Changes the current directory

3.2.2 Syntax:

```
p = cdps(dir)
```

3.2.3 Purpose:

Changes the current MATLAB© directory. Used with user defined control models.

3.2.4 Input

dir the required new directory
p the full path of the new directory

3.2.5 Example

```
cd % this gives the current directory
C:\Program Files\MATLAB\R2007a\PSTV2\pstdat        %MATLAB OUTPUT
p=cdps('chapter10')% this changes the directory
p =                                                %MATLAB OUTPUT
C:\Program Files\MATLAB\R2007a\PSTV2\chapter10     %MATLAB OUTPUT
cd % this gives the new directory
C:\Program Files\MATLAB\R2007a\PSTV2\chapter10     %MATLAB OUTPUT
```

Note: this m-file should be modified to contain the users PST directory

3.3 chq_lim

3.3.1 Purpose:

Checks for generator reactive power outside limits

3.3.2 Syntax:

```
chq_lim(qg_max,qg_min)
```

3.4 dbcage

3.4.1 Purpose:

Calculates the equivalent single cage resistance and reactance of a double cage induction motor as a function of slip.

3.4.2 Syntax:

```
[r,x]=dbcage(r1,x1,r2,x2,s)
```

3.4.3 Inputs:

r1 the first cage resistance (PU on motor base)
x1 the first cage leakage reactance (PU on motor base)
r2 the second cage resitance (PU on motor base)
x2 the inter-cage reactance (PU on motor base)
s the motor slip

3.4.4 Outputs:

r the equivalent rotor resistance at slip s (PU on motor base)
x the equivalent rotor leakage reactance at slip s (PU on motor base)

3.4.5 Algorithm:
The rotor impedance is calculated at slip s and its real and imaginary parts used to define the the equivalent rotor resistance and reactance.

$$z = ix1 + (r1/s)(r2/s + ix2)/((r1 + r2)/s + ix2)$$
$$r = real(z); x = imag(z)$$

3.5 deepbar
3.5.1 Purpose:
Calculates the equivalent single cage resistance and reactance of a deep bar induction motor as a function of slip.
3.5.2 Syntax:
`[r,x]=deepbar(rro,B,s)`
3.5.3 Inputs:
rro the resistance of the rotor bar at zero slip (PU on motor base)
B the deep bar factor
s the motor slip
3.5.4 Outputs:
r the equivalent rotor resistance at slip s (PU on motor base)
x the equivalent rotor leakage reactance at slip s (PU on motor base)
3.5.5 Algorithm:
The equivalent rotor resistance and reactance as a function of slip is

$$b = B\sqrt{|s|};$$
$$r_o = rro/2;$$
$$a = (1+i)b;$$
$$z = r_o a(exp(a)+1)./(exp(a)-1);$$
$$r = real(z);x=imag(z)./s;$$

Where B is the deep bar factor which depends on the depth of the rotor bar,

$$B = d\sqrt{2\omega\mu_o\sigma}$$

and,
ω is the angular frequency of the motor supply
μ_o is the permeability of free space
σ is the conductivity of the rotor bar
3.6 dc_cont
3.6.1 Purpose:
Models the action of HVDC link pole controllers in dynamic simulation
3.6.2 Syntax:
`dc_cont(i,k,bus,flag)`
3.6.3 Description:
dc_cont contains the equations required for the initialization, network interface and rate of change of state evaluation for the rectifier and inverter controls of HVDC links.
3.6.4 Inputs:
i **i** = 0 all HVDC computations are performed using MATLAB vector methods
k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.
bus the solved bus specification matrix
flag indicates the mode of solution
- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.

- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.6.5 Output:

No output is used – the variables are passed to the anaysis programs as global variables

3.6.6 Global Variables:

basmva - system base MVA
dcsp_con - converter specification matrix
dcl_con - HVDC line specification matrix
dcc_con - HVDC pole control specification matrix
r_idx - rectifier index
i_idx - inverter index
n_dcl - number of HVDC lines
n_conv - number of HVDC converters
ac_bus - index of converter ac buses in the internal bus list
rec_ac_bus - index of rectifier ac buses in the internal bus list
inv_ac_bus - index of inverter ac buses in the internal bus list
Vdc - Matrix of HVDC voltages kV
i_dc - Matrix of HVDC line currents kA
dc_pot -
alpha - matrix of rectifier firing angles
gamma - matrix of inverter extinction angles
Vdc_ref - reference value for inverter extinction angle control
cur_ord - reference for current control at rectifier and inverter
dc_sig - external modulation control signal at rectifier and inverter
dcc_pot - matrix of pole control constants
i_dcr - rectifier line current kA
i_dci - inverter line current kA
v_dcc - HVDC line capacitance voltage kV
di_dcr - rate of change of rectifier HVDC line current
di_dci - rate of change of inverter HVDC line current
dv_dcc - rate of change of HVDC line capacitor voltage
v_conr - rectifier integral control state
dv_conr - rate of change of rectifier control state
v_coni - inverter integral control state
dv_coni - rate of change of inverter control stateData Format:
The pole control data is specified in the matrix **dcc_con**

Table 1 HVDC Control Format

Column	Variable
1	Converter number
2	Proportional Gain
3	Integral Gain
4	Output Gain
5	Maximum Integral Limit
6	Minimum Integral Limit
7	Maximum Output Limit
8	Minimum Output Limit
9	Control Type

Note: the order of the converters in **dcc_con** must be the same as that in **dcsp_con**.

3.6.7 Algorithm:

Figure 24 shows the rectifier pole control block diagram. The control of the rectifier firing angle is by means of a proportional plus integral controller used to keep the HVDC line current at a value specified by **cur_ord**. Figure 25 shows the inverter pole control block diagram. The control of the inverter extinction angle is by means of a proportional plus integral controller used to keep the inverter HVDC voltage at its initial value. If the inverter current falls below the inverter current order, the inverter pole control will take over current control.

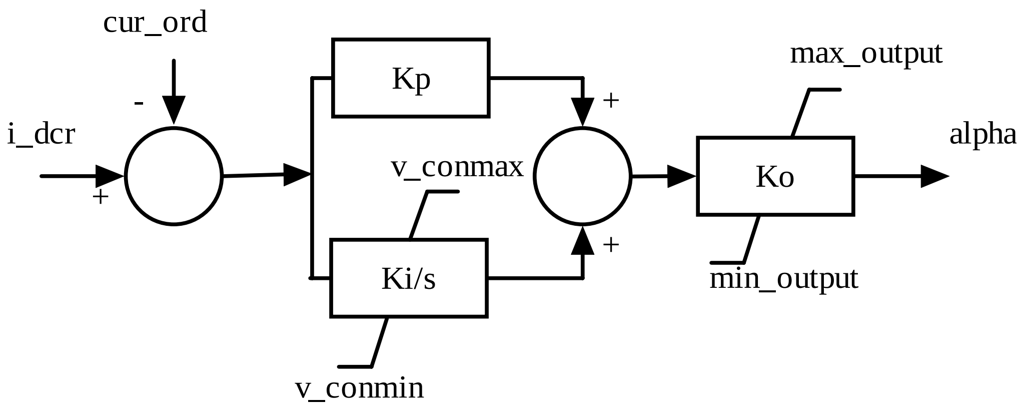


Figure 24 Rectifier Control Block Diagram

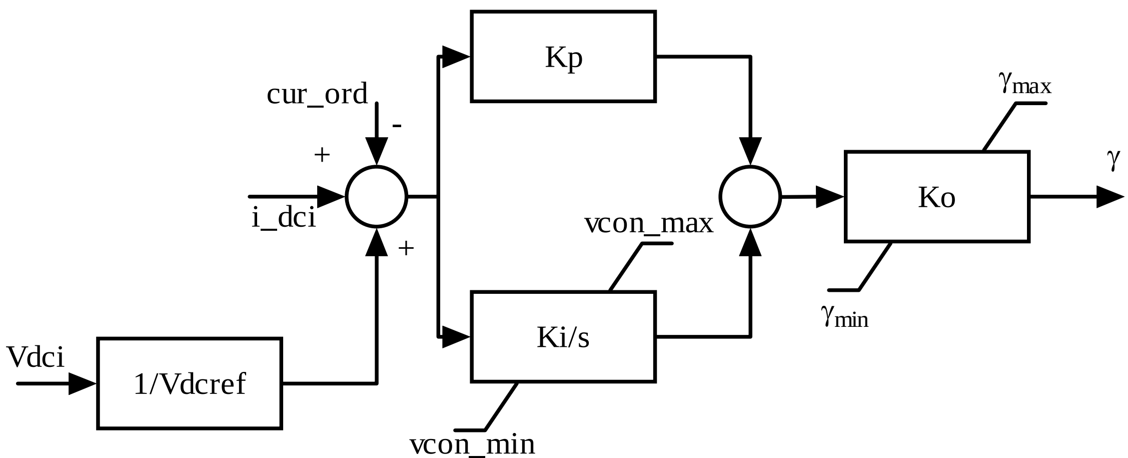


Figure 25 Inverter Pole Control Block Diagram

This algorithm is implemented in the M-file **dc_cont** in the POWER SYSTEM TOOLBOX.

3.7 dc_cur

3.7.1 Purpose:

Calculates the ac current load for use in the non-conforming load function **nc_load**

3.7.2 Syntax:

```
i_ac = dc_cur(V,k)
```

3.7.3 Description:

The function uses the current HT voltage estimate to determine the ac current load due to the HVDC links.

3.7.4 Inputs:

V - the current value of the equivalent HVDC HT terminal voltage
k - the current time step

3.7.5 Outputs:
i_ac - the ac load current in per unit due to the HVDC links

3.7.6 Global Variables:
r_idx - rectifier index
i_idx - inverter index
dcc_pot - dc control constant matrix
n_dcl - number of HVDC lines
basmva - system base MVA
i_dcr - rectifier current kA
i_dci - inverter current kA
alpha - rectifier firing angle
gamma - inverter extinction angle

3.7.7 Algorithm:
Calculates the HVDC voltages assuming that the currents, firing angle and extinction angle are constant. Calculates the equivalent active and reactive power load at the HVDC HT bus and from this calculates the equivalent alternating currents.
This algorithm is implemented in the M-file **dc_cur** in the POWER SYSTEM TOOLBOX.

3.8 dc_line

3.8.1 Purpose:
Forms the equations for HVDC line dynamics.

3.8.2 Syntax:
```
dc_line(i,k,bus,flag)
```

3.8.3 Description:
dc_line contains the equations necessary to model an HVDC line dynamically.

3.8.4 Inputs:
i — **i** = 0 all HVDC computations are performed using MATLAB vector methods
k — the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.
bus — the solved bus specification matrix
flag — indicates the mode of solution
- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.8.5 Global Variables:
dcsp_con - HVDC converter specification matrix
dcl_con - HVDC line specification matrix
dcc_con - converter control specification matrix
dcc_pot - converter control constants matrix
dc_pot - line constants matrix
r_idx - rectifier index
i_idx - inverter index
n_dcl - number of HVDC lines
n_conv - number of HVDC converters
Vdc - HVDC voltages kV
i_dc - HVDC currents kA
no_cap_idx - index of HVDC lines with no capacitance specified
cap_idx - index of HVDC lines with capacitance specified

no_ind_idx - index of HVDC lines with no inductance specified
l_no_cap - number of HVDC lines with no capacitance
l_cap - number of HVDC lines with capacitance
i_dcr - rectifier HVDC line current kA
i_dci - inverter HVDC line current kA
v_dcc - HVDC line capacitance voltage kV
di_dcr - rate of change of rectifier dc line current
di_dci - rate of change of inverter dc line current
dv_dcc - rate of change of dc line capacitance voltage

3.8.6 Algorithm:

The HVDC line is modelled as a T equivalent. The smoothing reactors are included. The capacitance of the line may be set to zero. In this case, the inverter current is always equal to the rectifier current.

3.9 dc_load

3.9.1 Purpose:

Calculates the non-linear Jacobian elements for the changes in ac current injection changes in the real and imaginary parts of the equivalent HT terminal voltage.

3.9.2 Syntax:

```
[Yrr,Yri,Yir,Yii] = dc_load(V,k)
```

3.9.3 Description:

Calculates:

$$Y_{rr} = \frac{\partial i_{acr}}{\partial V_r}; Y_{ri} = \frac{\partial i_{acr}}{\partial V_i}; Y_{ir} = \frac{\partial i_{aci}}{\partial V_r}; Y_{ii} = \frac{\partial i_{aci}}{\partial V_i}$$

3.9.4 Inputs:

V - the equivalent HT bus voltage
k - the current time step

3.9.5 Outputs:

$\mathbf{Y_{rr}}$, $\mathbf{Y_{ri}}$, $\mathbf{Y_{ir}}$, $\mathbf{Y_{ii}}$

3.9.6 Global Variables:

i_dci - the inverter dc current
i_dcr - the rectifier dc current
dcc_pot - the dc control constants
alpha - the rectifier firing angle
gamma - the inverter extinction angle
basmva - the system base MVA
r_idx - the rectifier index
i_idx - the inverter index
n_conv - the number of HVDC converter buses
n_dcl - the number of HVDC lines

3.9.7 Algorithm:

This algorithm is implemented in the M-file **dc_load** in the POWER SYSTEM TOOLBOX.

3.10 desat

3.10.1 Purpose:

Calculates the describing function for saturation

3.10.2 Syntax:

```
g = dessat(a,isat)
```

3.10.3 Inputs:

a the input amplitude
isat the saturation amplitude

3.10.4 Outputs:

g the ratio of the amplitude of the fundamental of a sine wave of amplitude a clipped at isat to a

3.10.5 Algorithm:

The fundamental of the clipped sine wave is calculated from

$$g = \frac{2}{\pi}(\tan^{-1} y + 0.5*\sin(2y)) \quad k \le 1$$
$$= 1 \qquad\qquad k > 1$$

where

$$k = \left|\frac{isat}{a}\right|; y = \frac{k}{\sqrt{1-k^2}}$$

The leakage inductances for the stator and rotor of an induction motor are calculated as

$$x_{sat} = x_{unsat}(1+g)/2$$

Called by: mac_ind

3.11 exc_dc12

3.11.1 Purpose:

Models IEEE Type DC1 and DC2 excitation system models

3.11.2 Synopsis:

`exc_dc12(i,k,bus,flag)`

3.11.3 Description:

`exc_dc12(i,k,bus,flag)` contains the equations of IEEE Type DC1 and DC2 excitation system models[1] for the initialization, machine interface and dynamics computation of the **ith** excitation system.

3.11.4 Inputs:

i the number of the exciter
if **i** = 0 all dc exciters computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.11.5 Global Variables

Efd	E_{fd}	excitation output voltage (= field voltage) in pu
V_R	V_R	regulator output voltage in pu
V_A	V_A	regulator output voltage in pu
V_As	V_{As}	regulator voltage state variable in pu
R_f	R_f	stabilizing transformer state variable
V_FB	V_{FB}	feedback from stabilizing transformer
V_TR	V_{TR}	voltage transducer output in pu
V_B	V_B	potential circuit voltage output in pu
dEfd	dE_{fd}/dt	
dV_R	dV_R/dt	
dV_As	dV_{As}/dt	
dR_f	dR_f/dt	
dV_TR	dV_{TR}/dt	

exc_sig V_{sup} supplementary signal input to the summing junction
exc_pot internally set matrix of exciter constants
exc_con matrix of exciter data supplied by user
The m.file **pst_var.m** contains all the global variables required for **exc_dc12**, and should be loaded in the program calling **exc_dc12**.

3.11.6 Data Format

The exciter data is contained in the **i**$^{\mathbf{th}}$ row of the matrix variable **exc_con**. The data format for **exc_dc12** is shown in Table 2.
A constraint on using **exc_dc12** is that $T_F \neq 0$. All other time constants can be set to zero. If T_E is set to zero, then $E_{fd} = V_R$. K_F can be set to zero to model simple first order exciter models. The state V_R is prevented from exceeding its limits by a non_wind up limit.
If K_E is set to zero on input, its value will be computed during initialization to make $V_R=0$. If V_{Rmax} is set to zero on input, the values of V_{Rmax} and V_{Rmin} will be computed assuming that E_2 is the nominal ceiling value of E_{fd}.

Table 2 **Data Format for exc_dc12**

column	variable	unit
1	exciter type 1 for DC1 2 for DC2	
2	machine number	
3	input filter time constant T_R	sec
4	voltage regulator gain K_A	
5	voltage regulator time constant T_A	sec
6	voltage regulator time constant T_B	sec
7	voltage regulator time constant T_C	sec
8	max voltage regulator output V_{Rmax}	pu
9	min voltage regulator output V_{Rmin}	pu
10	exciter constant K_E	
11	exciter time constant T_E	sec
12	E_1	pu
13	saturation function $S_E(E_1)$	
14	E_2	pu
15	saturation function $S_E(E_2)$	
16	stabilizer gain K_F	
17	stabilizer time constant T_F	sec

3.11.7 Algorithm:

Based on the exciter block diagram, the exciter is initialized using the generator field voltage E_{fd} to compute the state variables. In the network interface computation, the exciter output voltage is converted to the field voltage of the synchronous machine. In the dynamics calculation, generator terminal voltage and the external signal is used to calculate the rates of change of the excitation system states.
This algorithm is implemented in the M-file **exc_dc12.m** in the POWER SYSTEM TOOLBOX.
See also: loadflow, pst_var, smpexc, exc_st3, mac_tra, mac_sub.

3.11.8 Reference:

1. IEEE Committee Report, "Excitation System Models for Power System Stability Studies," *IEEE Transactions of Power Apparatus and Systems*, vol. PAS-100, pp. 494-509, 1981.

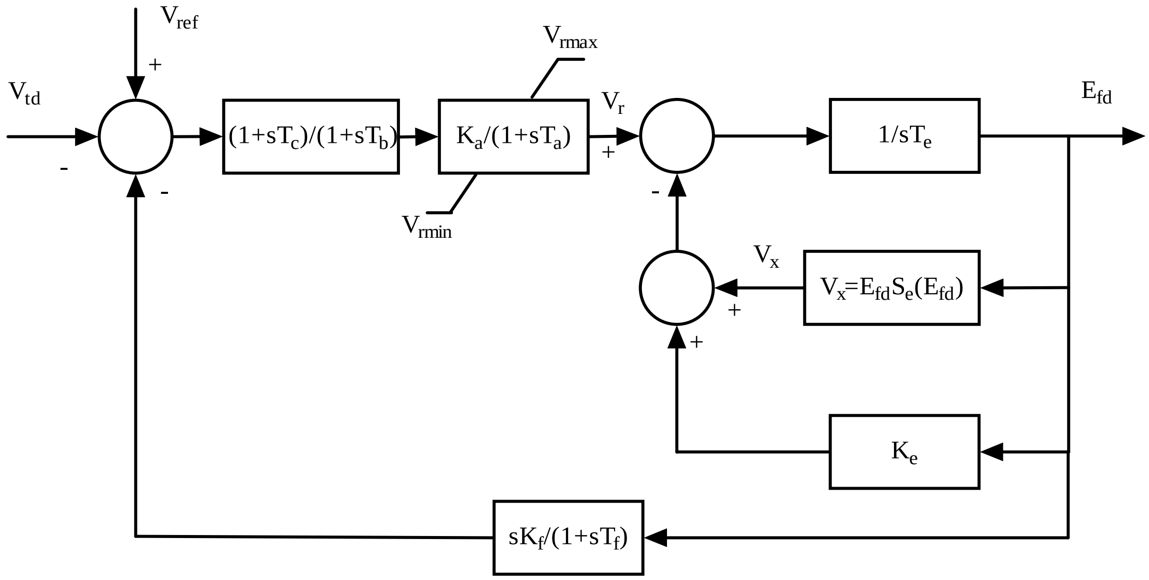


Figure 26 DC Exciter Type 1

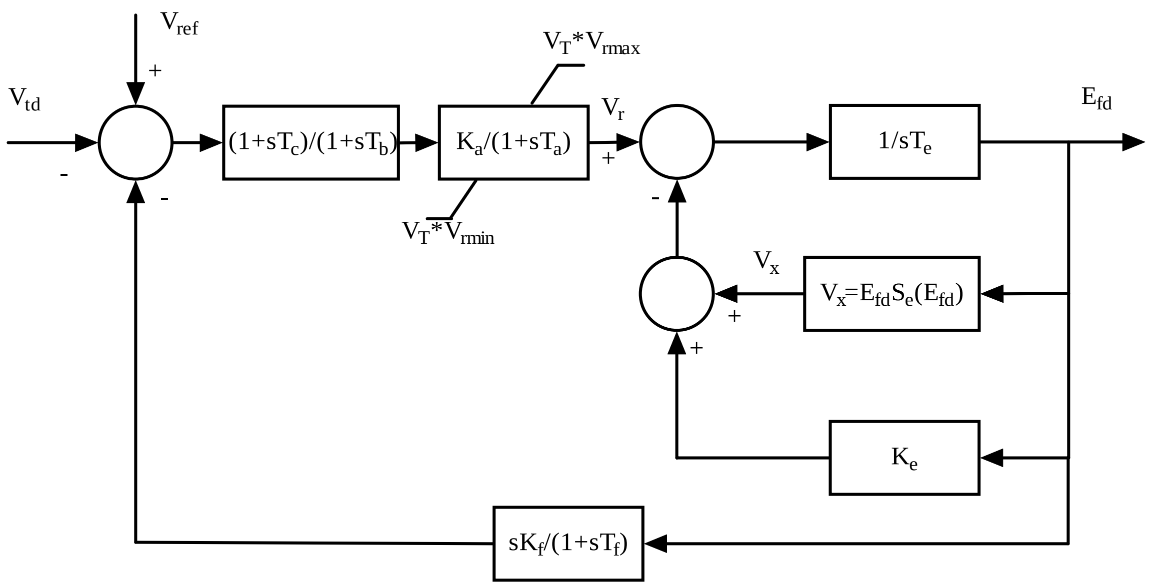


Figure 27 DC Exciter Type 2

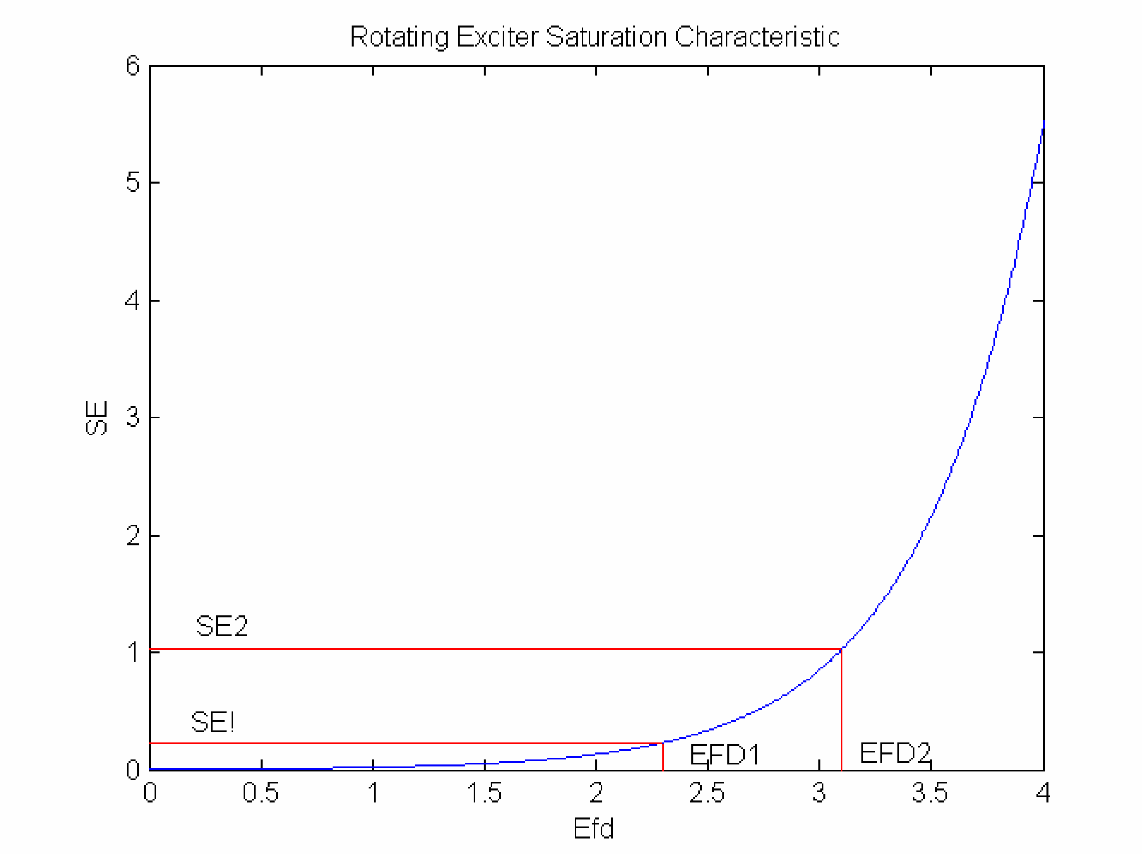


Figure 28 Exciter Saturation Function

3.12 exc_indx

3.12.1 Purpose:

Forms indexes for the exciters to enable vector computation to be used with mixed exciter models.

3.12.2 Syntax:

`exc_indx`

3.12.3 Description:

Checks the exciter input matrix **exc_con** for the type of exciter and the parameters specified. It produces indexes for the various exciter types and their parameters which are used in the corresponding model functions.

3.12.4 Global Variables

3.12.4.1 Exciter Indexes

exc_pot - exciter constants calculated on ititialization
exc_con - exciter data specification matrix
n_exc - number of exciters
smp_idx - index of simple exciters
n_smp - number of simple exciters
dc_idx - index of dc exciters
n_dc - number of dc exciters
dc2_idx - index of type 2 dc exciters
n_dc2 - number of type 2 dc exciters
st3_idx - index of st3 exciters
n_st3 - number of st3 exciters

3.12.4.2 Variable Indexes

smp_TA - the value of T_A for simple exciters (exc_con(smp_idx,5))
smp_TA_idx - the index of simple exciters having a $T_A > 0.01$s
smp_noTA_idx - the index of simple exciters having a $T_A < 0.01$s
smp_TB - the value of T_B for simple exciters
smp_TB_idx - the index of simple exciters having a $T_B > 0.01$s
smp_noTB_idx - the index of simple exciters having a $T_B < 0.01$s
smp_TR - the value of T_R for simple exciters
smp_TR_idx - the index of simple exciters having a $T_R > 0.01$s exciters
smp_noTR_idx - the index of simple exciters having a $T_R < 0.01$s
dc_TA - the value of T_A for dc exciters (exc_con(dc_idx,5))
dc_TA_idx - the index of dc exciters having a $T_A > 0.01$s
dc_noTA_idx - the index of dc exciters having a $T_A < 0.01$s
dc_TB - the value of T_B for dc exciters
dc_TB_idx - the index of dc exciters having a $T_B > 0.01$s
dc_noTB_idx- the index of dc exciters having a $T_B < 0.01$s
dc_TE - the value of T_E for dc exciters
dc_TE_idx - the index of dc exciters having a $T_E > 0.01$s
dc_noTE_idx - the index of dc exciters having a $T_E < 0.01$s
dc_TF - the value of T_F for dc exciters
dc_TF_idx - the index of dc exciters having a $T_F > 0.01$s
dc_TR - the value of T_R for dc exciters
dc_TR_idx - the index of dc exciters having a $T_R > 0.01$s
dc_noTR_idx - the index of dc exciters having a $T_R < 0.01$s
st3_TA - the value of T_A for st3 exciters
st3_TA_idx - the index of st3 exciters having a $T_A > 0.01$s
st3_noTA_idx - the index of st3 exciters having a $T_A < 0.01$s
st3_TB - the value of T_B for st3 exciters
st3_TB_idx - the index of st3 exciters having a $T_B > 0.01$s
st3_noTB_idx - the index of st3 exciters having a $T_B < 0.01$s
st3_TR - the value of T_R for st3 exciters
st3_TR_idx - the index of st3 exciters having a $T_R > 0.01$s
st3_noTR_idx - the index of st3 exciters having a $T_R < 0.01$s

3.12.5 Algorithm

This algorithm is implemented in the M-file **exc_indx.m** in the POWER SYSTEM TOOLBOX.

3.13 exc_st3

3.13.1 Purpose:

Models IEEE Type ST3 compound source rectifier exciter models

3.13.2 Synopsis:

```
exc_st3(i,k,bus,flag)
```

3.13.3 Description:

exc_st3(i,k,bus,flag) contains the equations of IEEE Type ST3 excitation system models [1] for the initialization, network interface and dynamics computation of the **i**th excitation system. The block diagram is shown in Figure 22.

The m.file **pst_var.m** containing all the global variables required for **exc_st3** should be loaded in the program calling **exc_st3**. The exciter data is contained in the **i**th row of the matrix variable **exc_con**.

3.13.4 Inputs:

i the number of the exciter
if **i** = 0 all st_3 exciter computations are performed using MATLAB vector methods.
This is the preferred mode.

k the integer time step in a simulation

In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.13.5 Global Variables:

3.13.5.1 System variables

psi_re	ψ_{re}	real and imaginary components of voltage
psi_im	ψ_{im}	source on system reference frame
cur_re	i_{re}	real and imaginary components of bus
cur_im	i_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.13.5.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu
eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	i_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base
vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech	P_m	mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user
n_mac		number of generators
n_em		number of em (classical) generator models

n_tra	number of transient generator models
n_sub	number of subtransient generator models
mac_tra_idx	index of transient generator models
mac_sub_idx	index of subtransient generator models

3.13.5.3 Exciter Variables

Efd	E_{fd}	exciter output voltage - generator field voltage pu
V_R	V_R	regulator output voltage in pu
V_A	V_A	regulator output voltage in pu
V_As	V_{As}	regulator voltage state variable in pu
R_f	R_f	stabilizing transformer state variable
V_FB	V_{FB}	feedback from stabilizing transformer
V_TR	V_{TR}	voltage transducer output in pu
V_B	V_B	potential circuit voltage output in pu
dEfd	dE_{fd}/dt	
dV_R	dV_R/dt	
dV_As	dV_{As}/dt	
dR_f	dR_f/dt	
dV_TR	dV_{TR}/dt	
exc_sig	V_{sup}	supplementary input signal to exciter ref input
exc_pot		matrix of internally set exciter constants
exc_con		matrix of exciter data set by user
st3_idx		index of st3 exciters
n_st3		number of st3 exciters
st3_TA		**exc_con(st3_idx,5)**
st3_TA_idx		index of nonzero TA for st3 exciter
st3_noTA_idx		index of zero TA for st3 exciter
st3_TB		**exc_con(st3_idx,6)**
st3_TB_idx		index of nonzero TB for st3 exciter
st3_noTB_idx		index of zero TB for st3 exciter
st3_TR		**exc_con(st3_idx,3)**
st3_TR_idx		index of nonzero TR for st3 exciter
st3_noTR_idx		index of zero TR for st3 exciter

3.13.6 Data Format:

The data format for **exc_st3** is given in Table 3. The time constants T_R and T_B can be set to zero if desired. However, T_A cannot be set to zero.

Table 3 ST3 Exciter Data Format

column	data	unit
1	exciter type	3 for ST3
2	machine number	
3	input filter time constant T_R	sec
4	voltage regulator gain K_A	
5	voltage regulator time constant T_A	sec
6	voltage regulator time constant T_B	sec

7	voltage regulator time constant T_C	sec
8	maximum voltage regulator output V_{Rmax}	pu
9	minimum voltage regulator output V_{Rmin}	pu
10	maximum internal signal V_{Imax}	pu
11	minimum internal signal V_{Imin}	pu
12	first state regulator gain K_J	
13	potential circuit gain coefficient K_P	
14	potential circuit phase angle q_P	degrees
15	current circuit gain coefficient K_I	
16	potential source reactance X_L	pu
17	rectifier loading factor K_C	
18	maximum field voltage E_{fdmax}	pu
19	inner loop feedback constant K_G	
20	maximum inner loop voltage feedback V_{Gmax}	pu

3.13.7 Example:

A typical data set for **st3** exciters is
exc_con = [...
3 1 0 7.04 0.4 6.67 1.0 7.57 0 0.2 -0.2 200 4.365 20 4.83 0.091 1.096 6.53 1 6.53];

3.13.8 Algorithm:

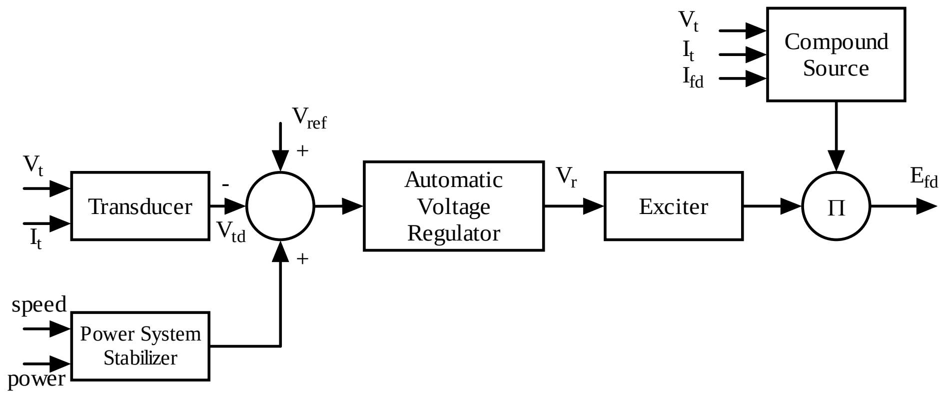
Based on the exciter block diagram, the exciter is initialized using the generator field voltage E_{fd} to compute the state variables. In the network interface computation, the exciter output voltage is converted to the field voltage of the synchronous machine. In the dynamics calculation, generator terminal voltage and the external signal is used to calculate the rates of change of the excitation system states.

This algorithm is implemented in the M-file **exc_st3.m** in the POWER SYSTEM TOOLBOX.

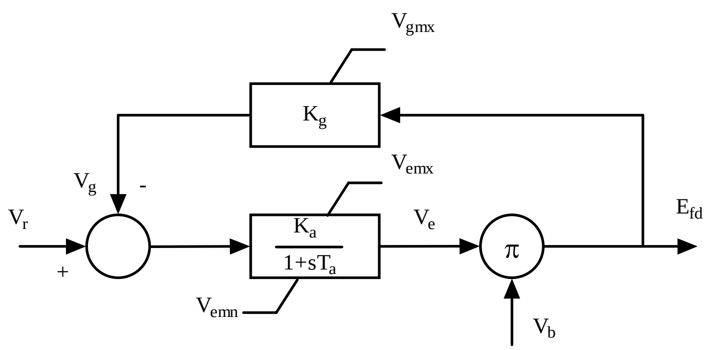
See also: loadflow, pst_var, exc_dc12, smpexc

3.13.9 Reference

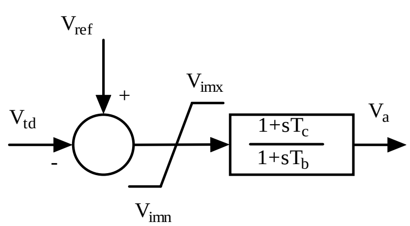
1. IEEE Committee Report, “Excitation System Models for Power System Stability Studies,” *IEEE Transactions of Power Apparatus and Systems*, vol. PAS-100, pp. 494-509, 1981.



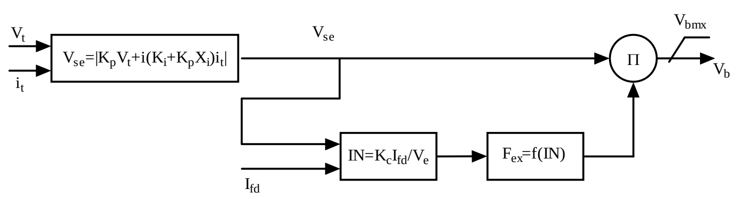
(a) System Block Diagram



(b) Exciter



(c) Automatic Voltage Regulator



(d) Compound Source

Figure 29 ST3 Excitation System

3.14 Imtspeed

3.14.1 Purpose:

Calculates the torque, power, reactive power and stator current as slip varies from 0 to 1.

3.14.2 Syntax:

[t,p,q,is,s]=imtspeed(V,rs,xs,Xm,rr,xr,rr2,xr2,dbf,isat)

3.14.3 Inputs:

V stator voltage magnitude PU on motor base

rs stator resistance PU on motor base

xs	stator leakage reactance PU on motor base
Xm	magnetizing reactance PU on motor base
rr	rotor reactance PU on motor base if double cage, the first cage resitance if deep bar the bar resistance at zero slip
xr	rotor leakage reactance PU on motor base if double cage, the leakage reactance of the first cage
rr2	the rotor resistance of the second cage PU on motor base zero if single cage or deep bar rotor
xr2	the rotor inter-cage leakage reactance PU on motor base zero if single cage or deep bar rotor
dbf	deep bar factor zero if motor single or double cage
isat	the current at which leakage inductance saturation occurs

3.14.4 Outputs:

t	torque
p	power
q	reactive power
is	stator current
s	slip

3.15 i_simu

3.15.1 Purpose:

To set the reduced Y matrix and the voltage recovery matrix to the appropriate values for the switching condition. Calculates the generator currents , the induction motor and generator currents and powers, the ac voltages (magnitudes and angles) and the HVDC voltages and currents.

3.15.2 Syntax:

```
h_sol = i_simu(k,ks,k_inc,h,ntot,bus_sim,Y_r,rec_V,bo)
```

3.15.3 Inputs:

ks - indicates the switching times
k - the current time step
k_inc - the number of time steps between switching points
h - vector of time steps
ntot - total number of machines (gen + motor)
bus_sim - value of bus matrix at this switching time
Y_r - reduced Y matrix at this switching time
rec_V - voltage recovery matrix at this switching time
bo - bus order for this switching time

3.15.4 Outputs:

h_sol - the time step at this value of k_s

3.15.5 Global variables:

psi_re - real part of generator internal bus voltage
psi_im - imaginary part of generator internal bus voltage
vdp - induction motor d axis voltage behind transient impedance
vqp - induction motor q axis voltage behind transient impedance
n_mot - number of induction motors
n_conv - number of HVDC converters
nload - number of non-conforming load buses
bus_int - internal bus number vector
cur_re - real part of generator current
cur_im - imaginary part of generator current
idmot - d axis motor current
iqmot - q axis motor current
p_mot - motor active power

q_mot - motor reactive power
idig - d axis induction generator current
iqig - q axis induction generator current
pig - induction generator active power
qig - induction generator ractive power

3.15.6 Algorithm:

This algorithm is implemented in the M-file **i_simu** in the POWER SYSTEM TOOLBOX.

3.16 line_pq

3.16.1 Purpose:

Line power flow computation

3.16.2 Synopsis:

```
[S1,S2] = line_pq(V1,V2,R,X,B,tap,phi)
```

3.16.3 Description:

line_pq(V1,V2,R,X,B,tap,phi) computes the power flow on transmission lines. with resistance **R**, reactance **X**, line charging **B**, tap ratio **tap** and phase shifter angle **phi** (in degrees). The voltages **V1** and **V2** describe the from and to bus voltages respectively. They may be vectors, or they may be a matrix, such as that obtained at the end of a transient simulation, i.e., **V1** may have the form
V1(1:number of buses, 1:number of time steps). The tap is at the from bus and represents the step down ratio, i.e., V1' = V1/(tap*exp(j*phi*pi/180)); and i1' = i1*tap*exp(j*phi*pi/180)
Note: V1 and **V2** must have the same size.

3.16.4 Inputs:

V1 from bus complex voltage matrix
V2 to bus complex voltage matrix
R line resistance vector
X line reactance vector
B line charging vector
tap tap ratio vector
phi phase shifter angle vector in degrees

3.16.5 Outputs:

S1 complex power injection matrix at from bus
S2 complex power injection matrix at to bus

3.16.6 Algorithm:

This algorithm is implemented in the M-file **line_pq** in the POWER SYSTEM TOOLBOX.

3.16.7 Example:

To calculate the complex power flow from the solved load flow

```
busnum = bus_sol(:,1)
FromBus = line(:,1)
ToBus = line(:,2)
From_idx = com_index(busnum,FromBus)
To_idx = com_index(busnum,ToBus)
V1 = bus_sol(From_idx,2);
V1ang = bus_sol(From_idx,3)*pi/180;
V1 = V1.*exp(j*V1ang);
V2 = bus_sol(To_idx,2);
V2ang = bus_sol(To_idx,3)*pi/180;
V2 = V2.*exp(j*V2ang);
R = line(:,3);
X = line(:,4);
B = line(:,5);
tap = line(:,6);
phi = line(:,7);
[V1 V2 R X B tap phi]
ans =
0.95781+0.32048i  0.97597+0.19995i   0      0.0167     0      1      0
1.0008+0.13576i   0.98457+0.016753i  0      0.0167     0      1      0
0.97085-0.12679i  0.98213-0.17727i   0      0.005      0      0.975  0
0.97085-0.12679i  0.98457+0.016753i  0.001  0.01       0.0175 1      0
```

```
0.97085-0.12679i  0.93357-0.35937i   0.011  0.11       0.1925 1       0
0.97085-0.12679i  0.93357-0.35937i   0.011  0.11       0.1925 1       0
0.97597+0.19995i  0.98457+0.016753i  0.0025 0.025      0.0437 1       0
0.99753-0.15822i  0.96088-0.26959i   0      0.0167     0       1      0
0.95428-0.33083i  0.88883-0.43064i   0      0.0167     0       1      0
0.81458-0.55074i  0.93357-0.35937i   0.011  0.11       0.1925 1       0
0.81458-0.55074i  0.93357-0.35937i   0.011  0.11       0.1925 1       0
0.81458-0.55074i  0.78173-0.63349i   0      0.005      0      0.9688 0
0.81458-0.55074i  0.88883-0.43064i   0.001  0.01       0.0175 1       0
0.96088-0.26959i  0.88883-0.43064i   0.0025 0.025      0.0437 1       0
```

Make the call:

```
[S1,S2] = line_pq(V1,V2,R,X,B,tap,phi)
ans =

       7.2611 +     1.2712i       -7.2611 -     0.38161i
            7 +      1.942i            -7 -      1.0781i
       9.2781 +       1.41i         -9.76 -          1i
      -13.922 +     1.8714i        14.128 +     0.17044i
        2.081 -    0.23939i        -2.031 +     0.55021i
        2.081 -    0.23939i        -2.031 +     0.55021i
       7.2611 +    0.38161i       -7.1279 +     0.90765i
            7 +     1.1342i            -7 -     0.31092i
            7 +     1.7631i            -7 -     0.90999i
      -1.9853 +    0.18247i         2.031 +    0.085233i
      -1.9853 +    0.18247i         2.031 +    0.085233i
       16.566 +     2.3872i        -17.65 -          1i
      -13.679 +     1.9259i        13.877 +     0.03125i
            7 +    0.31092i       -6.8767 +     0.87874i
```

To calculate the complex power flow from the transient simulation results

```
V1= bus_v(From_idx,:); V2= bus_v(To_idx,:);
[S1,S2] = line_pq(V1,V2,R,X,B,tap,phi);
subplot(2,1,1);plot(t,real(S1))
subplot(2,1,2);plot(t,imag(S1))
```

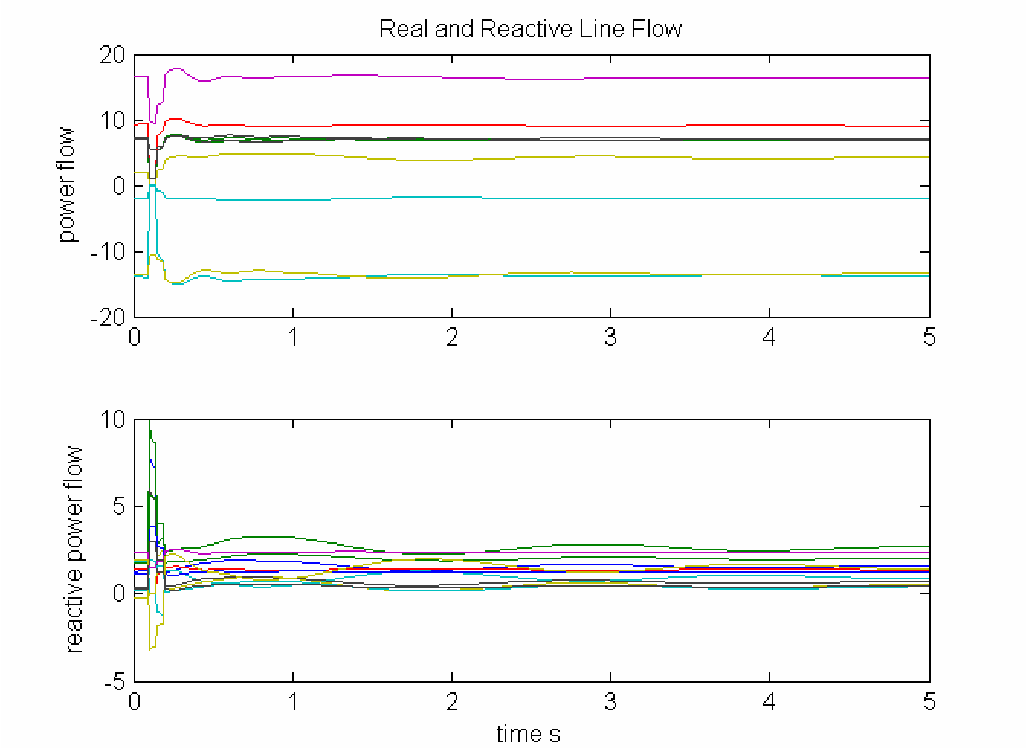


Figure 30 Real and Reactive Lineflow

3.17 lmod

3.17.1 Purpose:

A load modulation control for transient simulation

3.17.2 Synopsis:

`lmod(i,k,bus,flag)`

3.17.3 Description:

lmod contains the equations of a load modulation control system for the initialization, machine interface and dynamics computation of the **i**$^{\mathbf{th}}$ load modulation control.
Modulation is controlled through the global variable **lmod_sig**. This is modified by the function **ml_sig** which should be written by the user to obtain the required load modulation charactersitic.
The m.file **pst_var.m** containing all the global variables required for **lmod** should be loaded in the program calling **lmod**.

3.17.4 Inputs:

i — the number of the load modulation control
if **i** = 0 all load modulation computations are performed using MATLAB vector methods.
This is the preferred mode.

k — the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus — the solved bus specification matrix

flag — indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- There is no need to perform a network interface calculation for **lmod**
- The rates of change of the lmod state is calculated when **flag** = 2, using the modulating signal **lmod_sig** at the time specified by **k**

3.17.5 Global Variables

3.17.5.1 System variables

basmva — system base MVA
bus_int — array to store internal bus ordering

3.17.5.2 Load Modulation Variables

lmod_st	*lm*	load modulation state
dlmod_st	*dlm/dt*	
lmod_sig	V_{sup}	supplementary signal into the reference input
lmod_con		matrix of lmod parameters supplied by user
lmod_pot		internally calculated matrix of lmod constants
n_lmod		number of load modulation controls
lmod_idx		index of modulation controls included in **load_con**

3.17.6 Data Format

The load modulation control data is contained in the **i**$^{\mathbf{th}}$ row of the matrix **lmod_con**. The data format for **lmod_con** is given in Table 4.

Table 4 Data Format for Load Modulation Control

column	variable	unit
1	load modulation number	
2	bus number	
3	modulation base MVA	MVA
4	maximum conductance *lmod_max*	pu
5	minimum conductance *lmod_min*	pu
6	regulator gain K	pu
7	regulator time constant T_R	sec

3.17.7 Algorithm:

To use the **lmod** function, the load modulation buses must be declared via **load_con** as non-conforming load buses. The **lmod** buses may also have non-conforming loads In the network interface computation, the load modulation output is used to adjust the conductance at the control buses in the solution for the bus voltages in **nc_load**. In the dynamics calculation, the rate of change of the load modulation control state is adjusted according to the signal **lmod_sig**. An anti-windup limit is used to reset the state variable.

This algorithm is implemented in the M-file **lmod** in the POWER SYSTEM TOOLBOX.

See also: nc_load, pst_var, ml_sig.

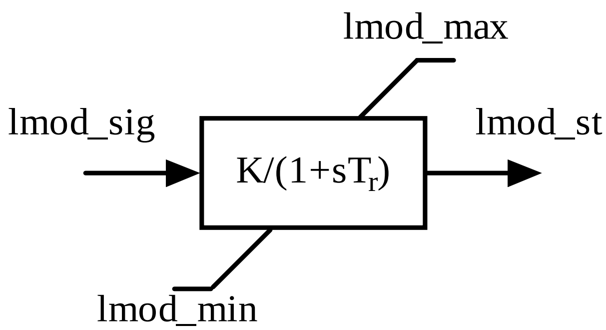


Figure 31 Load Modulation Control Block Diagram

3.18 mac_em

3.18.1 Purpose:

Model a synchronous machine with the classical electromechanical model

3.18.2 Synopsis:

mac_em(i,k,bus,flag)

3.18.3 Description:

mac_em(i,k,bus,flag) contains the electromechanical model equations for the initialization, network interface and dynamics computation of the **i**$^{\mathbf{th}}$ synchronous machine.

The m.file **pst_var.m** containing all the global variables required for **mac_em** should be loaded in the program calling **mac_em**.

3.18.4 Inputs:

i the number of the generator
if **i** = 0 all em generator computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the em generator states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.18.5 Global Variables:

3.18.5.1 System variables

psi_re	y_{re}	real and imaginary components of voltage
psi_im	y_{im}	source on system reference frame
cur_re	i_{re}	real and imaginary components of bus
cur_im	i_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.18.5.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu
eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	I_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base

vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech	P_m	mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user
n_mac		number of generators
n_em		number of em (classical) generator models
n_tra		number of transient generator models
n_sub		number of subtransient generator models
mac_tra_idx		index of transient generator models
mac_sub_idx		index of subtransient generator models

3.18.6 Data Format

The machine data is contained in the **i**$^{\mathbf{th}}$ row of the matrix variable **mac_con**. The data format for **mac_em** is shown in Table 5.

Table 5 Data for mac_em

column	variable	unit
1	machine number	
2	bus number	
3	base MVA	MVA
7	d-axis transient reactance x_d'	pu
16	Inertia Constant H	sec
17	damping coefficient d_o	pu
19	bus number	
22	active power fraction	
23	reactive power fraction	

Generators are numbered internally according to the order of the machines in **mac_con**. This information is contained in the array **mac_int** and is set up automatically by the Y matrix reduction function **red_ybus**.

3.18.7 Example:

The generator data in the 3 machine, 9 bus system [1] are
mac_con =[...

1	1	100	0	0	0	0.0608	0	0	0	0	0	0	0	0	23.64	9.6	0;	1
2	2	100	0	0	0	0.1198	0	0	0	0	0	0	0	0	6.4	2.5	0;	2
3	3	100	0	0	0	0.1813	0	0	0	0	0	0	0	0	3.01	1.0	0;]	3

Note : If the power fractions are left out of **mac_con**, they are automatically set to unity.

3.18.8 Algorithm:

Based on the generator vector diagram

- the initialization uses the solved loadflow bus voltages and angles to compute the internal voltage and the rotor angle. The d-axis voltage is identically zero for all time.
- the network interface computation generates the voltage behind the transient reactance on the system reference frame.
- in the dynamics calculation, the rotor torque imbalance and the speed deviation are used to compute the rates of change of the two state variables mac_ang and mac_spd.

This algorithm is implemented in the M-file **mac_em.m** in the POWER SYSTEM TOOLBOX.

See also: loadflow, mac_tra, mac_sub.

3.18.9 Reference:

1. J.H. Chow, editor, Time-Scale Modeling of Dynamic Networks with Applications to Power Systems, Springer-Verlag, Berlin, 1982.

3.19 mac_ib

3.19.1 Purpose:

Model a synchronous generator as an infinite bus

3.19.2 Synopsis:

```
mac_ib(i,k,bus,flag)
```

3.19.3 Description:

mac_ib(i,k,bus,flag) contains routines for the initialization, network interface and dynamics computation of the **i**$^{\mathbf{th}}$ synchronous machine modelled as an infinite bus.

The m.file **pst_var.m** containing all the global variables required for **mac_ib** should be loaded in the program calling **mac_ib**.

3.19.4 Inputs:

i the number of the generator
if **i** = 0 all em generator computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the em generator states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.19.5 Global Variables:

3.19.5.1 System variables

basmva		system base MVA
basrad		2 π* system frequency
sys_freq		system frequency in pu
bus_v	V	bus voltage magnitude in pu
bus_ang	θ	bus voltage angle in rad
psi_re	y_{re}	real and imaginary components of voltage
psi_im	y_{im}	source on system reference frame
cur_re	i_{re}	real and imaginary components of bus
cur_im	i_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.19.5.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu
eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	I_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base
vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech	P_m	mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user
ibus_con		vector specifying infinite buses set by user
n_ib		number of generators modelled as infinite buses
n_ib_em		number of em (classical) generators modeled as infinite buses
n_ib_tra		number of transient generators modeled as infinite buses
n_ib_sub		number of subtransient generators modeled as infinite buses
mac_ib_idx		index of generators modeled as infinite buses
not_ib_idx		index of generators not modelled as infinite buses

3.19.6 Data Format

The infinite buses are specified in the vector **ibus_con**. The vector is of length equal to the number of generators. It has zero entries for non-infinite bus generators and unity for infinite bus generators.

3.19.7 Example:

To represent generator 2 in the single generator infinite bus system as an infinite bus

```
ibus_con = [0 1]';
 mac_con = [
1 1 991     0.15 0  2.0    0.245 0.2  5.0  0.031 ...
                    1.91   0.42  0.2  0.66 0.061 ...
                    2.8756 0.0   0    1    0       0;
2 2 100000 0.00  0  0.     0.01  0    0    0     ...
                    0      0     0    0    0     ...
                    3.0    2.0   0    2    0       0];
```

3.19.8 Algorithm:
On initialization the internal voltage behind either transient or subtransient impedance is determined. Thereafter this voltage is maintained constant. This algorithm is implemented in the M-file **mac_ib.m** in the POWER SYSTEM TOOLBOX.

See also: loadflow, mac_em, mac_tra, mac_sub.

3.20 mac_igen

3.20.1 Purpose:
Models an induction generator

3.20.2 Synopsis:
```
[bus_new] = mac_igen(i,k,bus,flag)
```

3.20.3 Description:
mac_igen(i,k,bus,flag) contains the model equations for the initialization, network interface and dynamics computation of induction generators. The m.file **pst_var.m** containing all the global variables required for **mac_igen** should be loaded in the program calling **mac_igen**.

The induction generators are numbered internally according to the order of the machines in **igen_con**. This information is contained in the array **igen_int** and is set up automatically by the Y matrix reduction function **red_ybus**.

Note: The induction generator is modelled as a negative load in the loadflow, since induction generators cannot control voltage. The generator reactive power is not known until after the generator is initialized. After initialization, **bus_new** contains the load data with the generator active and reactive powers subtracted from the load specified in the original data file. This means that the induction generators must be initialized before the reduced y matrices are determined.

3.20.4 Inputs:
i = 0 ; **vector computation is the only option for induction generators**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the induction motor states are calculated when **flag** = 2, using the motor terminal voltage and the motor load torque at the time specified by **k**

3.20.5 Output:
bus_new a modified **bus** matrix, in which the induction generator active and reactive powers are subtracted from the original load active and reactive powers

3.20.6 Global Variables:

3.20.6.1 System Variables

basmva	system base MVA
basrad	2 π* system frequency
bus_int	array to store internal bus ordering

3.20.6.2 Induction Generator Variables

tmig	induction generator mechanical torque pu on motor base
pig	generator active power in p.u. on generator base
qig	generator reactive power in p.u. on generator base
vdig	generator direct axis stator voltage in p.u.
vqig	generator quadrature axis stator voltage in p.u.
idig	generator direct axis stator current in p.u.
iqig	generator quadrature axis voltage im p.u.

igen_con matrix of induction generator parameters set by user
igen_pot matrix of induction generator constants set internally
igen_int index of internal induction generator buses
igbus buses to which induction generators are connected
vdpig V'd direct axis transient voltage for induction generators (state)
vqpig V'q quadrature axis transient voltage for induction generators (state)
slig fractional slip (state)
dvdpig dV'_d/dt
dvqpig dV'_q/dt
dslig ds/dt

3.20.7 Data Format:

The induction generator data is contained in the **i^{th}** row of the matrix variable **igen_con**. The data format for **mac_igen** is shown in Table 6.

Table 6 Data for mac_igen

column	variable	unit
1	generator number	
2	bus number	
3	generator base MVA	MVA
4	stator resistance r_s	pu
5	stator leakage reactance x_s	pu
6	magnetizing reactance X_m	pu
7	rotor resistance r_r	pu
8	rotor leakage reactance x_r	pu
9	inertia constant H of generator plus turbine	sec
15	fraction of active bus load	

3.20.8 Example:

The induction generator data in the 3 machine, 9 bus system are

```
 igen_con =[...
1   8   60   0.001   0.01   3.   0.009   0.01   0.7   0   0   0   0   0   1];
```

3.20.9 Algorithm:

Initialization (flag = 0) uses the solved load flow bus voltages and angles to compute the slip required to generate the specified power. The power is specified as a fraction of the load at the specified load bus. This should be set to a negative value in the load flow specification matrix. The slip is calculated using a Newton Raphson iteration. Failure to converge within 30 iterations causes an error message to be generated. Once the initial slip is known, the generator's reactive power is calculated. The generator's real and reactive powers are then subtracted from the corresponding bus loads.

The dynamic model is that formulated by Brereton, Lewis and Young[1] for an induction motor. In this model the three states are the d and q voltages behind transient reactance and the slip. For an induction generator, the initial slip is negative

This algorithm is implemented in the M-file **mac_igen.m** in the POWER SYSTEM TOOLBOX.

See also: loadflow, mac_tra, mac_sub, mac_ind, red_ybus.

3.20.10 References

1. D.S. Brereton, D.G. Lewis and C.C. Young, " Representation of Induction Motor Loads during Power System Stability Studies", AIEE Trans, vol 76, Part III, August 1957, pp 451-460.

3.21 mac_ind

3.21.1 Purpose:

Models an induction motor.

3.21.2 Synopsis:

```
bus_new = mac_ind(i,k,bus,flag)
```

3.21.3 Description:

mac_ind(i,k,bus,flag) contains the model equations for the initialization, network interface and dynamics computation of induction motors.

The m.file **pst_var.m** containing all the global variables required for **mac_ind** should be loaded in the program calling **mac_ind**.

The induction motors are numbered internally according to the order of the machines in **ind_con**. This information is contained in the array **ind_int** and is set up automatically by the Y matrix reduction function **red_ybus**.

Note: The motor reactive power is not known until after the motor is initialized. After initialization, **bus_new** contains the load data with the motor real and reactive load powers subtracted from the load specified in the original data file. This means that the motors must be initialized before the reduced y matrices are determined.

3.21.4 Inputs:

i the number of the induction motor
if **i** = 0 all induction motor computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the induction motor states are calculated when **flag** = 2, using the motor terminal voltage and the motor load torque at the time specified by **k**

3.21.5 Output:

bus_new a modified **bus** matrix, in which the motor active and reactive powers are subtracted from the original load active and reactive powers

3.21.6 Global Variables:

3.21.6.1 System Variables

basmva	system base MVA
basrad	2 π* system frequency
bus_int	array to store internal bus ordering

3.21.6.2 Induction Motor Variables

tload	motor load torque
t_init	initial motor load torque in pu. on motor base
p_mot	motor active power in pu. on system base
q_mot	motor reactive power in pu. on system base
vdmot	motor direct axis stator voltage in pu.
vqmot	motor quadrature axis stator voltage in pu.
idmot	motor direct axis stator current in pu.
iqmot	motor quadrature axis voltage im pu.
ind_con	matrix of induction motor parameters set by user
ind_pot	matrix of induction motor constants set internally

ind_int index of internal induction motor buses
motbus buses to which induction motors are connected
vdp V'd direct axis transient voltage (state)
vqp V'q quadrature axis transient voltage (state)
slip fractional slip (state)
dvdp dV'_d/dt
dvqp dV'_q/dt
dslip ds/dt

Table 7 ind_pot variable definitions

Index Number	Variable
1	Scaled MVA base
2	Motor Base KV
3	$X_s = x_s + X_m$
4	$X_r = x_r + X_m$
5	$X_s^{'} = x_s + \frac{x_r X_m}{X_r}$
6	$X_s - X_s^{'}$
7	$1/T_r = \omega_0 r_r / X_r$

With deep bar and double cage motors, the ind_pot variables 3 to 7 vary with the motor slip, and are updated automatically during simulations. When leakage inductance saturation is specified, these variables change when the stator current exceeds the saturation current level.

3.21.7 Data Format:

The induction motor data is contained in the **ith** row of the matrix variable **ind_con**. The data format for **mac_ind** is shown in Table 8.

Table 8 ind_con data format

column	variable	unit
1	motor number	
2	bus number	
3	motor base MVA	MVA
4	stator resistance r_s	pu
5	stator leakage reactance x_s	pu
6	magnetizing reactance X_m	pu
7	rotor resistance r_r	pu
8	rotor leakage reactance x_r	pu
9	inertia constant H	Sec
10	second cage resistance r_2	pu
11	intercage reactance x_2	pu
12	deep bar ratio	pu
13	leakage saturation current	pu
15	fraction of active bus load	

If the fraction of active bus load is set to zero, the induction motor will be initialized as though disconnected from the network. The motor will connect as soon as a simulation is started.
The motor load is a function of speed as calculated in the m-file **ind_ldto**. Data associated with the load torque is specified using the matrix **mld_con**. Each row of **mld_con** represents the motors load/speed characteristic. Its form is shown in Table 9.

Table 9 mld_ con data format

column	variable	unit
1	motor number	
2	motor bus number	
3	stiction load coefficient - f_1	pu on motor base
4	stiction load index- i_1	
5	main load coefficient - f_2	pu on motor base
6	main load index - i_2	

The form of the motor load is as follows:

For a running motor the load torque is

$$t_l = \frac{t_{init}}{t_0}(f_1 s^{i_1} + f_2(1-s)^{i_2})$$

where

$$t_0 = f_1 s_0^{i_1} + f_2(1-s_0)^{i_2}$$ and s_0 is the initial slip

For a starting motor the load torque is

$$t_l = f_1 s^{i_1} + f_2(1-s)^{i_2}$$

Typical values are f_1=.1; i_1 = 1; f_2=.7; i_2=2

3.21.8 Example:

The induction motor data in the 3 machine, 9 bus system are

```
ind_con = [ ...
  1  7  25. .001 .01 3 .009 .01  2. 0 0 0 0 0 .15
  2  9  25. .001 .01 3 .009 .01  2. 0 0 0 0 0 .15
];

mld_con = [ ...
1  7  .1  1  .7  2
2  9  .1  1  .7  2
];
```

3.21.9 Algorithm:

Initialization (flag = 0) uses the solved load flow bus voltages and angles to compute the slip required for the motor to draw the specified power. The slip is calculated using a Newton Raphson iteration. Failure to converge within 30 iterations causes an error message to be generated. Once the initial slip is known, the motor's reactive power is calculated. The motor's real and reactive powers are then subtracted from the corresponding bus loads.

The dynamic model is that formulated by Brereton, Lewis and Young[1]. In this model the three states are the d and q voltages behind transient reactance and the motor slip.
If a double cage rotor is specified (non-zero values in columns 10 and 11 of ind_con), the effective rotor resistance and reactance (r_{re} and x_{re}) will vary with slip.

$z = ix_r + (r_r./s).*(r_2./s + i*x_2)./((r_r+r_2)./s + ix_2);$

$r_{re} = s.*real(z);$

$x_{re} = imag(z);$

If a deep bar rotor is specified (a non-zero value in column 12 of ind_con), the effective rotor resistance and reactance vary with slip

$$b = B\mathrm{sqrt}(\mathrm{abs}(s));$$
$$r_o = r_r/2;$$
$$a = (1+i)b;$$
$$z = r_o a[(\exp(a)+1)/(\exp(a)-1)];$$
$$r_e = \mathrm{real}(z); x_e = \mathrm{imag}(z)/s;$$

where B is the deep bar factor.
If leakage inductance saturation is specified, the stator and rotor leakage reactances vary according to the describing function for saturation. For the stator current greater than the saturation current

$$\theta = \mathrm{atan2}(i_{sat}, \sqrt{(i_s^2 - i_{sat}^2)})$$
$$g = (2/\pi)*(\theta + \sin(2\theta)/2)$$
$$x_{sn} = x_s g/2$$
$$x_{rn} = x_r g/2$$

This algorithm is implemented in the M-file **mac_ind.m** in the POWER SYSTEM TOOLBOX.
See also: loadflow, mac_tra, mac_sub, red_ybus.

3.21.10 Reference:

1. D.S. Brereton, D.G. Lewis and C.C. Young, "Representation of Induction Motor Loads during Power System Stability Studies", AIEE Trans., vol. 76, Part III, August 1957, pp 451-460.

3.22 mac_sub

3.22.1 Purpose:

Models a synchronous machine with the voltage behind subtransient reactance model

3.22.2 Synopsis:

```
mac_sub(i,k,bus,flag)
```

3.22.3 Description:

mac_sub contains the voltage behind the subtransient reactance model equations [1] for the initialization, network interface and dynamics computation of the **i**$^{\mathbf{th}}$ synchronous machine (see block diagram in Figure 1).
The m.file **pst_var.m** containing all the global variables required for **mac_sub** should be loaded in the program calling **mac_sub**.
The generators are numbered internally according to their order in **mac_con**. This information is contained in the array **mac_int** and is set up automatically by the Y matrix reduction function **red_ybus**.

3.22.4 Inputs:

i the number of the generator
if **i** = 0 all subtransient model generator computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and their rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution
- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the subtransient generator states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.22.5 Global Variables

3.22.5.1 System variables

basmva		system base MVA
basrad		2 π* system frequency
syn_ref		synchronous reference
mach_ref		reference machine
sys_freq		system frequency in pu
bus_v	V	bus voltage magnitude in pu
bus_ang	θ	bus voltage angle in rad
psi_re	ψ_{re}	real and imaginary components of voltage
psi_im	ψ_{im}	source on system reference frame
cur_re	i_{re}	real and imaginary components of bus
cur_im	i_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.22.5.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu
eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	I_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base
vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech	P_m	mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
dmac_ang	$d\delta/dt$	
dmac_spd	$d\omega/dt$	
deqprime	dE_q'/dt	
dedprime	dE_d'/dt	
dpsikd	$d\psi_{kd}/dt$	
dpsikq	$d\psi_{kq}/dt$	
mac_int		array to store internal machine ordering

mac_pot internally set matrix of machine constants

mac_pot(:,1) - System Base MVA/Generator Base MVA
mac_pot(:,2) - Base Voltage
mac_pot(:,3:5) - Saturation Model
mac_pot(:,6) - (xd-xd')(xd'-xd")/(xd'-xl)^2
mac_pot(:,7) - (xd-xl)(xd"-xl)/(xd'-xl)
mac_pot(:,8) - xd'-xl
mac_pot(:,9) - (xd"-xl)/(xd'-xl)
mac_pot(:,10) - (xd'-xd")/(xd'-xl)
mac_pot(:,11) - (xq-xq')(xq'-xq")/(xq'-xl)^2
mac_pot(:,12) - (xq-xl)(xq"-xl)/(xq'-xl)
mac_pot(:,13) - xq'-xl
mac_pot(:,14) - (xq"-xl)/(xq'-xl)
mac_pot(:,15) - (xq'-xq")/(xq'-xl)

mac_con matrix of generator parameters see Table 10
n_mac number of generators
n_sub number of subtransient generator models
mac_sub_idx index of subtransient generator models

3.22.6 Data Format

The data format for **mac_sub** is given in Table 10.
A constraint on using **mac_sub** is that $x_d" = x_q"$. This is because of the way in which the subtransient reactance is used in the network interface. **mac_sub** checks that the direct and quadrature subtransient reactances are equal, if they are unequal it makes them equal.
The definitions of the saturation factors are given in saturation curve diagram (Figure 34). It is assumed that there is no saturation for field current less than 0.8 pu. Setting the saturation factors to zero eliminates the saturation effect.

Table 10 Data format for mac_sub

column	variable	unit
1	machine number	
2	bus number	
3	base MVA	MVA
4	leakage reactance x_l	pu
5	resistance r_a	pu
6	d-axis synchronous reactance x_d	pu
7	d-axis transient reactance x_d'	pu
8	d-axis subtransient reactance x_d''	pu
9	d-axis open circuit time constant T_{do}'	sec
10	d-axis open circuit subtransient time constant T_{do}''	sec
11	q-axis synchronous reactance x_q	pu
12	q-axis transient reactance x_q'	pu
13	q-axis subtransient reactance x_q''	pu
14	q-axis open circuit time constant T_{qo}'	sec
15	q-axis open circuit subtransient time constant T_{qo}''	sec
16	Inertia constant H	sec
17	local damping coefficient d_o	pu
18	system damping coefficient d_1	pu
19	bus number	

20	saturation factor *S(1.0)*	
21	saturation factor *S(1.2)*	
22	active power fraction	
23	reactive power fraction	

3.22.7 Example:

The machine data of a single machine infinite bus system are

```
mac_con = [
1 1 991      0.15 0  2.0    0.245 0.2  5.0  0.031 ...
                     1.91   0.42  0.2  0.66 0.061 ...
                     2.8756 0.0   0    1    0       0;
2 2 100000 0.00  0   0.     0.01  0    0    0     ...
                     0      0     0    0    0     ...
                     3.0    2.0   0    2    0       0];
```

The first generator data is that for a subtransient model, the second data is that for an electromechanical generator model used to represent the infinite bus. In small signal stability simulations, the second generator should be declared as an infinite bus (see **mac_ib**).

3.22.8 Algorithm:

Based on the machine vector diagram

- the initialization uses the solved load flow bus voltages and angles to compute the internal voltage and the rotor angle.
- In the network interface computation, the voltage behind the subtransient reactance on the system reference frame is generated.
- In the dynamics calculation, the power imbalance and the speed deviation are used to compute the time derivative of the state variables.

This algorithm is implemented in the M-file **mac_sub** in the POWER SYSTEM TOOLBOX.

See also: loadflow, pst_var, mac_em, mac_tra.

3.22.9 Reference:

1. R. P. Schulz, "Synchronous Machine Modeling," presented at the Symposium ``Adequacy and Philosophy of Modeling: System Dynamic Performance," San Francisco, July 9-14, 1972.

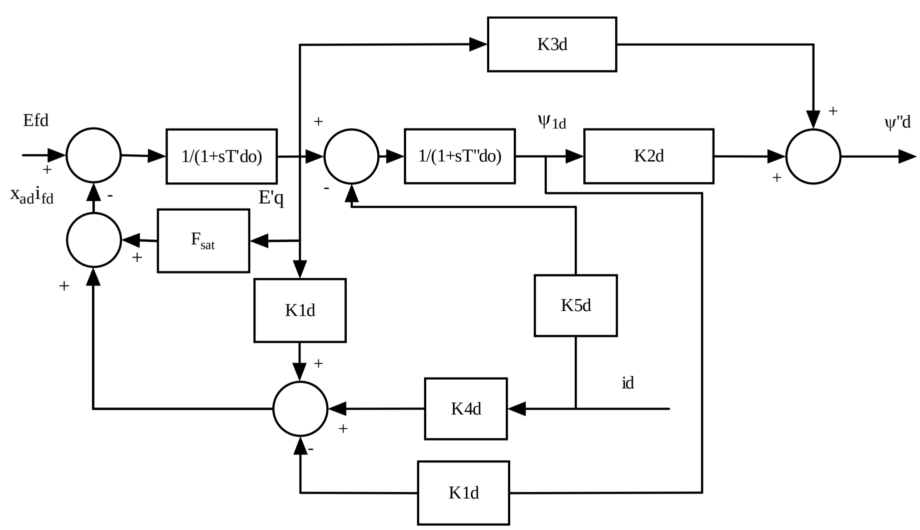


Figure 32 Block Diagram Of Direct Axis

The block diagram coefficients are defined as

$$K_{1d} = \frac{\left(x_d^{'} - x_d^{''}\right)\left(x_d - x_d^{'}\right)}{\left(x_d^{'} - x_l\right)^2}; K_{2d} = \frac{\left(x_d^{'} - x_d^{''}\right)}{\left(x_d^{'} - x_l\right)}$$

$$K_{3d} = \frac{\left(x_d^{''} - x_l\right)}{\left(x_d^{'} - x_l\right)}; K_{4d} = \frac{\left(x_d - x_d^{'}\right)\left(x_d^{'} - x_d^{''}\right)}{\left(x_d^{'} - x_l\right)}; K_{5d} = x_d^{'} - x_l$$

F_{sat} represents the magnetic saturation of the d axis.

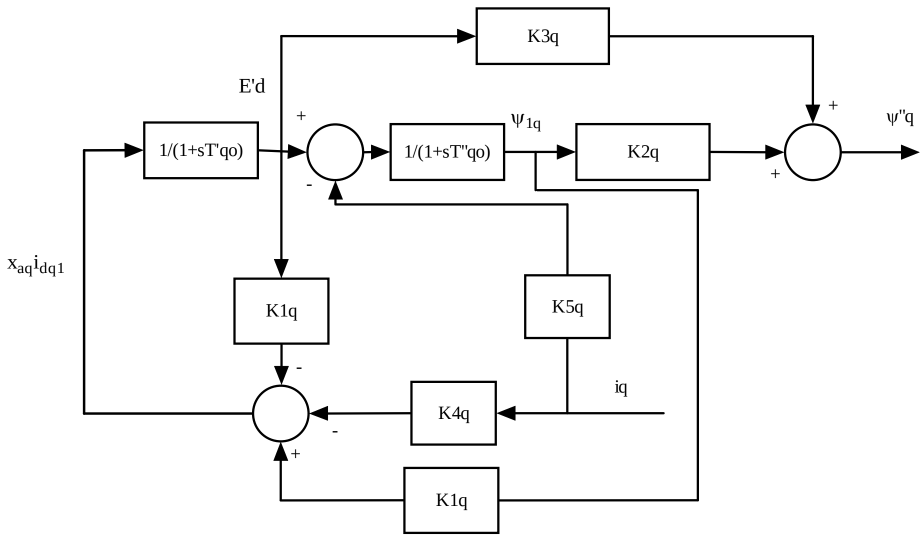


Figure 33 Block Diagram of Quadrature Axis

The block diagram coefficients are defined as

$$K_{1q} = \frac{\left(x_q^{'} - x_q^{''}\right)\left(x_q - x_q^{'}\right)}{\left(x_q^{'} - x_l\right)^2}; K_{2q} = \frac{\left(x_q^{'} - x_q^{''}\right)}{\left(x_q^{'} - x_l\right)}$$

$$K_{3q} = \frac{\left(x_q^{''} - x_l\right)}{\left(x_q^{'} - x_l\right)}; K_{4q} = \frac{\left(x_q - x_q^{'}\right)\left(x_q^{'} - x_q^{''}\right)}{\left(x_q^{'} - x_l\right)}; K_{5q} = x_q^{'} - x_l$$

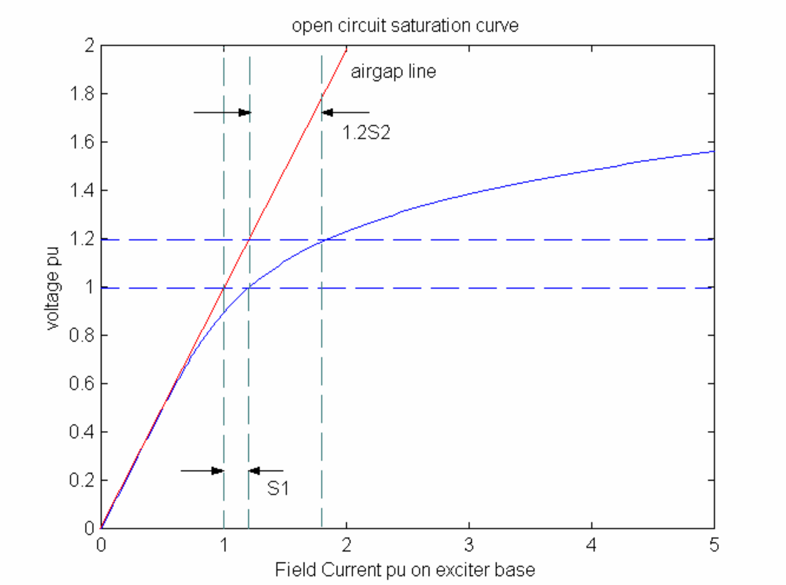


Figure 34 Synchronous Generator Field Saturation Characteristic

3.23 mac_tra

3.23.1 Purpose:

Models a synchronous machine with a voltage behind transient reactance model

3.23.2 Synopsis:

`mac_tra(i,k,bus,flag)`

3.23.3 Description:

mac_tra contains the voltage behind the transient reactance model equations for the initialization, network interface and dynamics computation of the **i**$^{\mathbf{th}}$ synchronous machine (see block diagram in Figure 1).
The m.file **pst_var.m** containing all the global variables required for **mac_tra** should be loaded in the program calling **mac_tra**.
The machines are numbered internally according to the order of the machines in **mac_con**. This information is contained in the array **mac_int** and is set up
automatically by the Y matrix reduction function **red_ybus**.

3.23.4 Inputs:

i the number of the generator
if **i** = 0 all transient model generator computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and their rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the transient generator states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.23.5 Global Variables

3.23.5.1 System variables

basmva		system base MVA
basrad		2 π* system frequency
syn_ref		synchronous reference
mach_ref		reference machine
sys_freq		system frequency in pu
bus_v	V	bus voltage magnitude in pu
bus_ang	θ	bus voltage angle in rad
psi_re	ψ_{re}	real and imaginary components of voltage
psi_im	ψ_{im}	source on system reference frame
cur_re	i_{re}	real and imaginary components of bus
cur_im	i_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.23.5.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu
eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	I_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base
vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech		P_m mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
dmac_ang	$d\delta/dt$	
dmac_spd	$d\omega dt$	
deqprime	dE_q'/dt	
dedprime	dE_d'/dt	
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user
n_mac		number of generators
n_tra		number of subtransient generator models

mac_tra_idx index of subtransient generator models

3.23.6 Data Format

The data format for **mac_tra** is given in Table 11.
The definitions of the saturation factors are given in saturation curve diagram (Figure 2). It is assumed that there is no saturation for field current less than 0.8 pu. Setting the saturation factors to zero eliminates the saturation effect.

Table 11 Data format for mac_tra

column	variable	unit
1	machine number	
2	bus number	
3	base MVA	MVA
5	resistance r_a	pu
6	d-axis synchronous reactance x_d	pu
7	d-axis transient reactance x_d'	pu
9	d-axis open circuit time constant T_{do}'	sec
11	q-axis synchronous reactance x_q	pu
12	q-axis transient reactance x_q'	pu
14	q-axis open circuit time constant T_{qo}'	sec
16	Inertia Constant H	sec
17	local damping coefficient d_o	pu
18	system damping coefficient d_1	pu
19	bus number	
20	saturation factor $S(1.0)$	
21	saturation factor $S(1.2)$	
22	active power fraction	
23	reactive power fraction	

3.23.7 Algorithm:

Based on the machine vector diagram

- the initialization uses the solved load flow bus voltages and angles to compute the internal voltage and the rotor angle.
- In the network interface computation, the voltage behind the transient reactance on the system reference frame is generated.

In the dynamics calculation, the power imbalance and the speed deviation are used to compute the time derivatives of the state variables
This algorithm is implemented in the M-file **mac_tra.m** in the POWER SYSTEM TOOLBOX.
See also: loadflow, pst_var, mac_em, mac_sub.

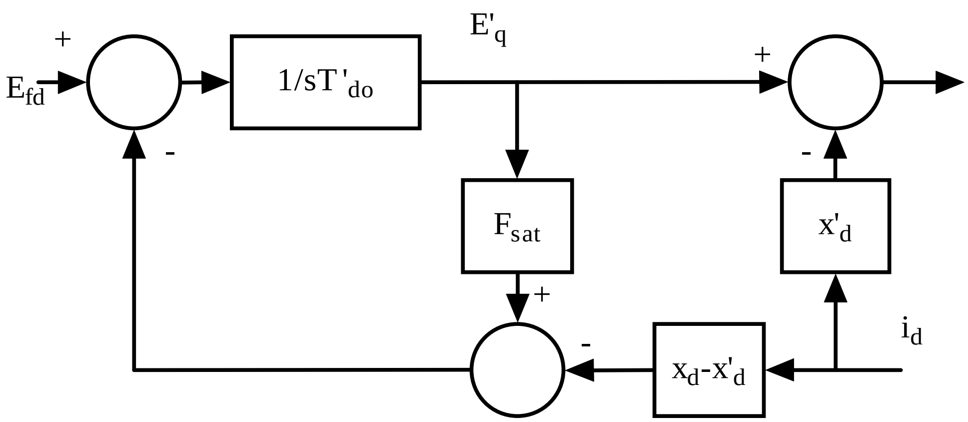


Figure 35 Block diagram direct axis transient generator model

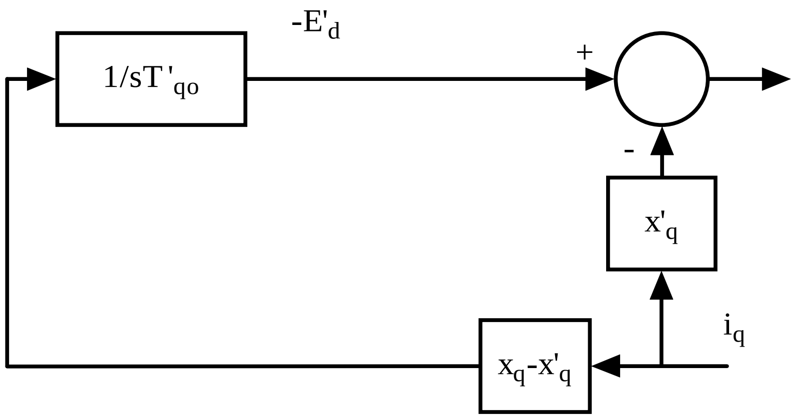


Figure 36 Block diagram quadrature axis transient generator model

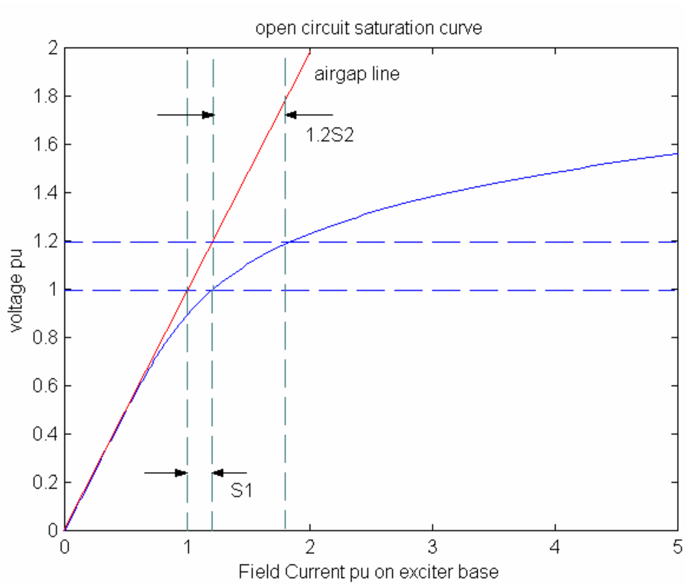


Figure 37 Field Saturation Characteristic

3.24 mdc_sig

3.24.1 Purpose:

Forms the dc controls modulation signal.

3.24.2 Synopsis:

```
mdc_sig(t, k)
```

3.24.3 Description:

f = **mdc_sig** forms the load modulation signal as a function of time. The modulation variable **dc_sig** is passed as a global variable.

The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **mdc_sig**.

3.24.4 Inputs:

t the time in seconds corresponding to **k**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.24.5 Global Variables

dc_sig	V_{sup}	supplementary load modulation signal
n_conv		number of HVDC converters

See also: **dc_cont**

3.24.6 Example

The following version of **mdc_sig** causes a step change in the first rectifier pole control reference after a time of 0.1 s.

```
function f = mdc_sig(t,k)
% Syntax: f = mdc_sig(t,k)
% 4:40 PM 21/08/97
% defines modulation signal for dc converter control
global dc_sig  r_idx i_idx n_conv
f=0; %dummy variable
dc_sig(:,k)=zeros(n_conv,1);
if n_conv~=0
  if t>=0.1
     dc_sig(r_idx(1),k) = .1;
  end
end
return
```

3.25 mexc_sig

3.25.1 Purpose:

Forms the exciter modulation signal.

3.25.2 Synopsis:

```
mexc_sig(t, k)
```

3.25.3 Description:

mexc_sig forms the exciter modulation signal as a function of time. The modulation variable **exc_sig** is passed as a global variable.

The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **mexc_sig**.

3.25.4 Inputs:

t the time in seconds corresponding to **k**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.25.5 Global Variables

exc_sig	V_{sup}	supplementary load modulation signal

n_exc number of exciters

See also: **exc_dc12, exc_st3, smpexc**

3.25.6 Example

The following version of **mexc_sig** causes a step change of 0.01 in Vref at exciter number 1 after a time of 0.1 s.

```
function f = mexc_sig(t,k)
% Syntax: f = mexc_sig(t,k)
% 1:20 PM 15/08/97
% defines modulation signal for exciter control
global exc_sig n_exc
f=0; %dummy variable
if n_exc~=0
   exc_sig(:,k)=zeros(n_exc,1);
end
if t<0.1
   exc_sig(1,k) = 0.01;
end
return
```

3.26 ml_sig

3.26.1 Purpose:

Forms the load modulation signal.

3.26.2 Synopsis:

```
ml_sig(t, k)
```

3.26.3 Description:

ml_sig forms the load modulation signal as a function of time. The modulation variable **lmod_sig** is passed as a global variable.

The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **ml_sig**.

3.26.4 Inputs:

t the time in seconds corresponding to **k**

k the integer time step in a simulation

In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.26.5 Global Variables

lmod_sig	V_{sup}	supplementary load modulation signal
n_lmod		number of load modulation controls

See also: **lmod**

3.26.6 Example

The following version of ml_sig causes a step change in load of 0.5 PU after a time of 0.1 s.

```
function f = ml_sig(t,k)
% Syntax: f = ml_sig(t,k)
%4:40 PM 15/08/97
% defines modulation signal for lmod control
global lmod_sig n_lmod
f=0; %dummy variable
% you modify the following to do what you want with the load
% lmod_con must be specified in the data file
% and the load bus must be in the nonconforming load list.
if n_lmod~=0
  if t<0.1
     lmod_sig(:,k)= zeros(n_lmod,1);
  else
     lmod_sig(1,k) = 0.5;
  end
end
return
```

3.27 msvc_sig

3.27.1 Purpose:

Forms the svc modulation signal.

3.27.2 Synopsis:

```
msvc_sig(t, k)
```

3.27.3 Description:

msvc_sig forms the load modulation signal as a function of time. The modulation variable **svc_sig** is passed as a global variable.

The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **msvc_sig**.

3.27.4 Inputs:

t the time in seconds corresponding to **k**

k the integer time step in a simulation

In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.27.5 Global Variable

svc_sig V_{sup} supplementary load modulation signal

n_svc number of svc controls

See also: **svc**

3.27.6 Example

The following version of **msvc_sig** causes a step change in all the svc reference voltages after a time of 0.1s.

```
function f = msvc_sig(t,k)
% Syntax: f = msvc_sig(t,k)
% 4:39 PM 15/08/97
% defines modulation signal for svc control
global svc_sig n_svc
f=0; %dummy variable
if n_svc ~=0
    svc_sig(:,k) = zeros(n_svc,1);
  if t<=0.1
     svc_sig(:,k) = 0.1;
  end
end
return
```

3.28 mtg_sig

3.28.1 Purpose:

Forms the turbine governor modulation signal.

3.28.2 Synopsis:

```
mtg_sig(t, k)
```

3.28.3 Description:

mtg_sig forms the turbine governor modulation signal as a function of time. The modulation variable **tg_sig** is passed as a global variable.

The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **mtg_sig**.

3.28.4 Inputs:

t the time in seconds corresponding to **k**

k the integer time step in a simulation

In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.28.5 Global Variables

tg_sig V_{sup} supplementary power order modulation signal

n_tg number of turbine governor controls

See also: **tg**

3.28.6 Example

The following version of mtg_sig causes a step change of 0.01 in governor power demand at generator 1 after a time of 0.1 s.

```
function f = mtg_sig(t,k)
% Syntax: f = mtg_sig(t,k)
% 1:19 PM 15/08/97
% defines modulation signal for turbine governor control
global tg_sig n_tg
f=0; %dummy variable
if n_tg~=0
  tg_sig(:,k) = zeros(n_tg,1);
  if t>0.1
     tg_sig(:,k) = zeros(n_tg,1);
  end
end
return
```

3.29 nc_load

3.29.1 Purpose:

Solves the complex voltages at non-conforming load buses

3.29.2 Synopsis:

```
V_nc = nc_load(bus,flag,Y22,Y21,psi,Vo,tol)
V_nc = nc_load(bus,flag,Y22,Y21,psi,Vo,tol,k)
```

3.29.3 Description:

nc_load(bus,flag,Y22,Y21,psi,Vo,tol) computes the complex voltage **V** at the non-conforming load buses the SVC buses and the HVDC HT buses using a Newton-Raphson algorithm.

nc_load(bus,flag,Y22,Y21,psi,Vo,tol,k) is used in the simulation process at each network interface calculation.

The m.file **pst_var.m** containing all the global variables required for **nc_load** should be loaded in the program calling **nc_load**.

3.29.4 Inputs:

bus	solved loadflow bus data
flag	solution mode control 0 - initialization 1 - network interface computation 2 - dynamic calculation not needed in this model
Y22	reduced Y matrix of non-conforming loads (output from red_ybus)
Y21	reduced Y matrix connecting non conforming load current to machine internal voltages
psi	machine internal voltage, not used in initialization
V_o	initial non conforming load bus voltage vector, not used in initialization
tol	tolerance for Newton's algorithm convergence, not used in initialization
k	integer time step (only for svc/facts models), not used in initialization

3.29.5 Outputs:

V_nc	solved non-conforming load bus voltage vector

3.29.6 Global Variables:

load_con : non-conforming bus specification matrix
load_pot : non-conforming bus constants
bus_int : internal bus number vector
svc_con : svc specification matrix
svc_idx: svc index vector
n_svc : number of svcs
svc_pot : svc constants
B_cv: svc state
i_dci : inverter dc current
i_dcr : rectifier dc current
dcc_pot : dc controls constants
alpha: rectifier firing angle
gamma: inverter extinction angle

basmva : base MVA
r_idx: rectifier converter index
i_idx: inverter converter index
n_conv : number of HVDC converters
n_dcl : number of HVDC lines
ldc_idx: HVDC line index

3.29.7 Data Format

The non-conforming load data is contained in the **i**$^{\mathbf{th}}$ row of the matrix variable **load_con**. The data format for **load_con** is given in Table 12.

Table 12 Data format for load_con

column	variable	unit
1	bus number	
2	fraction of constant active power load	
3	fraction of constant reactive power load	
4	fraction of constant active current load	
5	fraction of constant reactive current load	

Note: SVCs obtain their initial values from the generator reactive power specified in bus. If an SVC bus has loads specified also, these may be defined as non conforming in the same way as any load bus. If there is no load, then the SVC bus must still be declared as non conforming, but with zero entries for the load fractions. HVDC buses are specified in the load flow as the Low Tension buses, these buses cannot have loads, other than the HVDC loads.

3.29.8 Algorithm:

The current balance equation at the non-conforming load buses is given by

$$Y21\psi + Y22V = (Icc(V) + Icp(V))$$

where Icc is the current injection due to the constant current components and Icp is the current injection due to the constant power components. These injections are functions of the bus voltage. The constant impedance components are included in **Y22** (which is computed in the function **red_ybus**). Sensitivities of these injections with respect to the voltage is used to formulate a Newton's algorithm to solve this nonlinear equation. The initial guess **Vo** is typically the bus solution at the previous time step.
See **s_simu.m** and **svm_mgen.m** for examples of use.
This algorithm is implemented in the M-file **nc_load.m** in the POWER SYSTEM TOOLBOX.
See also: pst_var, red_ybus, svc, s_simu, svm_mgen, i_simu

3.30 pss

3.30.1 Purpose:

Models power system stabilizers

3.30.2 Synopsis:

```
pss(i,k,bus,flag)
```

3.30.3 Description:

pss(i,k,bus,flag) contains the equations of a power system stabilizer (PSS) model shown in Figure 1 for the initialization, machine interface and dynamics computation of the **i**$^{\text{th}}$ excitation system.
The m.file **pst_var.m** containing all the global variables required for **pss** should be loaded in the program calling **pss**.

3.30.4 Inputs:

i the number of the generator which the PSS is controlling
if **i** = 0 all PSS computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.30.5 Global Variables

3.30.5.1 System variables

basmva		system base MVA

3.30.5.2 Synchronous Generator Variables

mac_spd	ω	machine speed in pu
pelect	P_e	electrical active output power in pu on system base
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user

3.30.5.3 Excitation System Variable

exc_sig	V_{sup}	supplementary input signal to exciter ref input

3.30.5.4 PSS variables

pss1	washout state variable
pss2	first lead-lag compensator state variable
pss3	second lead-lag compensator state variable
dpss1	
dpss2	
dpss3	
pss_con	matrix of pss parameters specified by user
pss_pot	internally computed matrix of pss constants
n_pss	number of pss
pss_idx	index of pss
pss_T	**pss_con(pss_idx,4)**
pss_T2	**pss_con(pss_idx,6)**
pss_T4	**pss_con(pss_idx,8)**
pss_T4_idx	index of nonzero T4 for pss

pss_noT4 index of zero T4 for pss
pss_sp_idx index of pss with speed input
pss_p_idx index of pss with power input

3.30.6 Data Format

The pss data is contained in the $\mathbf{i^{th}}$ row of the matrix variable **pss_con**. The data format for **pss** is shown in Table 13.

Table 13 Data format for PSS

column	data	unit
1	type 1 speed input 2 power input	
2	machine number	
3	gain G_{pss}	
4	washout time constant T_w	sec
5	lead time constant T_{n1}	sec
6	lag time constant T_{d1}	sec
7	lead time constant T_{n2}	sec
8	lag time constant T_{d2}	sec
9	maximum output limit y_{max}	pu
10	minimum output limit y_{min}	pu

A constraint on using **pss** is that $T_1 \neq 0$ and $T_2 \neq 0$. The output of the power system stabilizer is limited by an upper and a lower limit.

Note: The PSS gain $G_{pss}T_w$ is equal to the normally defined stabilizer multiplied by the washout time constant.

3.30.7 Algorithm:

Based on the pss block diagram

- on initialization the washout state variable is set to
  the generator speed for type = 1
  the electrical power on the generator base if type = 2
  the remaining states are set to zero. The PSS output is also zero.
- In the network interface computation, the PSS output signals exc_sig are set.
- In the dynamics calculation, the input machine speed or electrical power is used to calculate the rates of change of the PSS states.

This algorithm is implemented in the M-file **pss** in the POWER SYSTEM TOOLBOX.

See also: pst_var, smpexc, exc_dc12, exc_st3.

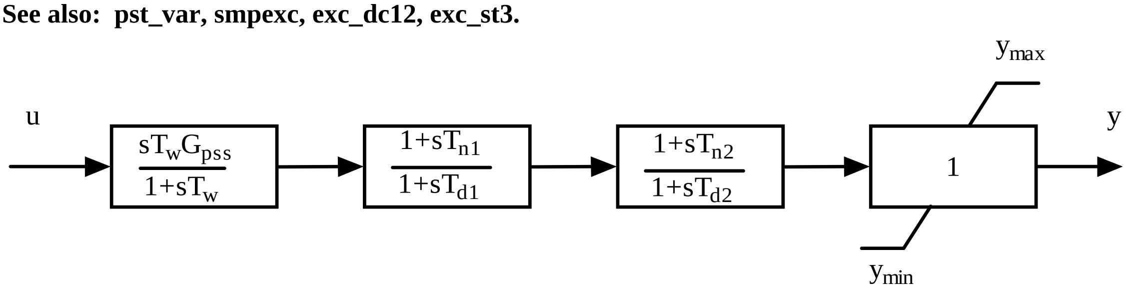


Figure 38 Power System Stabilizer Model Block Diagram

3.31 pss_des

3.31.1 Purpose:

Allows trial and error determination of PSS parameters to fit an ideal frequency response

3.31.2 Syntax:

```
[tw,t1,t2,t3,t4] = pss_des(a,b,c,d,rot_idx)
```

3.31.3 Global variables

There are no global variable in this file

3.31.4 Description:

This function allows the user to select, on a cut-and-try basis, power system stabilizer parameters which fit as closely as desired the ideal phase lead between V_{ref} and the generator electrical torque necessary to produce a damping torque over the matched frequency range.

3.31.5 Inputs:

a the state matrix of the system for which the PSS is to be designed
b the input matrix associated for the exciter reference input
c the output matrix associated with the generator mechanical torque
d the feed forward matrix between the voltage reference and the generator mechanical torque. Normally zero
rot_idx the index of rotor angle states
rot_idx = sort([ang_idx;ang_idx+1])

The inputs are normally obtained by running **svm_mgen**

3.31.6 Outputs:

tw the washout time constant (s)
t1 the first lead time constant (s)
t2 the first lag time constant (s)
t3 the second lead time constant (s)
t4 the second lag time constant (s)

3.31.7 Algorithm:

The user is asked to provide a set of PSS parameters - default settings are provided. The ideal stabilizer frequency response is calculated from the response between the exciter voltage reference input and the generator electrical power output with the rotor angle states removed. The rotor states are removed from the a,b and c matrices supplied using rot_idx. The frequency response of the modified state space system is calculated using **statef**. The ideal response is plotted together with the stabilizer frequency response.
The user can then perform an additional check with new parameters in order to obtain a sufficiently close fit to the ideal frequency response characteristic.
This algorithm is implemented in the M-file **pss_des** in the POWER SYSTEM TOOLBOX.

3.31.8 Example

For the system in d2asbeg:

```
a=a_mat;b=b_vr(:,1);c=c_p(1,:);d=0;
rot_idx = sort([ang_idx;ang_idx+1])
rot_idx =
     1
     2
    12
    13
    23
    24
    34
    35
[tw,t1,t2,t3,t4] = pss_des(a,b,c,d,rot_idx);
enter the start frequency (Hz) [0.1]
enter the frequency step (Hz) [0.01]
enter the end frequency (Hz) [2.0]
input the washout time constant in secs:[5]10
the first lead time constant in secs:[.2].07
the first lag time constant in secs:[.02]
the second lead time constant in secs:[.2].07
the second lag time constant in secs:[.02]
Current plot held
```

```
Do you wish to try another pss design: Y/N[Y]n
more_plots =n
```

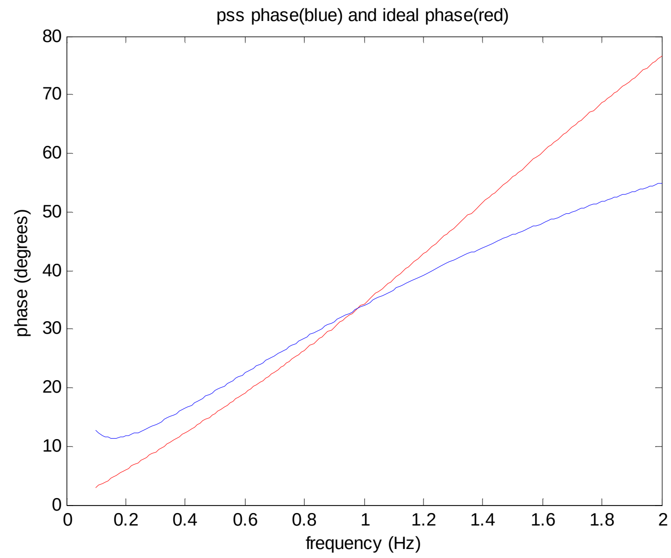


Figure 39 Ideal and PSS Phase Lead

3.32 pst_var

3.32.1 Purpose:

Declare global variables for functions in POWER SYSTEM TOOLBOX

3.32.2 Synopsis:

`pst_var`

3.32.3 Description:

pst_var declares all the global variables required for the functions in POWER SYSTEM TOOLBOX. All these variables can be displayed in matrix form or graphically by MATLAB. **pst_var** is inserted at the top of script files (m.files) for simulation and building state matrices. To start a new simulation, the memory should be cleared by typing clear and clear global. This is done automatically in s_simu and svm_mgen.

3.32.4 Global Variables:

3.32.4.1 System variables

basmva		system base MVA
basrad		2 π* system frequency
syn_ref		synchronous reference
mach_ref		reference machine
sys_freq		system frequency in Hz
bus_v	V	bus voltage magnitude in pu
bus_ang	θ	bus voltage angle in rad
psi_re	ψ_{re}	real and imaginary components of voltage
psi_im	ψ_{im}	source on system reference frame
cur_re	I_{re}	real and imaginary components of bus
cur_im	I_{im}	current on system reference frame
bus_int		array to store internal bus ordering

3.32.4.2 Synchronous Generator Variables

mac_ang	δ	machine angle in rad/sec
mac_spd	ω	machine speed in pu

eqprime	E_q'	pu on machine base
edprime	E_d'	pu on machine base
psikd	ψ_{kd}	pu on machine base
psikq	ψ_{kq}	pu on machine base
curd	i_d	d-axis current on system base
curq	i_q	q-axis current on system base
curdg	i_{dg}	d-axis current on machine base
curqg	i_{qg}	q-axis current on machine base
fldcur	I_{fd}	field current on machine base
psidpp	ψ_d''	pu on machine base
psiqpp	ψ_q''	pu on machine base
vex	V_{ex}	field voltage on machine base
eterm	E_T	machine terminal voltage in pu
theta	θ	terminal voltage angle in rad
ed	E_d	d-axis terminal voltage in pu
eq	E_q	q-axis terminal voltage in pu
pmech	P_m	mechanical input power in pu
pelect	P_e	electrical active output power in pu
qelect	Q_e	electrical reactive output power in pu
dmac_ang	$d\delta/dt$	
dmac_spd	$d\omega dt$	
deqprime	dE_q'/dt	
dedprime	dE_d'/dt	
dpsikd	$d\psi_{kd}/dt$	
dpsikq	$d\psi_{kq}/dt$	
mac_int		array to store internal machine ordering
mac_pot		internally set matrix of machine constants
mac_con		matrix of generator parameters set by user
ibus_con		vector specifying infinite buses set by user
n_mac		number of generators
n_em		number of em (classical) generator models
n_tra		number of transient generator models
n_sub		number of subtransient generator models
n_ib		number of infinite buses
mac_em_idx		index of em generator models, i.e. mac_con(mac_em_idx,:) picks out the em data
mac_tra_idx		index of transient generator models
mac_sub_idx		index of subtransient generator models
mac_ib_idx		index of infinite buses
not_ib_idx		index of generators which are not modelled as infinite buses
mac_ib_em		index of em generatoirs modelled as infinite buses
mac_ib_tra		index of transient generators modelled as infinite buses
mac_ib_sub		index of subtransient generators modelled as infinite buses
n_ib_em		number of em generators modelled as infinite buses
n_ib_tra		number of transient generators modelled as infinite buses
n_ib_sub		number of subtransient generators modelled as infinite buses

3.32.4.3 Excitation System Variables

Efd	E_{fd}	exciter output voltage, equal to generator field voltage pu
V_R	V_R	regulator output voltage in pu
V_A	V_A	regulator output voltage in pu
V_As	V_{As}	regulator voltage state variable in pu
R_f	R_f	stabilizing transformer state variable
V_FB	V_{FB}	feedback from stabilizing transformer
V_TR	V_{TR}	voltage transducer output in pu
V_B	V_B	potential circuit voltage output in pu
dEfd	dE_{fd}/dt	
dV_R	dV_R/dt	
dV_As	dV_{As}/dt	
dR_f	dR_f/dt	
dV_TR	dV_{TR}/dt	
exc_sig	V_{sup}	supplementary input signal to exciter ref input
exc_pot		matrix of internally set exciter constants
exc_con		matrix of exciter data set by user
smp_idx		index of simple exciters, i.e., exc_con(smp_idx,:)
n_smp		number of simple exciters
dc_idx		index of dc exciters
n_dc		number of dc exciters
dc2_idx		index of type 2 dc exciters
n_dc2		number of type 2 dc exciters
st3_idx		index of st3 exciters
n_st3		number of st3 exciters
smp_TA		**exc_con(smp_idx,5)**
smp_TA_idx		index of non-zero TA for simple exciter
smp_noTA_idx		index of zero TA for simple exciter
smp_TB		**exc_con(smp_idx,6)**
smp_TB_idx		index of nonzero TB for simple exciters
smp_noTB_idx		index of zero TB for simple exciters
smp_TR		**exc_con(smp_idx,3)**
smp_TR_idx		index of nonzero TR for simple exciter
smp_no_TR_idx		index of zero TR for simple exciter
dc_TA		**exc_con(dc_idx,5)**
dc_TA_idx		index of nonzero TA for dc exciter
dc_noTR_idx		index of zero TA for dc exciter
dc_TB		**exc_con(dc_idx,6)**
dc_TB_idx		index of non-zero TB for dc exciter
dc_noTB_idx;		index of zero TB for dc exciter
dc_TE		**exc_con(dc_idx,11)**
dc_TE_idx		index of nonzero TE for dc exciter
dc_noTE_idx		index of zero TE for dc exciter
dc_TF		**exc_con(dc_idx,17)**
dc_TF_idx		index of TF for dc exciter
dc_TR		**exc_con(dc_idx, 3)**
dc_TR_idx		index of nonzero TR for dc exciter
dc_noTR_idx		index of zero TR for dc exciter
st3_TA		**exc_con(st3_idx,5)**
st3_TA_idx		index of nonzero TA for st3 exciter

st3_noTA_idx	index of zero TA for st3 exciter
st3_TB	**exc_con(st3_idx,6)**
st3_TB_idx	index of nonzero TB for st3 exciter
st3_noTB_idx	index of zero TB for st3 exciter
st3_TR	**exc_con(st3_idx,3)**
st3_TR_idx	index of nonzero TR for st3 exciter
st3_noTR_idx	index of zero TR for st3 exciter

3.32.4.4 Power System Stabilizer Variables

pss1	washout state variable
pss2	first lead-lag compensator state variable
pss3	second lead-lag compensator state variable
dpss1	
dpss2	
dpss3	
pss_con	matrix of pss parameters specified by user
pss_pot	Internally computed matrix of pss constants
n_pss	number of pss
pss_idx	index of pss
pss_T	**pss_con(pss_idx,4)**
pss_T2	**pss_con(pss_idx,6)**
pss_T4	**pss_con(pss_idx,8)**
pss_T4_idx	index of nonzero T4 for pss
pss_noT4	index of zero T4 for pss
pss_sp_idx	index of pss with speed input: pss_con(pss_sp_idx,1) = 1
pss_p_idx	index of pss with power input: pss_con(pss_p_idx,1) = 2

3.32.4.5 Turbine-governor Variables

tg1	
tg2	
tg3	
tg4	
tg5	
dtg1	
dtg2	
dtg3	
dtg4	
dtg5	
tg_con	matrix of turbine governor specifications set by user
tg_pot	internally set matrix of turbine governor constants
n_tg	number of turbine governors
tg_idx	index of turbine governors

3.32.4.6 Induction Motor Variables

tload	motor load torque as a fraction of the initial torque
t_init	initial motor load torque in pu. on motor base
pot	motor active power in pu. on motor base
qmot	motor reactive power in pu. on motor base
vdmot	motor direct axis stator voltage in pu.
vqmot	motor quadrature axis stator voltage in pu.
idmot	motor direct axis stator current in pu.
iqmot	motor quadrature axis voltage im pu.
ind_con	matrix of induction motor parameters set by user

ind_pot	matrix of induction motor constants set internally
ind_int	index of internal induction motor buses
motbus	buses to which induction motors are connected
vdp	V'd direct axis transient voltage (state)
vqp	V'q quadrature axis transient voltage (state)
slip	fractional slip (state)
dvdp	dV'_d/dt
dvqp	dV'_q/dt
dslip	ds/dt

3.32.4.7 Induction generator variables

tmig	mechanical torque from driving turbine
pig	generator active power
qig	generator reactive power
vdig	d axis stator voltage
vqig	q axis stator current
idig	d axis stator current
iqig	q axis stator current
igen_con	matrix of induction generator data
igen_pot	matrix of induction generator constants
igen_int	internal numbers for induction generators
igbus	internal bus numbers for induction generators
n_ig	number of induction generators
vdpig	d axis voltage behind transient impedance
vqpig	d axis voltage behind transient impedance
slig	induction generator slip
dvdpig	rate of change of **vdpig**
dvqpig	rate of change of **vqpig**
dslig	rate of change of **slig**

3.32.4.8 Non Conforming Load Variables

load_con	matrix of non conforming load parameters set by user
load_pot	matrix of non-conforming load constants set internally

3.32.4.9 Static VAR Compensator Variables

B_cv	B_{cv}	svc susceptance in pu
dB_cv	dB_{cv}/dt	
svc_sig	V_{sup}	supplementary signal into the reference input
svc_con		matrix of svc parameters supplied by user
svc_pot		internally calculated matrix of svc constants
n_svc		number of svcs
svc_idx		index of svcs included in **load_con**

3.32.4.10 HVDC System Variables

dcsp_con	HVDC converter specification matrix
dcl_con	HVDC line specification matrix
dcc_con	HVDC pole control specification matrix
r_idx	rectifier converter index
i_idx	inverter converter index
n_dcl	number of HVDC lines
n_conv	number of HVDC converters
ac_bus	index of converter ac buses

rec_ac_bus	index of rectifier ac buses
inv_ac_bus	index of inverter ac buses
inv_ac_line	index of inverter ac lines
rec_ac_line	index of rectifier ac lines
ac_line	index of converter ac lines
dcli_idx	index of HVDC lines
tap	HVDC transformer tap settings
tapr	HVDC rectifier transformer tap settings
tapi	HVDC inverter transformer tap settings
tmax	HVDC tap maximum values
tmin	HVDC tap minimum values
tstep	HVDC tap steps
tmaxr	rectifier maximum tap values
tmaxi	inverter maximum tap values
tminr	rectifier minimum tap values
tmini	inverter minimum tap values
tstepr	rectifier tap step
tstepi	inverter tap step
Vdc	HVDC Voltage kV
i_dc	HVDC current kA
dc_pot	HVDC line constant matrix
alpha	rectifier firing angle
gamma	inverter extinction angle
dc_sig	HVDC external modulation signal
cur_ord	HVDC current order
Vdc_ref	inverter HVDC voltage reference
dcc_pot	HVDC pole controls constant matrix
no_cap_idx	index of HVDC lines having no capacitance
cap_idx	index of HVDC lines having capacitance
no_ind_idx	index of HVDC lines having no inductance
l_no_cap	number of HVDC lines having no capacitance
l_cap	number of HVDC lines having capacitance
i_dcr	rectifier HVDC current kA (state)
i_dci	inverter HVDC current kA (state)
v_dcc	HVDC line capacitance voltage kV (state)
di_dcr	rate of change of rectifier HVDC current
di_dci	rate of change of inverter HVDC current
dv_dcc	rate of change of HVDC line capacitance voltage
v_conr	HVDC rectifier pole control state
dv_conr	rate of change of HVDC rectifier pole control state
v_coni	HVDC inverter pole control state
dv_coni	rate of change of HVDC inverter pole control state

3.32.4.11 Load Modulation Variables

lmod_st	*lm*	load modulation state
dlmod_st	*dlm/dt*	
lmod_sig	V_{sup}	supplementary signal into the reference input
lmod_con		matrix of lmod parameters supplied by user
lmod_pot		internally calculated matrix of lmod constants
n_lmod		number of load modulation controls
lmod_idx		index of modulation controls included in **load_con**

3.32.4.12 Reactive Load Modulation Variables

rlmod_st	r*lm*	reactive load modulation state
drlmod_st	*drlm/dt*	
lmod_sig	V_{sup}	supplementary signal into the reference input
rlmod_con		matrix of rlmod parameters supplied by user
rlmod_pot		internally calculated matrix of rlmod constants
n_rlmod		number of reactive load modulation controls
rlmod_idx		index of reactive modulation controls included in **load_con**

3.33 red_ybus

3.33.1 Purpose:

Forms the reduced admittance matrix used in simulations.

3.33.2 Synopsis:

```
[red_Y,rec_V] = red_ybus(bus,line)
[Y11,Y12,Y21,Y22,rec_V1,rec_V2,bus_ord] = red_ybus(bus,line)
```

3.33.3 Description:

```
[red_Y,rec_V] = red_ybus(bus,line)
```

Returns the reduced admittance matrix **red_Y** and the voltage reconstruction matrix **rec_V** so that

$$Ig = red\_Y * Vg$$

$$Vb = rec\_V * Vg$$

where Ig is a column vector of generator current injection, Vg and Vb are column vectors of generator bus voltages and load bus voltages, respectively.

```
[Y11,Y12,Y21,Y22,rec_V1,rec_V2,bus_ord] = red_ybus(bus,line)
```

Returns the reduced admittance matrix in partitioned form. This is required when there are non-conforming load buses in the system. The function uses the bus data in **bus**, the line data in **line**, the machine reactance in **mac_con** and **ind_con** ,and the load data in **load_con** to return the reduced admittance matrices **Y11, Y12, Y21, Y22** and the voltage reconstruction matrix **rec_V1**, **rec_V2** so that

$$Ig = Y11 * Vg + Y12 * Vnc$$

$$Vb = rec\_V1 * Vg + rec\_V2 * Vnc$$

where Vnc is the column vector of the non-conforming load bus voltages.

The matrices **Y21, Y22** and the bus reordering information contained in the column vector **bus_ord**, are used in **nc_load**.

If the full input is specified when **load_con** is empty, the additional outputs are set to the null matrix [].

The output variables of **red_ybus** are all in full matrix form. The user can convert them to sparse matrix form if necessary.

3.33.4 Inputs:

bus	a solved bus data set
line	a solved line data set

3.33.5 Outputs:

Y11	the reduced admittance matrix connecting the generator current injections to the internal generator and induction motor voltages
Y12	the admittance matrix component which gives the generator and motor currents due to the voltages at non conforming load and SVC buses
Y21	the admittance matrix component which gives the non conforming load and SVC currents in terms of the generator and induction motor internal voltages
Y22	the admittance matrix connecting the non conforming load and SVC currents to the voltages at the non conforming load and SVC buses
rec_V1	The voltage reconstruction matrix which gives the original bus voltages components due to the generator and induction motor internal bus voltages
rec_V2	The voltage reconstruction matrix which gives the original bus voltages components due to the non conforming load and SVC bus voltages

bus_ord An index vector giving the non conforming loads first followed by the conforming loads

3.33.6 Global Variables:

basmva system base MVA
bus_int array to store internal bus ordering
mac_int array to store internal machine ordering
mac_con matrix of generator parameters set by user
ind_con matrix of induction motor parameters set by user
ind_int index of internal induction motor buses
ind_pot matrix of induction motor constants set internally
igen_con matrix of induction generator parameters set by user
igen_int index of internal induction generator buses
igen_pot matrix of induction generator constants set internally
load_con matrix of non conforming load parameters set by user

3.33.7 Example:

Consider the 11 bus, four generator, 2 Area System in **d2a_sub.m**.
The following is a diary record of a call to red_ybus

```
pst_var
d2asub
basmva = 100;
[Y_red, V_rec] = red_ybus(bus,line)
Y_red =

   1.2365 - 9.9183i   1.3727 + 6.9137i   0.4129 + 0.5492i   0.6755 + 0.8344i
   1.3727 + 6.9137i   2.5317 -11.7642i   0.6755 + 0.8344i   1.1017 + 1.2650i
   0.4129 + 0.5492i   0.6755 + 0.8344i   1.6591 -10.3017i   2.0111 + 6.2912i
   0.6755 + 0.8344i   1.1017 + 1.2650i   2.0111 + 6.2912i   3.4936 -12.7722i
V_rec =

   0.7245 - 0.0343i   0.1920 - 0.0381i   0.0153 - 0.0115i   0.0232 - 0.0188i
   0.1920 - 0.0381i   0.6732 - 0.0703i   0.0232 - 0.0188i   0.0351 - 0.0306i
   0.2746 - 0.0841i   0.4241 - 0.1467i   0.0498 - 0.0423i   0.0755 - 0.0688i
   0.5589 - 0.0550i   0.3075 - 0.0611i   0.0244 - 0.0184i   0.0371 - 0.0300i
   0.0153 - 0.0115i   0.0232 - 0.0188i   0.7138 - 0.0461i   0.1748 - 0.0559i
   0.0232 - 0.0188i   0.0351 - 0.0306i   0.1748 - 0.0559i   0.6452 - 0.0970i
   0.0498 - 0.0423i   0.0755 - 0.0688i   0.2358 - 0.1221i   0.3614 - 0.2039i
   0.3075 - 0.0611i   0.4768 - 0.1126i   0.0371 - 0.0300i   0.0563 - 0.0490i
   0.1688 - 0.0666i   0.2599 - 0.1134i   0.1485 - 0.0863i   0.2271 - 0.1431i
   0.0244 - 0.0184i   0.0371 - 0.0300i   0.5418 - 0.0738i   0.2798 - 0.0894i
   0.0371 - 0.0300i   0.0563 - 0.0490i   0.2798 - 0.0894i   0.4319 - 0.1554i
```

Note: It is necessary to have **basmva** specified before calling **red_ybus**. The calling sequence is more complex if induction motors or generators or non-conforming loads are specified. You can see the necessary calling sequence by looking in the **s_simu** code.

3.33.8 Algorithm:

The function **red_ybus** sets up an admittance matrix which includes of the generator and induction motor internal buses and the load buses: the load buses include the SVC and the HVDC equivalent HT buses. Then Kron reduction is performed to eliminate all the load buses not specified in **load_con**. The buses are reordered so that the non conforming load buses are first, in the order in which they are specified in **load_con**. The other buses follow in the order in which they are specified in **bus**. Initially, the HVDC ac buses are the transformer LT buses specified in the load flow. However, **red_ybus** transforms these so that in transient simulation the bus voltage retained in Y_red is the equivalent HT converter bus. This makes the ac/dc interface much easier and yet still gives the freedom to specify a Thevenin equivalent reactance for the HVDC commutating reactance.
This algorithm is implemented in the M-file **red_ybus.m** in the POWER SYSTEM TOOLBOX.
See also: loadflow, ybus, pst_var, mac_em, mac_ind, mac_igen, mac_tra, mac_sub, nc_load, s_simu, svm_mgen y_switch

3.34 rlmod

3.34.1 Purpose:

A reactive load modulation control for transient simulation

3.34.2 Synopsis:

```
rlmod(i,k,bus,flag)
```

3.34.3 Description:

rlmod(i,k,bus,flag) contains the equations of a reactive load modulation control system for the initialization, machine interface and dynamics computation of the **ith** modulation control.
Modulation is controlled through the global variable **rlmod_sig**. This is modified by the function **rml_sig** which should be written by the user to obtain the required load modulation charactersitic.
The m.file **pst_var.m** containing all the global variables required for **rlmod** should be loaded in the program calling **rlmod**.

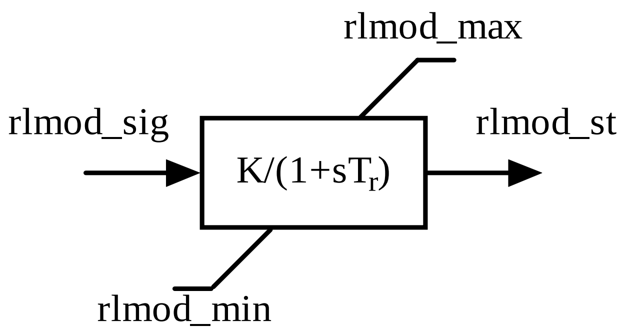


Figure 40 rlmod

3.34.4 Inputs:

i	the number of the reactive load modulation control if **i** = 0 all load modulation computations are performed using MATLAB vector methods. **This is the preferred mode.**
k	the integer time step in a simulation In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.
bus	the solved bus specification matrix
flag	indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- There is no need to perform a network interface calculation for **rlmod**
- The rates of change of the rlmod state is calculated when **flag** = 2, using the modulating signal **rlmod_sig** at the time specified by **k.**

3.34.5 Global Variables

3.34.5.1 System variables

basmva	system base MVA
bus_int	array to store internal bus ordering

3.34.5.2 Load Modulation Variables

rlmod_st	*rlm*	reactive load modulation state
drlmod_st	*drlm/dt*	
rlmod_sig	V_{sup}	supplementary signal into the reference input
rlmod_con		matrix of rlmod parameters supplied by user
rlmod_pot		internally calculated matrix of rlmod constants

n_rlmod number of reactive load modulation controls
rlmod_idx index of reactive modulation controls included in **load_con**

3.34.6 Data Format

The load modulation control data is contained in the **i**$^{\mathbf{th}}$ row of the matrix **rlmod_con**. The data format for **rlmod_con** is given in Table 14.

Table 14 Data format for rlmod

column	variable	unit
1	reactive load modulation number	
2	bus number	
3	modulation base MVA	MVA
4	maximum susceptance r*lmod_max*	pu
5	minimum susceptance *rlmod_min*	pu
6	regulator gain K	pu
7	regulator time constant T_R	sec

3.34.7 Algorithm:

To use the **rlmod** function, the reactive load modulation buses must be declared via **load_con** as non-conforming load buses. The **rlmod** buses may also have non-conforming loads In the network interface computation, the reactive load modulation output is used to adjust the susceptance at the control buses in the solution for the bus voltages in **nc_load**. In the dynamics calculation, the rate of change of the load modulation control state is adjusted according to the signal **rlmod_sig**. An anti-windup limit is used to reset the state variable.
This algorithm is implemented in the M-file **rlmod** in the POWER SYSTEM TOOLBOX.
See also: nc_load, pst_var, rml_sig.

3.35 rml_sig

3.35.1 Purpose:

Forms the reactive load modulation signal.

3.35.2 Synopsis:

```
rml_sig(t, k)
```

3.35.3 Description:

rml_sig forms the reactive load modulation signal as a function of time. The modulation variable **rlmod_sig** is passed as a global variable.
The m.file **pst_var.m** containing all the global variables should be loaded in the program calling **rml_sig**.

3.35.4 Inputs:

t the time in seconds corresponding to **k**
k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

3.35.5 Global Variables

rlmod_sig V_{sup} supplementary reactive load modulation signal
n_rlmod number of reactive load modulation controls
See also: **rlmod**

3.35.6 Example

The following version of **rml_sig** causes a step change in reactive load 1 of 0.5 PU at t=0.

```
function f = rml_sig(t,k)
% Syntax: f = rml_sig(t,k)
%5:43 PM 27/8/97
% defines modulation signal for rlmod control
global rlmod_sig n_rlmod
f=0; %dummy variable
if n_rlmod~=0
   rlmod_sig(:,k) = zeros(n_rlmod,1);
   rlmod_sig(1,k) = 0.5;
end
return
```

3.36 s_simu

3.36.1 Purpose:

Acts as driver for transient simulation

3.36.2 Syntax:

```
s_simu
```

3.36.3 Description:

s_simu is a MATLAB script file which calls the models of the POWER SYSTEM TOOLBOX to

- select a data file
- perform a load flow
- initialize the non-linear simulation models
- do a step-by-step integration of the non-linear dynamic equations to give the response to a user specified system fault

3.36.4 Global variables

pst_var

3.36.5 Algorithm:

s_simu is the driver for transient stability analysis in the Power System Toolbox. It requires an input data set comprising of the following specification matrices

3.36.5.1 obligatory

- **bus** a bus specification matrix - not necessarily solved
- **line** a line specification matrix - not necessarily solved
- **mac_con** a generator specification matrix
- **sw_con** a switching specification file

3.36.5.2 optional

- **exc_con** an exciter specification matrix
- **pss_con** a power system stabilizer specification matrix
- **tg_con** a turbine governor specification matrix
- **ind_con** an induction motor specification matrix
- **mld_con** a motor load specification matrix
- **load_con** a non conforming load specification matrix
- **svc_con** an SVC specification matrix
- **dcsp_con** a dc converter specification matrix
- **dcl_con** a dc line specification matrix
- **dcc_con** a dc control specification matrix

3.36.5.3 Preliminary

1. After reading the data, **svm_mgen** performs a load flow if requested, otherwise the solved load flow data is extracted from a **mat** file with the same name as the data file.
   If the data contains dc specification files, a combined ac/dc load flow is performed.
2. The data is organized by calling the index m-files. These check to see which data is available.

3.36.5.4 Initialization

The non-linear models are initialized at the operating point set by the solved load flow. The induction motor, SVC and HVDC models are initialized before a reduced network admittance matrix is constructed since they alter the entries in the solved load flow bus specification matrix.
Reduced admittance matrices are constructed, using **red_ybus**, which relate the currents injected into the generators and motors to the internal generator and motor voltages and the voltages at the non conforming load and SVC buses (see **red_ybus**) under the fault conditions specified in **sw_con**.
The time vector **t** is defined based on the fault timing and time steps specified in **sw_con**. Switching points occur at the times specified in **sw_con**. To achieve this, the specified time steps are a guide only. The closest smaller time step which gives the required switching points is substituted for the time step specified.

3.36.5.5 Simulation

A predictor-corrector algorithm is used for the step-by-step integration of the system equations. At each time step

1. A network interface calculation is performed **-** flag = 1 in the device models. The non-linear equations for the load at the non-conforming load buses are solved to give the voltage at these buses. The current injected by the generators and absorbed by the motors is calculated from the reduced admittance matrix appropriate to the specified fault condition at that time step based on the machine internal voltages and the non-conforming load bus voltages.
2. The rates of change of the dynamic device model state variables is calculated **-** flag = 2 in the device models.
3. A predictor integration step is performed which gives an estimate of the states at the next time step.
4. A second network interface step is performed.
5. The rates of change of the dynamic device model state variables are recalculated.
6. A corrector integration step is performed to obtain the final value of the states at the next time step.

All calculations are performed using the MATLAB vector calculation facility. This results in a simulation time which is largely a function of the number of time steps. The time increases only slightly with the system size. However, in most simulations there are at least 500 time steps, and simulation is quite time consuming .
After every ten simulation time steps, the response of the bus voltage magnitude at the fault bus is shown on the screen. This allows the user to abort simulations which are clearly unsatisfactory (press control-c to abort the simulation).
At the completion of the simulation, a menu of plots is presented to the user.
Many other variables are available for plotting if required. These include

- all dynamic states
- induction motor active and reactive powers (p_mot and q_mot)
- generator terminal voltage magnitudes (eterm)
- bus voltages (magnitude: abs(bus_v); angle: angle(bus_v))
- HDVC variables, Vdc, i_dc, alpha, gamma, dc control states, dc line states

For example, to plot all the generator terminal voltages against time use **plot(t,eterm)**
This algorithm is implemented in the M-file **s_simu** in the POWER SYSTEM TOOLBOX.

3.37 Example

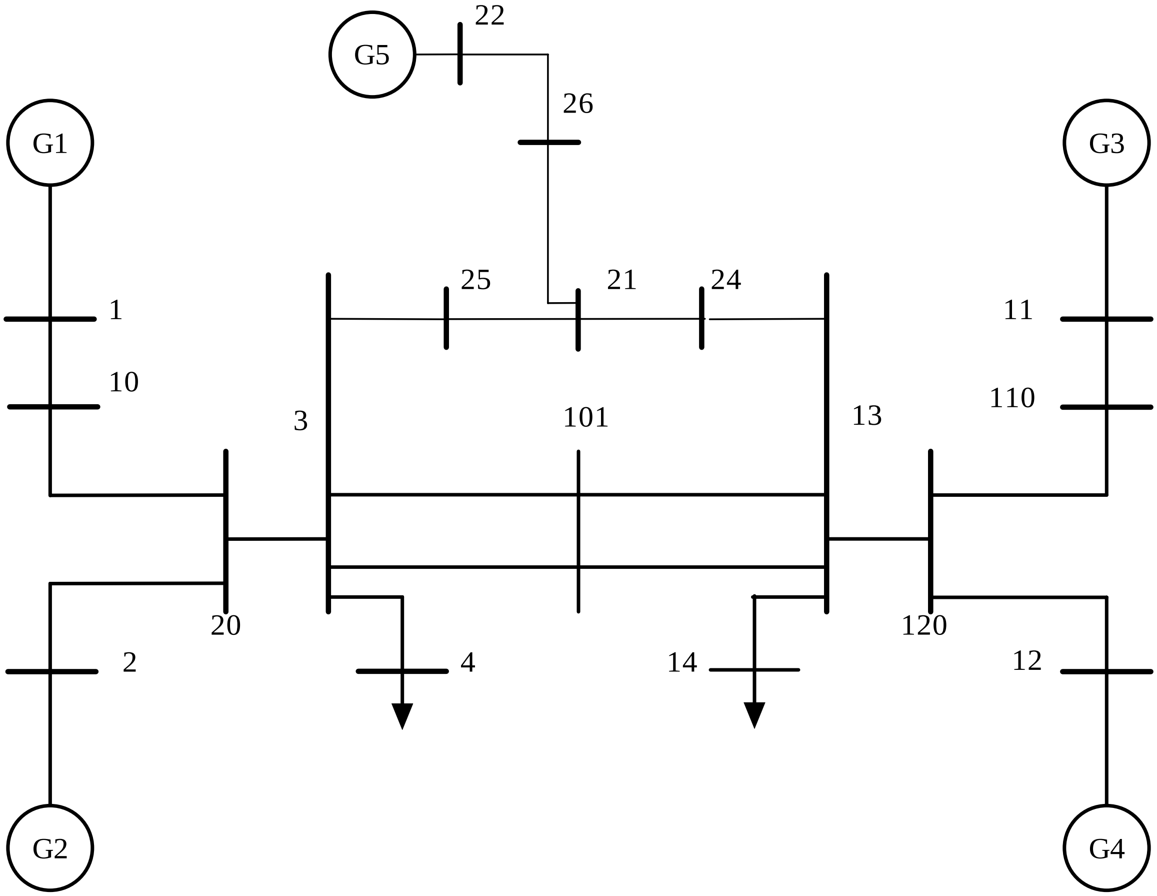


Figure 41 Two-Area System with added Load Area

The system shown in Figure 41 has the following data set.

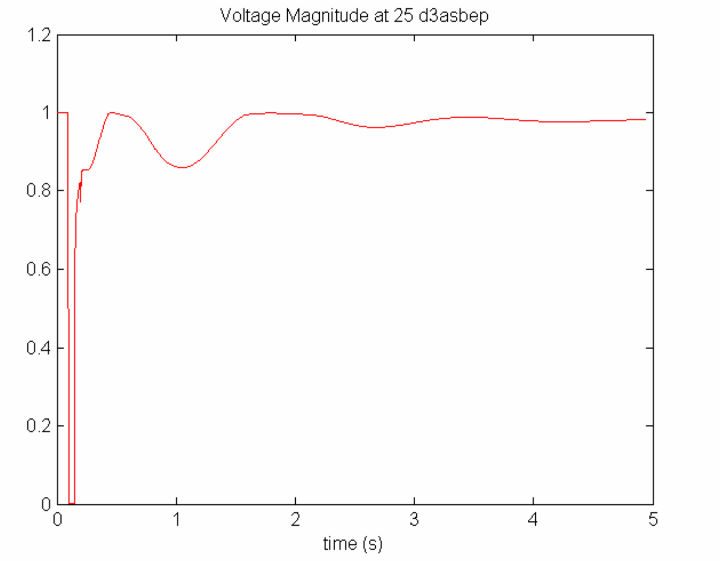
```
bus = [...
1  1.03    18.5   7.00   1.61  0.00  0.00  0.00  0.00  1  99.0  -99.0 22.0  1.1  .9;
2  1.01    8.80   7.00   1.76  0.00  0.00  0.00  0.00  2  99.0  -99.0 22.0  1.1  .9;
3  0.9781  -6.1   0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0  500.0  1.5  .5;
4  0.95    -10    0.00   0.00  10.0  1.00  0.00  0.00  3  0.0   0.0  115.0  1.05 .95;
10 1.0103  12.1   0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0  230.0  1.5  .5;
11 1.03    -6.8   7.00   1.49  0.00  0.00  0.00  0.00  2  99.0  -99.0 22.0  1.1  .9;
12 1.01    -16.9  7.50   1.39  0.00  0.00  0.00  0.00  2  99.0  -99.0 22.0  1.1  .9;
13 0.9899  -31.8  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0  500.0  1.5  .5;
14 0.95    -38    0.00   0.00  15.0  1.00  0.00  0.00  3  0.0   0.0  115.0  1.05 .95;
20 0.9876    2.1  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0  230.0  1.5  .5;
21 1.0       0    0.00   0.00  5.0   2.0   0.00  0.0   3  0.00  0.00 115.0  1.5  .5;
22 1.0       0    1.50   1.5   0.00  0.00  0.00  0.00  2  99.0  -99.0 18.0  1.1  .9;
24 1.0       0    0      0     0     0     0     0     3  0     0    500.0  1.5  .5;
25 1.0       0    0      0     0     0     0     0     2  0     0    500.0  1.5  .5;
26 1.0       0    0      0     0     0     0     0     3  0     0    115.0  1.5  .5;
101 1.05    -19.3  0.00   8.00  0.00  0.00  0.00  0.00 2  99.0  -99. 500.0  1.5  .5;
110 1.0125  -13.4  0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5;
120 0.9938  -23.6  0.00   0.00  0.00  0.00  0.00  0.00 3  0.0   0.0  230.0  1.5  .5 ];
line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.05;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
```

```
3   25  0.011   0.110    0.1925  1.0  0  0   0   0 ;
13  24  0.019   0.19     0.3     1.0  0  0   0   0 ;
22  26  0.0     0.05     0.0     1.0  0  0   0   0 ;
24  21  0.0     0.01     0.0     1.0  0  0   0   0 ;
25  21  0.0     0.01     0.0     1.0  0  0   0   0 ;
26  21  0.02    0.2      0.375   1.0  0  0   0   0 ;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.005    0.00    1.0  0. 1.2 0.8 0.05;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];
mac_con = [ ...

1 1 1000 0.200  0.0025  1.8  0.30  0.25 8.00  0.03...
                       1.7   0.55  0.25 0.4   0.05...
                       6.5   13    0    1;
2 2 1000 0.200  0.0025  1.8  0.30  0.25 8.00  0.03...
                       1.7   0.55  0.25 0.4   0.05...
                       6.5   13    0    2;
3 11 1000 0.200  0.0025  1.8 0.30  0.25 8.00  0.03...
                       1.7   0.55  0.25 0.4   0.05...
                       6.5   13    0    11;
4 12 1000 0.200  0.0025  1.8 0.30  0.25 8.00  0.03...
                       1.7   0.55  0.25 0.4   0.05...
                       6.5   13    0    12;
5 22 300 0.200  0.0025 1.8   0.3   0.25 5.00 0.03...
                       1.7   0.55  0.25 0.4  0.05...
                       5.0   10.0  0    22];
exc_con = [...
0 1 0.05 200.0  0     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 2 0.05 200.0  0     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 3 0.05 200.0  0     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 4 0.05 200.0  0     0     0    5.0  -5.0...
    0    0      0     0     0    0    0      0      0    0   0;
0 5 0.02 50.0   0.02  0.1   0.5  5.0  -2.0...
    0    0      0     0     0    0    0      0      0    0   0];
pss_con = [...
1 1 300.0  20.0  0.06  0.04  0.08  0.04  0.2  -0.05;
1 2 300.0  20.0  0.06  0.04  0.08  0.04  0.2  -0.05;
1 3 300.0  20.0  0.06  0.04  0.08  0.04  0.2  -0.05;
1 4 300.0  20.0  0.06  0.04  0.08  0.04  0.2  -0.05;
1 5 100.0  20.0  0.06  0.04  0.08  0.04  0.05 -0.01];
tg_con = [...
1  1  1  1.0  25.0  0.1  0.5 0.0 1.25 5.0;
1  2  1  1.0  25.0  0.1  0.5 0.0 1.25 5.0;
1  3  1  1.0  25.0  0.1  0.5 0.0 1.25 5.0;
1  4  1  1.0  25.0  0.1  0.5 0.0 1.25 5.0];

ind_con = [ ...
1  21  240.0 .001 .1 4 .015 .1  0.6 0 0 0 0 0 0.4];

mld_con = [ ...
1  21  .1  1  .7  5];

load_con = [21  0  0  0  0];

svc_con = [1  21 100  1  0  50  0.02];

sw_con = [...
0      0    0    0    0    0    0.01;%sets initial time step
0.1    25   3    0    0    0    0.005; %apply three phase fault at bus 25, on line 25-3
0.15   0    0    0    0    0    0.005; %clear fault at bus 25
0.20   0    0    0    0    0    0.005; %clear remote end
5.0    0    0    0    0    0    0.0 ]; % end simulation
```

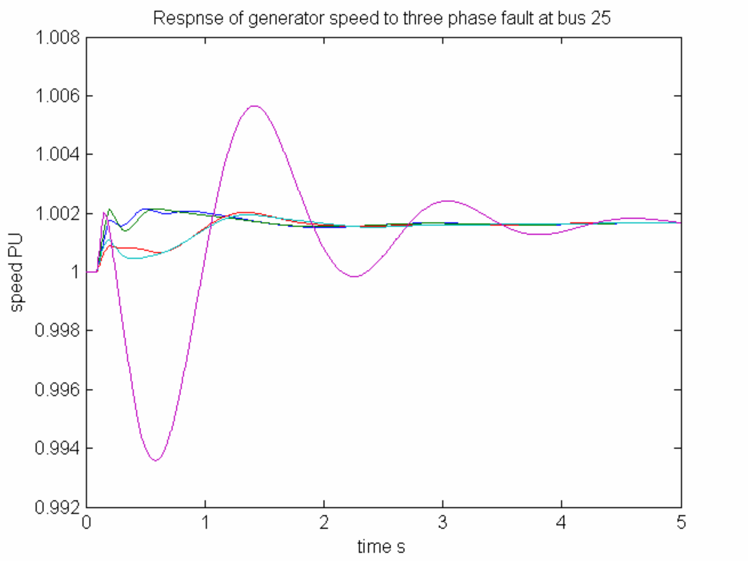
The system has 5 generators at buses 1, 2, 11, 12 and 22. All generators have simple exciters and a power system stabilizer. The first four generators have turbine/governors modelled. There are three load buses, 4, 14 and 21. The load at bus 21 has 40% motor content, the remaining loads are constant impedance. There is an SVC set to control the voltage at bus 21.
At 0.1s, a three phase fault is applied at bus 25 on line 3-25. At 0.15 s the line is disconnected at bus 25. The fault persists until 0.2 s when the line is disconnected from bus 3.
The simulation runs for 5 s. The time step is small (0,005 s) throughout because of the induction motor model.

It is good practice to run a simulation for a short time before applying a fault. This checks that the system has a satisfactory, stable initial condition.

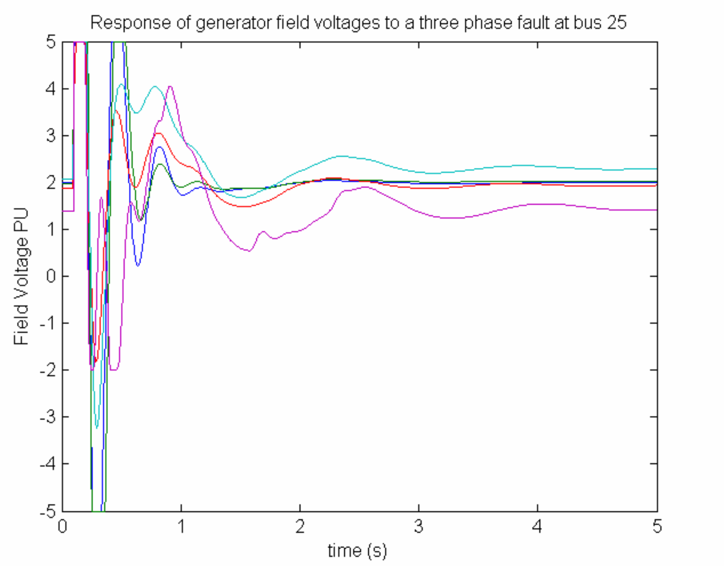
The following plots illustrate the system's behaivior
Fault bus voltage screen plot



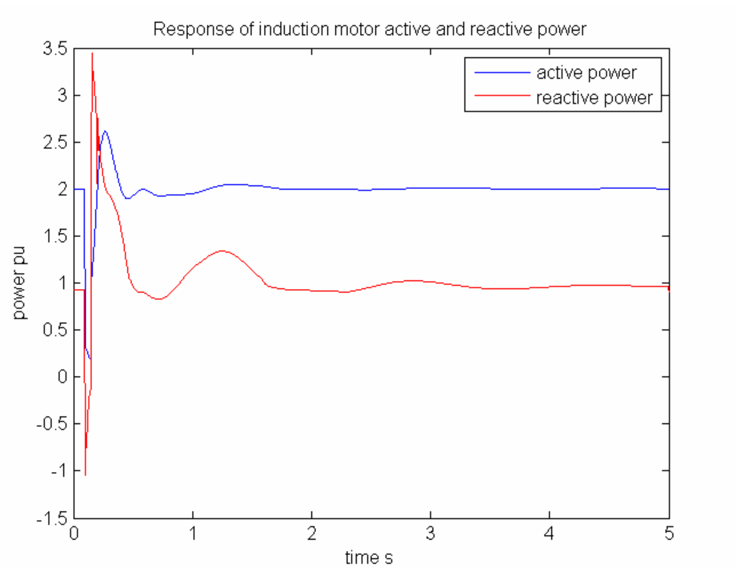
Generator speed deviations



exciter output voltages



induction motor active and reactive load power



3.38 smpexc

3.38.1 Purpose:

Models simplified excitation systems

3.38.2 Synopsis:

`smpexc(i,k,bus,flag)`

3.38.3 Description:

smpexc(i,k,bus,flag) models the simplified excitation system shown in Figure 1. The m.file **pst_var.m** containing all the global variables required for **smpexc** should be loaded in the program calling **smpexc**.

3.38.4 Inputs:

i the number of the exciter
if **i** = 0 all simple exciter computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the field voltage of the synchronous machine is set to the exciter output voltage.
- The rates of change of the exciter states are calculated when **flag** = 2, using the generator terminal voltage and the external system values at the time specified by **k**

3.38.5 Global Variables:

Efd	E_{fd}	exciter output voltage
V_R	V_R	regulator output voltage in pu
V_A	V_A	regulator output voltage in pu
V_As	V_{As}	regulator voltage state variable in pu
R_f	R_f	stabilizing transformer state variable
V_FB	V_{FB}	feedback from stabilizing transformer
V_TR	V_{TR}	voltage transducer output in pu
V_B	V_B	potential circuit voltage output in pu
dEfd	dE_{fd}/dt	
dV_R	dV_R/dt	
dV_As	dV_{As}/dt	
dR_f	dR_f/dt	
dV_TR	dV_{TR}/dt	
exc_sig	V_{sup}	supplementary input signal to exciter ref input
exc_pot		matrix of internally set exciter constants
exc_con		matrix of exciter data set by user
smp_idx		index of simple exciters, i.e., exc_con(smp_idx,:)
n_smp		number of simple exciters

3.38.6 Data Format:

The exciter data are contained in the **i**$^{\mathbf{th}}$ row of the matrix variable **exc_con**. The data format for **smpexc** is shown in Table 15.

Table 15 Data format for smpexc

column	Variable	unit
1	exciter type	0
2	generator number	
3	transducer filter time constant T_R	sec
4	voltage regulator gain K_A	pu
5	voltage regulator time constant T_A	sec

6	transient gain reduction time constant T_B	sec
7	transient gain reduction time constant T_C	sec
8	maximum voltage regulator output V_{Rmax}	pu
9	minimum voltage regulator output V_{Rmin}	pu

If T_B is set to zero, then there will be no transient gain reduction.

3.38.7 Algorithm:

Based on the exciter block diagram, the exciter is initialized using the generator field voltage E_{fd} to compute the state variables. In the network interface computation, the exciter output voltage is converted to the field voltage of the synchronous machine. In the dynamics calculation, generator terminal voltage and the external signal is used to calculate the rates of change of the excitation system states.

This algorithm is implemented in the M-file **smpexc** in the POWER SYSTEM TOOLBOX.

See also: loadflow,pst_var,exc_dc12,exc_st3,mac_tra,mac_sub.

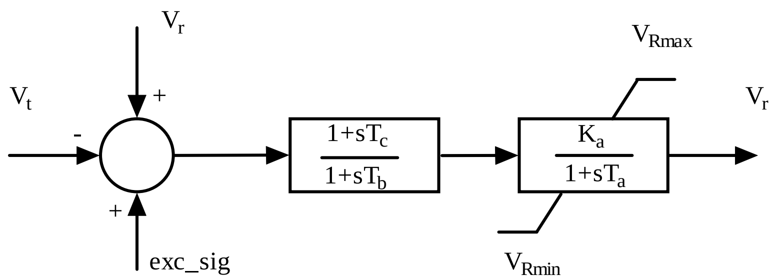


Figure 42 Simple Exciter

3.39 statef

3.39.1 Purpose:

Calculates the frequency response from system equations in state space form

3.39.2 Syntax:

```
[f,ymag,yphase]=statef(a,b,c,d,fstart,fstep,fend)
```

3.39.3 Description:

statef calculates the frequency response between a single input and a single output from the state space model of the system. It is used in **pss_des**.

3.39.4 Inputs:

a the state matrix of the system for which frequency response is to be calculated
b the input vector
c the output row vector
d the feed forward between input and output.
fstart the starting frequency (Hz)
fstep the frequency step (Hz)
fend the end frequency (Hz)

3.39.5 Outputs:

f the frequency vector
ymag the output magnitude vector
yphase the output phase vector (degrees)

3.39.6 Algorithm:

This algorithm is implemented in the M-file **statef.m** in the POWER SYSTEM TOOLBOX.

3.40 step_res

3.40.1 Purpose:
Step response from state space system definition

3.40.2 Synopsis:
```
[res t] = step_res(a,b,c,d,v_in,tmax)
```

3.40.3 Description:
step_res computes the step response from a state space system description.

$\dot{x} = ax + bu$

$y = cx + du$

The response is plotted on successful completion.

3.40.4 Inputs:

a	the state matrix of size ns by ns
b	the input matrix of size nx by nin
c	the out put matrix of size nout by nx
d	the feed forward matrix of size nout by nin
v_in	a column vector of length(nin) specifying the magnitude of the applied step
tmax	the maximum time of the response calculation (s)

3.40.5 Output:

res	a matrix of the response size(nx by length(t))
t	a vector of time

3.40.6 Algorithm:
The time step is chosen from the eigenvalues of a to give 5 time steps in the largest frequency or over the time constant of the fastest exponential decay.

The matrix exponential of (**a * t_step**) is calculated using **expm**.
The response is y is calculated from

$x(:,k) = \exp(a*t\_step)\ x(:,k-1) - (I + \exp(a*t\_step))\mathrm{inv}(a)\ b\ v\_in$

$y(:,k) = c\ x(:,k) + d\ v\_in$

The state matrices for a power system may be computed using **svm_mgen**.

3.41 svc

3.41.1 Purpose:
Models static VAR control systems

3.41.2 Synopsis:
```
bus_new = svc(i,k,bus,flag,v_sbus)
```

3.41.3 Description:
svc(i,k,bus,flag,v_sbus) contains the equations of a static var control system [1] for the initialization, machine interface and dynamics computation of the **i**$^{\mathbf{th}}$ static var system.
A system oscillation damping control signal can be input to the static var system through the global variable **svc_sig** [1].
The m.file **pst_var.m** containing all the global variables required for **svc** should be loaded in the program calling **svc**.

3.41.4 Inputs:

i	the number of the SVC if **i** = 0 all SVC computations are performed using MATLAB vector methods. **This is the preferred mode.**
k	the integer time step in a simulation In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix
flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- There is no need to perform a network interface calculation for **svc**
- The rates of change of the SVC state is calculated when **flag** = 2, using the SVC terminal voltage value and the modulating signal **svc_sig** at the time specified by **k**

v_sbus The SVC bus voltage

3.41.5 Output:

bus_new On initialization **bus_new = bus** with the reactive generation at the SVC buses set to zero
In other modes **bus_new = bus**

3.41.6 Global Variables

3.41.6.1 System variables

basmva system base MVA
bus_int array to store internal bus ordering

3.41.6.2 Static VAR Compensator Variables

B_cv B_{cv} svc susceptance in pu
dB_cv dB_{cv}/dt
svc_sig V_{sup} supplementary signal into the reference input
svc_con matrix of svc parameters supplied by user
svc_pot internally calculated matrix of svc constants
n_svc number of svcs
svc_idx index of svcs included in **load_con**

3.41.7 Data Format

The static var system data is contained in the **ith** row of the matrix **svc_con**. The data format for **svc_con** is given in Table 16.

Table 16 Data format for SVC

column	variable	unit
1	svc number	
2	bus number	
3	svc base MVA	MVA
4	maximum susceptance B_{cvmax}	pu
5	minimum susceptance B_{cvmin}	pu
6	regulator gain K_R	pu
7	regulator time constant T_R	sec
8	compensator lag time constant T_B	sec
9	compensator lead time constant T_B	sec
10	Fraction of bus B picked up by svc	

3.41.8 Algorithm:

To use the **svc** function, the static var system buses must be declared via **load_con** as non-conforming load buses with zero constant power and current components. The buses should be set to be generator buses, since the SVC picks up the reactive power generation to determine its initial susceptance setting. In the network interface computation, the static var system output is used to adjust the reduced network admittance matrix to solve for the bus voltages. This function is automatically performed in **nc_load**. In the

dynamics calculation, the rate of change of the SVC state is adjusted according to the voltage error. An anti-windup limit is used to reset the susceptance state variable.
This algorithm is implemented in the M-file **svc** in the POWER SYSTEM TOOLBOX.
See also: nc_load, pst_var.

3.41.9 Reference:

1. E. V. Larsen and J. H. Chow, "SVC Control Concepts for System Dynamic Performance," in *Application of Static Var Systems for System Dynamic Performance*, IEEE Publications 87TH0187-5-PWR, 1987.

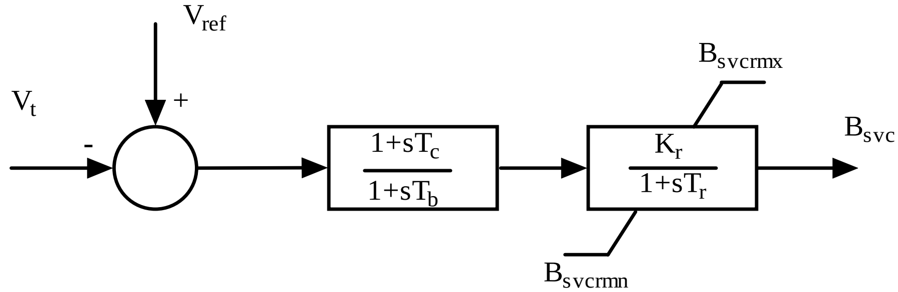


Figure 43 SVC Model Block Diagram

3.42 svc_indx

3.42.1 Purpose:

Forms indexes for svc calculation and checks for correct svc calling.

3.42.2 Syntax:

svc_indx

3.42.3 Global Variables:

3.42.3.1 Non Conforming Load Variables

load_con matrix of non conforming load parameters set by user

3.42.3.2 Static VAR Compensator Variables

svc_con matrix of svc parameters supplied by user
n_svc number of svcs
svc_idx index of svcs included in **load_con**

3.42.4 Algorithm:

Called before **svc** to set index. Finds the number of **svc's** and checks to see if they are declared correctly on **load_con**.
This algorithm is implemented in the M-file **svc_indx.m** in the POWER SYSTEM TOOLBOX.

3.43 svm_mgen

3.43.1 Purpose:

Forms the state matrices of a power system model, linearized about an operating point set by a load flow and performs modal analysis.

3.43.2 Syntax:

svm_mgen

3.43.3 Description:

svm_mgen is a MATLAB script file which calls the models of the POWER SYSTEM TOOLBOX to

- select a data file
- perform a load flow
- form a linearized model by perturbing each state in turn
- do a modal analysis of the system

3.43.4 Global variables

pst_var

3.43.5 Algorithm:

svm_mgen is the driver for small signal stability analysis in the Power System Toolbox. It requires an input data set comprising the following specification matrices

obligatory

- **bus** a bus specification matrix **-** not necessarily solved
- **line** a line specification matrix **-** not necessarily solved
- **mac_con** a generator specification matrix

optional

- **exc_con** an exciter specification matrix
- **pss_con** a power system stabilizer specification matrix
- **tg_con** a turbine governor specification matrix
- **ind_con** an induction motor specification matrix
- **mld_con** a motor load specification matrix
- **load_con** a non conforming load specification matrix
- **svc_con** an SVC specification matrix
- **ibus_con** an infinite bus specification vector
- **lmon_con** a line monitor specification vector

3.43.5.1 Preliminary

1. After reading the data, **svm_mgen** performs a load flow: the user is given the opportunity to revise **bus** and **line** to produce a post-fault, rather than pre-fault load flow.
2. The data is organized by calling the index m-files. These check to see which data is available
3. The number of system states are determined and a permutation matrix is formed which organizes the order of the states in the state matrix. In general, the states in the state matrix are ordered as follows:
   I. The generator and generator control states-in internal generator number order
   II. The induction motor states in internal induction motor order
   III. The svc states

   The number of states in each device depends on the model data. However, the internal state matrices, as defined in **pst_var** have dimensions set only by the number of devices.

3.43.5.2 Initialization

All the devices are initialized at the operating point set by a system load flow. This gives the initial non-linear state vector. The infinite buses have no states, but their internal voltages are calculated from the original generator data and then stored. These voltages remain unchanged in following computations. The induction motor initialization (see **mac_ind**) determines the motor reactive power demand. This is subtracted from the bus reactive power load.

3.43.5.3 State matrix formation

Each state is perturbed in turn by a small value pert = (max(0.0001,0.001*state)). The rate of change **d_mat** of all the states is calculated. When the $\mathbf{i^{th}}$ state is perturbed the $\mathbf{i^{th}}$ column of the state matrix is calculated as

$$a\_mat(:,i) = p\_mat * d\_vector / pert$$

where a_mat is the state matrix and p_mat is the permutation matrix.

The input, output and feed forward matrices (**b**, **c**, **d**) are calculated at the same time for

inputs: exciter reference voltage $\mathbf{b_{vr}}$: turbine/governor power reference $\mathbf{b_{pr}}$: load modulation **b_lmod**: reactive load modulation **b_rlmod**

outputs: generator speed, $\mathbf{c_{sp}}$: generator electrical torque, $\mathbf{c_t}$: generator electrical power $\mathbf{c_p}$: line real and reactive power flow for monitored lines, **cpf1**, **cqfl**, **cpf2**, **cqf2.**
The monitored lines are specified in the input data by the vector **lmon_con** which has length equal to the number of lines, entries of unity in the positions corresponding to the monitored lines and zero elsewhere.

feed forward: from V_{ref} to electrical torque, $\mathbf{d_{vrt}}$;to electrical power $\mathbf{d_{vrp}}$: from P_{ref} to electrical torque, $\mathbf{d_{prt}}$; to electrical power, $\mathbf{d_{prp}}$

3.43.5.4 Modal Analysis

Modal analysis is performed on the state matrix using the MATLAB **eig** function. This and storage considerations limits the total states of the modelled system to about 800.
The eigenvalues and right eigenvectors are calculated using **eig**. The left eigenvector is obtained by inverting the right eigenvector. The eigenvalues are ordered using **sort** and the columns of the eigenvector matrix are consistently permutated, They are stored in

l - eigenvalues vector
u - right eigenvector matrix (**i**$^{\mathbf{th}}$ column is the right eigenvector associated with **l**(i))
v - left eigenvector matrix (**i**$^{\mathbf{th}}$ row is the left eigenvector associated with **l**(i))

The participation vectors are stored as the columns of **p**. These values give the sensitivities of the eigenvalues to changes in the diagonal element of the state matrix. They are formed from

$$p(i,j) = u(i,j) * v(j,i)$$

The normalized participation vectors (the maximum modulus in each column is scaled to unity) are calculated and stored in **p_norm**. Values having a magnitude less than 0.1 are set to zero. The statement **sparse(abs(p_norm(:,j)))** indicates those states most influential in the control of the j^{th} eigenvalue.
Each of the columns of **p** and **p_norm** is associated with an eigenvalue, each of the rows is associated with a state.
Data describing the structure of the state matrix is also available.
state(k) - gives the number of states associated with the k^{th} generator
mac_state - has three columns

column 1 gives the overall state number

column 2 gives the state number within a particular generator and its controls

Generator
1 - δ
2 - ω
3 - E'_q
4 - ψ''_d
5 - E'_d
6 - ψ''_q

Exciter
7 - V_TR
8 - V_As
9 - V_R
10 - Efd
11 - R_f

Power System Stabilizer
12 - pss1
13 - pss2
14 - pss3

Turbine Governor
15 - tg1
16 - tg2
17 - tg3
18 - tg4
19 - tg5

column 3 gives the corresponding generator number

Thus, there are 19 possible states associated with each generator.
There are three states for each induction motor(v'_d, v'_q and s) which follow the generator states in the state vector in motor number order.
Each induction generator has three states which follow the induction motor states in the state vector in induction generator order.
Each svc has a single state (B_cv). The svc states follow the machine states in svc number order.
Each load modulation control has a single state (lmod_st). The load modulation states states follow the svc states in load modulation control number order.
Each HVDC link may have up to 5 states, these follow the svc states in the order, v_conr, v_coni, i_dcr, i_dci, v_dcc. If there is no line capacitor, the HVDC link model has only the first three states.
Thus the maximum number of states, which is the length of d_vector, is

$$19 * n\_mac + 3 * n\_mot + 3 * n\_ig + n\_svc + n\_lmod + n\_rlmod + 5 * n\_dcl$$

where n_mac is the number of generators, n_mot is the number of induction motors, n_ig is the number of induction generators, n_svc is the number of svcs, n_lmod is the number of load modulation controls, n_rlmod is the number of reactive load modulation controls and n_dcl is the number of HVDC lines.

3.43.6 Example

A two area system model data is contained in **d2adcensvc.m**. The m file listing is

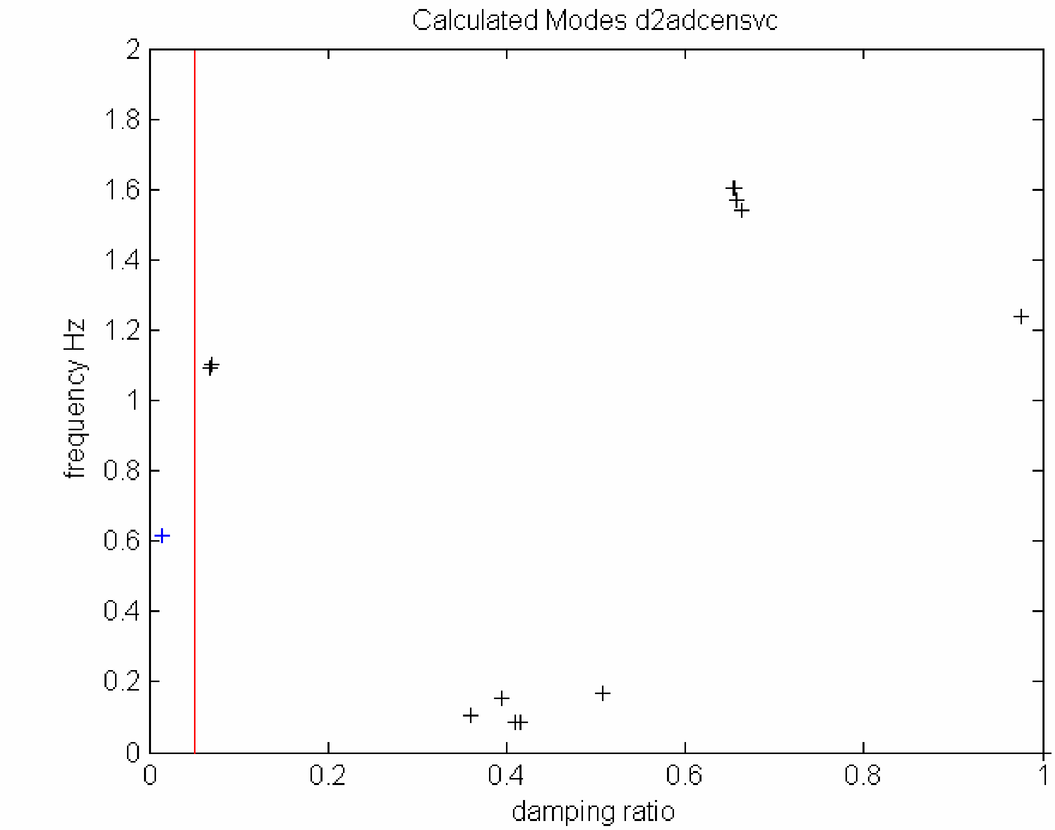
```
% Two Area Test Case
% subtransient generator models
% dc exciters
% turbine/governor
% 50% constant current/50% constant impedance loads
% reactive load modulation at bus 101

% bus data format
% bus:
% col1 number
% col2 voltage magnitude(pu)
% col3 voltage angle(degree)
% col4 p_gen(pu)
% col5 q_gen(pu),
% col6 p_load(pu)
% col7 q_load(pu)
% col8 G shunt(pu)
% col9 B shunt(pu)
% col10 bus_type
%       bus_type - 1, swing bus
%                - 2, generator bus (PV bus)
%                - 3, load bus (PQ bus)
% col11 q_gen_max(pu)
% col12 q_gen_min(pu)
% col13 v_rated (kV)
% col14 v_max  pu
% col15 v_min  pu

bus = [...
    1  1.03     18.5    7.00   1.61  0.00  0.00  0.00  0.00 1   5.0  -2.0  22.0   1.1  .9;
    2  1.01      8.80   7.00   1.76  0.00  0.00  0.00  0.00 2   5.0  -2.0  22.0   1.1  .9;
    3  1.0      -6.1    0.00   0.00  0.00  0.00  0.00  3.00 2   0.0   0.0  230.0  1.5  .5;
    4  0.97    -10      0.00   0.00  9.76  1.00  0.00  0.00 3   0.0   0.0  115.0  1.05 .95;
   10 1.0103   12.1     0.00   0.00  0.00  0.00  0.00  0.00 3   0.0   0.0  230.0  1.5  .5;
   11 1.03     -6.8     7.16   1.49  0.00  0.00  0.00  0.00 2   5.0  -2.0  22.0   1.1  .9;
   12 1.01    -16.9     7.00   1.39  0.00  0.00  0.00  0.00 2   5.0  -2.0  22.0   1.1  .9;
   13 1.0     -31.8     0.00   0.00  0.00  0.00  0.00  5.00 2   0.0   0.0  230.0  1.5  .5;
```

```
   14 0.97    -38      0.00   0.00 17.67  1.00  0.00  0.00 3   0.0   0.0  115.0  1.05 .95;
   20 0.9876    2.1    0.00   0.00  0.00  0.00  0.00  0.00 3   0.0   0.0  230.0  1.5  .5;
  101 1.05    -19.3    0.00   1.00  0.00  0.00  0.00  1.00 2   2.0   0.0  230.0  1.5  .5;
  110 1.0125  -13.4    0.00   0.00  0.00  0.00  0.00  0.00 3   0.0   0.0  230.0  1.5  .5;
  120 0.9938  -23.6    0.00   0.00  0.00  0.00  0.00  0.00 3   0.0   0.0  230.0  1.5  .5 ;
];


% line data format
% line: from bus, to bus, resistance(pu), reactance(pu),
%       line charging(pu), tap ratio, tap phase, tapmax, tapmin, tapsize

line = [...
1       10  0.0     0.0167  0.00    1.0  0. 0.  0.  0.;
2       20  0.0     0.0167  0.00    1.0  0. 0.  0.  0.;
3        4  0.0     0.005    0.00   1.0  0. 1.2 0.8 0.05;
3       20  0.001   0.0100  0.0175  1.0  0. 0.  0.  0.;
3      101  0.011   0.110   0.1925  1.0  0. 0.  0.  0.;
3      101  0.011   0.110   0.1925  1.0  0. 0.  0.  0.;
10      20  0.0025  0.025   0.0437  1.0  0. 0.  0.  0.;
11     110  0.0     0.0167  0.0     1.0  0. 0.  0.  0.;
12     120  0.0     0.0167  0.0     1.0  0. 0.  0.  0.;
13      14  0.0     0.005   0.00    1.0  0. 1.2 0.8 0.05;
13     101  0.011   0.11    0.1925  1.0  0. 0.  0.  0.;
13     101  0.011   0.11    0.1925  1.0  0. 0.  0.  0.;
13     120  0.001   0.01    0.0175  1.0  0. 0.  0.  0.;
110    120  0.0025  0.025   0.0437  1.0  0. 0.  0.  0.;
];

% Machine data format
% Machine data format
%       1. machine number,
%       2. bus number,
%       3. base mva,
%       4. leakage reactance x_l(pu),
%       5. resistance r_a(pu),
%       6. d-axis sychronous reactance x_d(pu),
%       7. d-axis transient reactance x'_d(pu),
%       8. d-axis subtransient reactance x"_d(pu),
%       9. d-axis open-circuit time constant T'_do(sec),
%      10. d-axis open-circuit subtransient time constant
%                 T"_do(sec),
%      11. q-axis sychronous reactance x_q(pu),
%      12. q-axis transient reactance x'_q(pu),
%      13. q-axis subtransient reactance x"_q(pu),
%      14. q-axis open-circuit time constant T'_qo(sec),
%      15. q-axis open circuit subtransient time constant
%                 T"_qo(sec),
%      16. inertia constant H(sec),
%      17. damping coefficient d_o(pu),
%      18. dampling coefficient d_1(pu),
%      19. bus number
%
% note: all the following machines use sub-transient model
mac_con = [ ...

1   1 900 0.200  0.0025  1.8  0.30  0.25 8.00  0.03...
                         1.7  0.55  0.25 0.4   0.05...
                         6.5  0  0  1;
2   2 900 0.200  0.0025  1.8  0.30  0.25 8.00  0.03...
                         1.7  0.55  0.25 0.4   0.05...
                         6.5  0  0  2;
3  11 900 0.200  0.0025 1.8  0.30  0.25 8.00  0.03...
                        1.7  0.55  0.25 0.4   0.05...
                        6.5  0  0  11;
4  12 900 0.200  0.0025 1.8  0.30  0.25 8.00  0.03...
                        1.7  0.55  0.25 0.4   0.05...
                        6.5  0  0  12;
                ];
```

```
% all dc exciters, no pss
% col1 type
% col2 machine number
% col3 Tr
% col4 Ka
% col5 Ta
% col6 Tb
% col7 Tc
% col8 Vrmax
% col9 Vrmin
% col10 Ke
% col11 Te
% col12 E1
% col13 Se(E1)
% col14 E2
% col15 Se(E2)
% col16 Kf
% col17 Tf
% cols 18 to 20 required for exc_st3 only
exc_con = [...
1 1 0.01 46.0   0.06  0     0    1.0   -0.9...
    0.0  0.46   3.1   0.33  2.3  0.1   0.1   1.0    0    0    0;
1 2 0.01 46.0   0.06  0     0    1.0   -0.9...
    0.0  0.46   3.1   0.33  2.3  0.1   0.1   1.0    0    0    0;
1 3 0.01 46.0   0.06  0     0    1.0   -0.9...
    0.0  0.46   3.1   0.33  2.3  0.1   0.1   1.0    0    0    0;
1 4 0.01 46.0   0.06  0     0    1.0   -0.9...
    0.0  0.46   3.1   0.33  2.3  0.1   0.1   1.0    0    0    0];


% governor model
% tg_con matrix format
%column        data         unit
%  1    turbine model number (=1)
%  2    machine number
%  3    speed set point   wf        pu
%  4    steady state gain 1/R       pu
%  5    maximum power order  Tmax   pu on generator base
%  6    servo time constant   Ts    sec
%  7    governor time constant  Tc  sec
%  8    transient gain time constant T3 sec
%  9    HP section time constant   T4   sec
% 10    reheater time constant    T5    sec

tg_con = [...
1  1  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  2  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  3  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0;
1  4  1  25.0  1.0  0.1  0.5 0.0 1.25 5.0];


% non-conforming load
% col 1           bus number
% col 2           fraction const active power load
% col 3           fraction const reactive power load
% col 4           fraction const active current load
% col 5           fraction const reactive current load

load_con = [...
3   0  0   0   0;
4   0  0   .5  0;
13  0  0   0   0;
14  0  0   .5  0;
101 0  0   0  0];
disp('50% constant current/50% constant impedance load, reactive load modulation at bus 3
13 and 101')
```

```
%reactive load modulation
%col1 modulation number
%col2 bus number
%col3 MVA base
%col4 max susceptance
%col5 min susceptance
%col6 regulator gain
%col7 time constant
rlmod_con = [...
    1 101 200 1 -1 1 0.05;
    2  3  200 1 -1 1 0.05;
    3  13 200 1 -1 1 0.05;
];
%svc
% col 1           svc number
% col 2           bus number
% col 3           svc base MVA
% col 4           maximum susceptance Bcvmax(pu)
% col 5           minimum susceptance Bcvmin(pu)
% col 6           regulator gain
% col 7       regulator time constant (s)

svc_con = [...
    %1  101  200  1  -1  10  0.05 0.65 0.2;
    2    3   200   1  -1   10   0.05  0.65 0.2;
    %3    13  200   1  -1   10   0.05   0.65 0.2;
];

%Switching file defines the simulation control
% row 1 col1  simulation start time (s) (cols 2 to 6 zeros)
%       col7  initial time step (s)
% row 2 col1  fault application time (s)
%       col2  bus number at which fault is applied
%       col3  bus number defining far end of faulted line
%       col4  zero sequence impedance in pu on system base
%       col5  negative sequence impedance in pu on system base
%       col6  type of fault  - 0 three phase
%                            - 1 line to ground
%                            - 2 line-to-line to ground
%                            - 3 line-to-line
%                            - 4 loss of line with no fault
%                            - 5 loss of load at bus
%                            - 6 no fault
%       col7  time step for fault period (s)
% row 3 col1  near end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for second part of fault (s)
% row 4 col1  far end fault clearing time (s) (cols 2 to 6 zeros)
%       col7  time step for fault cleared simulation (s)
% row 5 col1  time to change step length (s)
%       col7  time step (s)
%
%
%
% row n col1 finishing time (s)  (n indicates that intermediate rows may be inserted)

sw_con = [...
0    0    0    0    0    0    0.01;%sets intitial time step
0.1  3    101  0    0    0    0.005; %thee phase fault
0.15 0    0    0    0    0    0.005556; %clear fault at bus 3
0.20 0    0    0    0    0    0.005556; %clear remote end
0.50 0    0    0    0    0    0.01; % increase time step
1.0  0    0    0    0    0    0.01; % increase time step
10.0 0    0    0    0    0    0]; % end simulation
% monitor all line flows
lmon_con = (1:length(line(:,1)))';
```

The frequency against damping ratio plot of the system's modes are shown on completion.



All modes are stable. One complex mode pair (+) has a damping ratio less than 0.05.

Note that this m file contains a switching file - this data is ignored in **svm_mgen**. The data file also contains a line monitor specification vector and so the output matrices for line flow are calculated. Selected results of calling **svm_mgen** with this data set are given below.
The total number of states is given by NumStates

```
NumStates = 54
```

The type of variable represented by each generator state can be found from mac_state

```
mac_state =

     1     1     1
     2     2     1
     3     3     1
     4     4     1
     5     5     1
     6     6     1
     7     7     1
     8     9     1
     9    10     1
    10    11     1
    11    21     1
    12    22     1
    13    23     1
    14     1     2
    15     2     2
    16     3     2
    17     4     2
    18     5     2
    19     6     2
    20     7     2
    21     9     2
    22    10     2
```

```
23    11     2
24    21     2
25    22     2
26    23     2
27     1     3
28     2     3
29     3     3
30     4     3
31     5     3
32     6     3
33     7     3
34     9     3
35    10     3
36    11     3
37    21     3
38    22     3
39    23     3
40     1     4
41     2     4
42     3     4
43     4     4
44     5     4
45     6     4
46     7     4
47     9     4
48    10     4
49    11     4
50    21     4
51    22     4
52    23     4
```

Eigenvalues

```
[(1:NumStates)' l damp freq]

ans =

1          6.117e-006                    1                         0
2         -0.19612                       1                         0
3         -0.19847                       1                         0
4         -0.1985                        1                         0
5         -0.23396 -    0.52011i     0.41023                0.082777
6         -0.23396 +    0.52011i     0.41023                0.082777
7         -0.23734 -    0.52076i     0.41471                0.082881
8         -0.23734 +    0.52076i     0.41471                0.082881
9         -0.24917 -    0.64506i     0.36033                 0.10266
10        -0.24917 +    0.64506i     0.36033                 0.10266
11        -0.41623 -    0.96732i     0.39525                 0.15395
12        -0.41623 +    0.96732i     0.39525                 0.15395
13        -0.62353 -     1.0586i     0.50751                 0.16849
14        -0.62353 +     1.0586i     0.50751                 0.16849
15        -1.5607                        1                         0
16        -1.5934                        1                         0
17        -1.9364                        1                         0
18        -1.9701                        1                         0
19        -1.9709                        1                         0
20        -2.8372                        1                         0
21        -0.054603 -     3.8555i   0.014161                 0.61362
22        -0.054603 +     3.8555i   0.014161                 0.61362
23        -4.4197                        1                         0
24        -4.8999                        1                         0
25        -5.0457                        1                         0
26        -0.47282 -     6.8717i   0.068645                  1.0937
27        -0.47282 +     6.8717i   0.068645                  1.0937
28        -0.48929 -      6.936i   0.070368                  1.1039
29        -0.48929 +      6.936i   0.070368                  1.1039
30        -10.072                        1                         0
31        -10.072                        1                         0
32        -10.097                        1                         0
33        -10.11                         1                         0
34        -8.5811 -     9.6634i      0.66399                   1.538
35        -8.5811 +     9.6634i      0.66399                   1.538
36        -8.5917 -     9.8651i      0.65676                  1.5701
37        -8.5917 +     9.8651i      0.65676                  1.5701
38        -8.72   -     10.067i      0.65472                  1.6023
39        -8.72   +     10.067i      0.65472                  1.6023
40        -8.7223 -      10.08i      0.65434                  1.6043
41        -8.7223 +      10.08i      0.65434                  1.6043
42        -29.778                        1                         0
43        -33.056                        1                         0
44        -34.773                        1                         0
45        -35.409                        1                         0
46        -36.09                         1                         0
47        -35.579 -     7.7753i      0.97694                  1.2375
48        -35.579 +     7.7753i      0.97694                  1.2375
49        -37.075                        1                         0
50        -37.161                        1                         0
51        -99.995                        1                         0
52        -99.996                        1                         0
53        -99.998                        1                         0
54        -99.998                        1                         0
```

The eigenvalues are ordered from minimum to maximum modulus using the MATLAB function **sort**.

The Nature of the Modes

All modes are stable **-** they have negative real parts. Some are real and some are complex.
The first eigenvalue is effectively zero. This is characteristic of power systems and represents the non-uniqueness of the bus voltage angles.
There are 13 complex conjugate pairs of complex eigenvalues which represent the oscillatory system modes.

The least damped modes are 21 and 22 which have a damping ratio of 0.014161 and a frequency of 0.61362 Hz..
The states associated with this mode may be determined by

```
 sparse(abs(p_norm(:,22)))

ans =

   (1,1)        0.53675
   (2,1)        0.56796
  (14,1)        0.32977
  (15,1)        0.34894
  (27,1)        0.94504
  (28,1)              1
  (40,1)        0.54761
  (41,1)        0.57946
```

This indicates that the state with the largest normalized participation factor is state 28, which is the speed of generator 3. The nature of the other states may be determined using mac_state: states 1 and 2 are the rotor states for generator 1, states 14 and 15 are the rotor states for generator 2, 27 and 28 are the rotor angle states of generator 3 and states 40 and 41 are the rotor states for generator 4. This mode is an inter-area mode associated with all the system's generators.

3.44 tcsc

3.44.1 Purpose

Models a Thyristor Controlled Series Reactor

3.44.2 Synopsis

`tcsc(i,k,bus,flag)`

3.44.3 Description:

The tcsc model is shown in Figure 44.

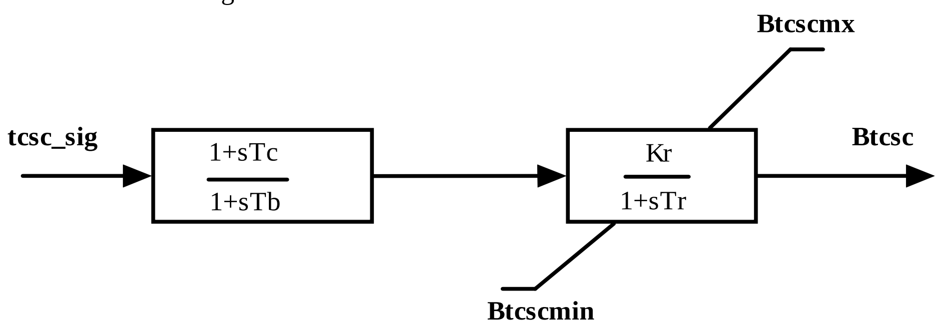


Figure 44 TCSC Model

tcsc_sig is the tcsc input, and Btcsc is the effective susceptance of the TCSC, The susceptance is non windup limited to Btcscmax and Btcscmin.

3.44.4 Inputs:

i the tcsc number
k the integer time step of the solution routine
bus the bus matrix
flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1.
- The network interface calculation is performed when **flag** = 1
- The rates of change of the tcsc states are calculated when **flag** = 2, using the value of **Btcsc** at **k.**

3.44.5 Global Variables

`tcsc_con, n_tcsc, B_tcsc, dB_tcsc, tcsc_sig, tcsc_dsig`

3.44.6 Data Format

The TCSC data is contained in the **i**th row of the matrix **tcsc_con**. The data format for **tcsc_con** is given in Table 17.

Table 17 Data format for TCSC

column	variable	unit
1	tcsc number	
2	From bus number	
3	To bus number	
4	maximum susceptance $B_{tcscmax}$	pu
5	minimum susceptance $B_{tcscmin}$	pu
6	regulator gain K_R	pu
7	regulator time constant T_R	sec
8	compensator lag time constant T_B	sec
9	compensator lead time constant T_B	sec

3.45 tg

3.45.1 Purpose:

Simplified turbine-governor system model

3.45.2 Synopsis:

`tg(i,k,bus,flag)`

3.45.3 Description:

tg(i,k,bus,flag) models the simplified turbine-governor system model shown in Figure 45.

3.45.4 Inputs:

i the number of turbine governor
if **i** = 0 all turbine governor computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the mechanical torque of the synchronous machine is set to the turbine output torque.
- The rates of change of the turbine governor states are calculated when **flag** = 2, using the generator speed deviation at the time specified by **k.**

3.45.5 Global Variables

3.45.5.1 System variables

basmva system base MVA

3.45.5.2 Synchronous Generator Variables

mac_spd ω machine speed in pu
pmech P_m mechanical input power in pu on generator base
pelect P_e electrical active output power in pu on system base
mac_int array to store internal machine ordering

3.45.5.3 Turbine-governor Variables

tg1	governor state variable
tg2	servo state variable
tg3	reheater state variable
dtg1	
dtg2	
dtg3	
tg_con	matrix of turbine governor specifications set by user
tg_pot	internally set matrix of turbine governor constants
n_tg	number of turbine governors
tg_idx	index of turbine governors

3.45.6 Data Format

The data format for the specification file **tg_con** is shown in Table 17.

Table 18 Data format for tg

column	variable	unit
1	turbine model number (=1)	
2	machine number	
3	speed set point ω_f	pu
4	steady state gain *1/r*	pu
5	maximum power order T_{max}	pu on generator base
6	servo time constant T_s	sec
7	HP turbine time constant T_c	sec
8	transient gain time constant T_3	sec
9	time constant to set HP ratio T_4	sec
10	reheater time constant T_5	sec

No time constant is allowed to be zero in this model. The function **tg** is used to model a steam turbine and governor.

3.45.7 Algorithm:

Based on the turbine-governor system model block diagram

- the initialization uses mechanical torque from the synchronous machine to compute the state variables on the integrators. If speed set point is not equal to 1 pu, the power order will be adjusted to give a torque output of the turbine which achieves steady state
- the network interface calculates the output mechanical torque for use by the corresponding generator
- the dynamics calculation determines the rates of change of the turbine governor state variables

This algorithm is implemented in **tg** in the POWER SYSTEM TOOLBOX.

See also: pst_var, mac_em, mac_tra, mac_sub

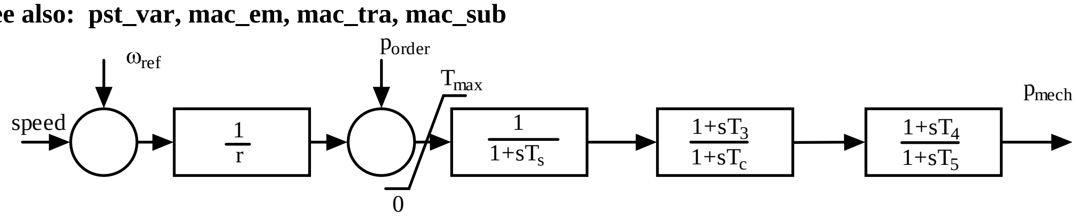


Figure 45 Simple Turbine Governor Model

3.46 tg_hydro

3.46.1 Purpose:

Simplified hydroturbine-governor system model

3.46.2 Synopsis:

`tg_hydro(i,k,bus,flag)`

3.46.3 Description:

tg_hydro(i,k,bus,flag) models the hydroturbine-governor system model shown in Figure 46.

3.46.4 Inputs:

i the number of turbine governor
if **i** = 0 all turbine governor computations are performed using MATLAB vector methods. **This is the preferred mode.**

k the integer time step in a simulation
In small signal simulation, only two values of **k** are used. At **k** = 1, the state variables and there rates of change are set to the initial values. At **k** = 2, the state variables are perturbed in turn and the rates of change of states correspond to those cause by the perturbation.

bus the solved bus specification matrix

flag indicates the mode of solution

- Initialization is performed when **flag** = 0 and **k** = 1. For proper initialization, the corresponding generators must be initialized first.
- The network interface calculation is performed when **flag** = 1, and the mechanical torque of the synchronous machine is set to the turbine output torque.
- The rates of change of the turbine governor states are calculated when **flag** = 2, using the generator speed deviation at the time specified by **k**.

3.46.5 Global Variables

3.46.5.1 Synchronous Generator Variables

mac_spd	ω	machine speed in pu
pmech	P_m	mechanical input power in pu on generator base
mac_int		array to store internal machine ordering

3.46.5.2 Turbine-governor Variables

tg1
tg2
tg3
tg4
tg5
dtg1
dtg2
dtg3
dtg4
dtg5

tg_con	matrix of turbine governor specifications set by user
tg_pot	internally set matrix of turbine governor constants
n_tgh	number of turbine governors
tgh_idx	index of hydro turbine governors

3.46.5.3 Synchronous Generator Variables

mac_spd	ω	machine speed in pu
pmech	P_m	mechanical input power in pu on generator base
pelect	P_e	electrical active output power in pu on system base
mac_int		array to store internal machine ordering

3.46.6 Data Format

The data format for the specification file **tg_con** is shown in Table 18.

Table 19 Data format for tg_hydro

column	variable	unit
1	turbine model number (=2)	
2	machine number	
3	speed set point ω_f	pu
4	permanent droop R_p	pu
5	transient droop R_t	pu on generator base
6	maximum power order *Tmax*	pu on generator base
7	maximum rate limit	pu on generator base
8	Minimum rate limit	pu on generator base
9	servo time constant T_s	sec
10	servo gain K_s	
11	governor time constant T_g	sec
12	reset time constant T_r	sec
13	water starting time T_w	sec

No time constant is allowed to be zero in this model. The function **tg_hydro** is used to model an hydraulic turbine and governor.

3.46.7 Algorithm:

Based on the turbine-governor system model block diagram

- the initialization uses mechanical torque from the synchronous machine to compute the state variables on the integrators. If speed set point is not equal to 1 pu, the power order will be adjusted to give a torque output of the turbine which achieves steady state
- the network interface calculates the output mechanical torque for use by the corresponding generator
- the dynamics calculation determines the rates of change of the turbine governor state variables

This algorithm is implemented in **tg** in the POWER SYSTEM TOOLBOX.

See also: pst_var, mac_em, mac_tra, mac_sub

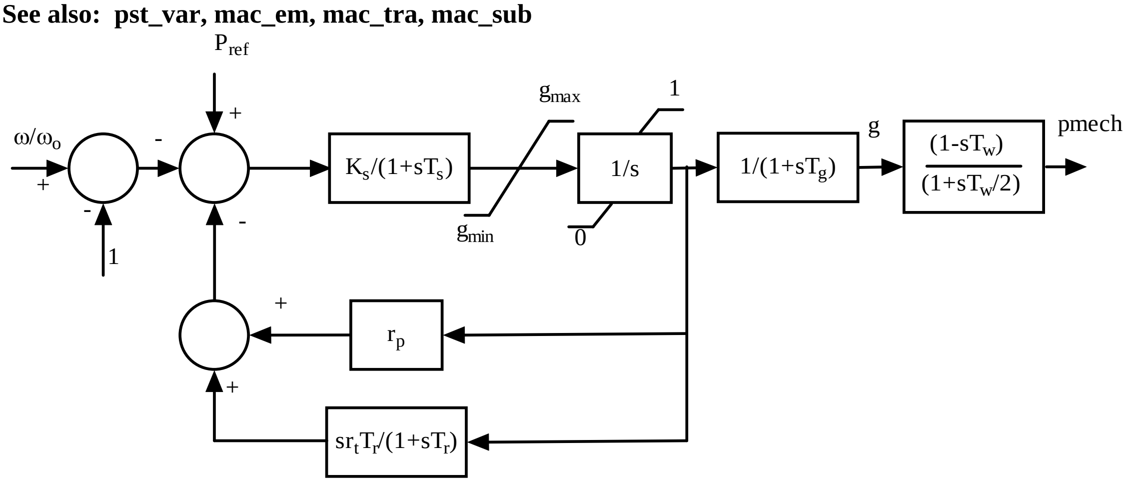


Figure 46 Hydro Turbine Governor Model

3.47 tg_indx

3.47.1 Purpose:

Determines indexes for the turbine generators

3.47.2 Syntax:

tg_indx

3.47.3 Global Variables:

3.47.3.1 Turbine-governor Variables

tg_con	matrix of turbine governor specifications set by user
n_tg	number of thermal turbine governors
n_tgh	numer of hydro turbine governors
tg_idx	index of thermal turbine governors
tgh_idx	index of hydro turbine governors

3.47.4 Algorithm:

Determines the number of turbine governor models and sets the turbine governor index.
This algorithm is implemented in the M-file **tg_indx** in the POWER SYSTEM TOOLBOX.

3.48 y_switch

3.48.1 Purpose:

Forms reduced admittance matrices to correspond with the switching conditions specified in **sw_con**.

3.48.2 Syntax:

`y_switch`

3.48.3 Description:

y_switch is a MATLAB script file which is called from **s_simu**. It is uses the switching data contained in **sw_con** to define the reduced admittance matrices required for transient simulation, i.e., for pre-fault, fault, immediate post-fault, final fault clear.

3.48.4 Data Format

The switching is specified in sw_con which has the format shown in Table 19.

Table 20 Switching file format

time of fault(s)	fault bus number	far bus number	zero sequence fault impedance (pu)	negative sequence fault impedance (pu)	type of fault	time step for fault period(s)
start time	0	0	0	0	0	initial time step
fault on time	fault b#	far b#	zs pu	zn pu	0 – 3 phase 1 - line to ground 2 - line-to-line-ground 3 - line-to-line 4 - loss of line no fault 5 - loss of load 6 - no fault	fault-on time step
initial fault clearing time	0	0	0	0	0	time step
final fault clearing time	0	0	0	0	0	time step
time to change time step						time step
end time						

There may be any number of entries changing the time step following final fault clearing. This allows the use of longer simulation time steps after any initial fast transients have decayed, so allowing faster computation time. The no fault option is useful when the effect of modulation of control signals is to be studied.

3.48.5 Example

The switching data file for the two-area system in **d2asb.m** is

```
sw_con = [...
0    0    0    0    0    0    0.01;%sets intitial time step
0.1  3    101  0    0    0    0.01; % fault at bus 3
0.15 0    0    0    0    0    0.01; %clear near end
0.20 0    0    0    0    0    0.01; %clear remote end
0.50 0    0    0    0    0    0.01; % increase time step
1.0  0    0    0    0    0    0.01; % increase time step
5.0  0    0    0    0    0    0]; % end simulation
```

Note: It is always worth while applying the fault at some short time after the start of the simulation. This allows a check on the unfaulted system which should remain in its initial state. If the initial states drift considerably, the initial rates of change of the states should be checked. These should all be zero, or very close to zero. Non-zero initial rates of change indicate the source of any problem.