IEEE PES Task Force on Benchmark Systems for Stability Controls
Report on the 68-Bus, 16-Machine, 5-Area System

Version 3.3- 3rd Dec, 2013

Abhinav Kumar Singh; and Bikash C. Pal

The present report refers to a small-signal stability study carried over the 68-Bus, 16-Machine, 5-Area System and validated on a widely known software package: MATLAB-Simulink (ver. 2012b). The 68-bus system is a reduced order equivalent of the inter-connected New England test system (NETS) and New York power system (NYPS), with five geographical regions out of which NETS and NYPS are represented by a group of generators whereas, the power import from each of the three other neighboring areas are approximated by equivalent generator models, as shown in Figure 1. This report has the objective to show how the simulation of this system must be done using MATLAB in order to get results that are comparable (and exhibit a good match with respect to the electromechanical modes) with the ones obtained using other commercial software packages and presented on the PES Task Force website on Benchmark Systems for Stability Controls [1]. To facilitate the comprehension, this report is divided into the following sections and in each section the simulation methods are explained, and the results are shown:

1. Load flow and calculation of initial conditions.
2. Small-signal stability assessment using eigenvalues.
3. Non-linear simulation of the system model using Simulink

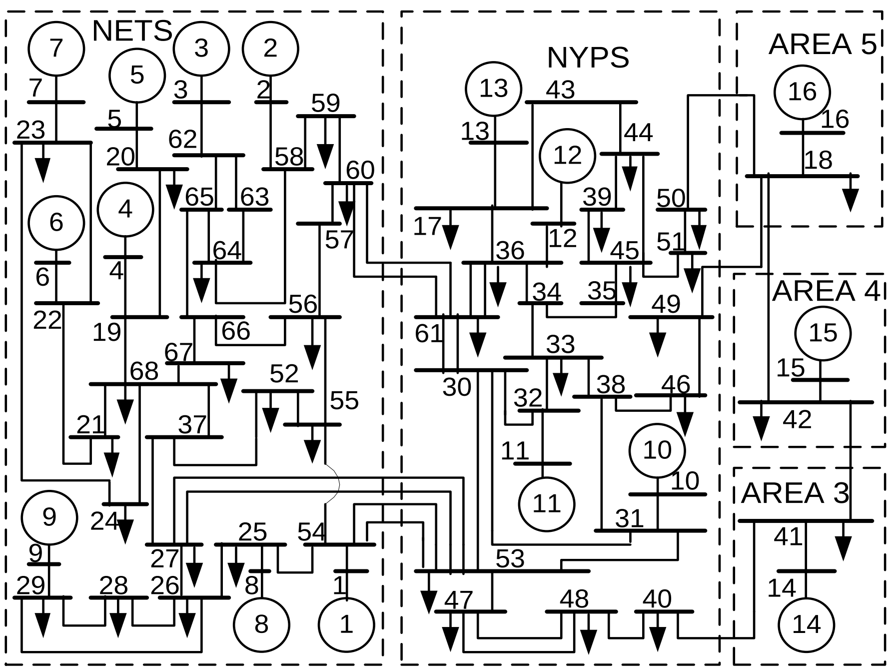

Figure 1- Line Diagram of the 68-bus system

I. LOAD FLOW AND CALCULATION OF INITIAL CONDITIONS

A. MATLAB-Simulink Input files

The bus data, the line data and the machine parameters for the system are given in data16m_benchmark.m. The Init_MultiMachine.m file is used for the calculation of initial conditions, load flow and eigenvalues. This file needs the following files to be present in the same folder as itself in order to run successfully:

1. calc.m
2. chq_lim.m
3. data16m_benchmark.m
4. form_jac.m
5. loadflow.m
6. y_sparse.
7. Benchmark_IEEE_standard.mdl

All of the above files can be downloaded from the website. The codes for all the above '.m' extension files are also given in the appendices. The Init_MultiMachine.m file can either be run by opening it and pressing the green 'Run' button on the editor window, or by typing Init_MultiMachine in MATLAB Command Window. After running, all the results can be seen in the Workspace window of MATLAB. The base value for power is 100 MVA and for frequency it is 60 Hz.

B. Results

The results for load flow can be seen in bus_sol and line_flow variables, where:

- column 1 of bus_sol represents the bus numbers,
- column 2 of bus_sol shows the voltage magnitudes in p.u.,
- column 3 of bus_sol shows the voltage angles in degrees,
- column 4 of bus_sol shows the active power generation in p.u.
- column 5 of bus_sol shows the reactive power generation in p.u.
- column 6 of bus_sol shows the active load in p.u.
- column 7 of bus_sol shows the reactive load generation in p.u.
- column 10 of bus_sol shows the bus type (1 for slack, 2 for PV, 3 for PQ)
- column 2 of line_flow shows "from bus"
- column 3 of line_flow shows "to bus"
- column 4 of line_flow shows the active power flow in p.u.
- column 5 of line_flow shows the reactive power flow in p.u.

The bus results are tabulated as follows:

Bus number	Voltage magnitude (p.u.)	Voltage angle (de-gree.)	Active generation (p.u.)	Reactive generation (p.u.)	Active load (p.u.)	Reactive load (p.u.)	Bus type
1	1.045	-8.9563	2.5	1.96	0	0	2
2	0.98	-0.9835	5.45	0.7001	0	0	2
3	0.983	1.6129	6.5	0.8078	0	0	2
4	0.997	1.6683	6.32	0.0027	0	0	2
5	1.011	-0.6276	5.05	1.1657	0	0	2

6	1.05	3.8425	7	2.5449	0	0	2
7	1.063	6.0307	5.6	2.9083	0	0	2
8	1.03	-2.841	5.4	0.4907	0	0	2
9	1.025	2.6524	8	0.5981	0	0	2
10	1.01	-9.6439	5	-0.131	0	0	2
11	1	-7.2245	10	0.0832	0	0	2
12	1.0156	-22.6313	13.5	2.8006	0	0	2
13	1.011	-28.6539	35.91	8.8504	0	0	2
14	1	10.962	17.85	0.4748	0	0	2
15	1	0.0168	10	0.7673	0	0	2
16	1	0	33.7953	0.9364	0	0	1
17	0.9499	-36.0269	0	0	60	3	3
18	1.0023	-5.8054	0	0	24.7	1.23	3
19	0.932	-4.2634	0	0	0	0	3
20	0.9806	-5.8744	0	0	6.8	1.03	3
21	0.9602	-7.0539	0	0	2.74	1.15	3
22	0.9937	-1.801	0	0	0	0	3
23	0.9961	-2.1606	0	0	2.48	0.85	3
24	0.9587	-9.8767	0	0	3.09	-0.92	3
25	0.9981	-9.9995	0	0	2.24	0.47	3
26	0.9869	-11.0194	0	0	1.39	0.17	3
27	0.9679	-12.8571	0	0	2.81	0.76	3
28	0.9897	-7.4968	0	0	2.06	0.28	3
29	0.9921	-4.5464	0	0	2.84	0.27	3
30	0.9762	-19.71	0	0	0	0	3
31	0.9838	-17.464	0	0	0	0	3
32	0.9699	-15.2375	0	0	0	0	3
33	0.9738	-19.758	0	0	1.12	0	3
34	0.98	-26.1159	0	0	0	0	3
35	1.043	-27.0886	0	0	0	0	3
36	0.9606	-28.8273	0	0	1.02	-0.1946	3
37	0.9555	-11.788	0	0	0	0	3
38	0.989	-18.7593	0	0	0	0	3
39	0.9915	-39.2902	0	0	2.67	0.126	3
40	1.0442	-13.64	0	0	0.6563	0.2353	3
41	0.9996	9.4272	0	0	10	2.5	3
42	0.999	-0.8435	0	0	11.5	2.5	3
43	0.9765	-37.9088	0	0	0	0	3
44	0.9775	-37.9863	0	0	2.6755	0.0484	3
45	1.0471	-29.3626	0	0	2.08	0.21	3
46	0.9903	-20.5314	0	0	1.507	0.285	3
47	1.0184	-19.4976	0	0	2.0312	0.3259	3
48	1.0337	-18.3761	0	0	2.412	0.022	3
49	0.9936	-19.8196	0	0	1.64	0.29	3

50	1.0602	-19.0521	0	0	1	-1.47	3
51	1.0634	-27.2882	0	0	3.37	-1.22	3
52	0.9545	-12.8334	0	0	1.58	0.3	3
53	0.9863	-18.9373	0	0	2.527	1.1856	3
54	0.9857	-11.537	0	0	0	0	3
55	0.9571	-13.2179	0	0	3.22	0.02	3
56	0.9208	-11.9593	0	0	2	0.736	3
57	0.9102	-11.2129	0	0	0	0	3
58	0.909	-10.4023	0	0	0	0	3
59	0.9037	-13.311	0	0	2.34	0.84	3
60	0.9062	-14.0368	0	0	2.088	0.708	3
61	0.9556	-23.2222	0	0	1.04	1.25	3
62	0.9121	-7.3117	0	0	0	0	3
63	0.9096	-8.37	0	0	0	0	3
64	0.8367	-8.377	0	0	0.09	0.88	3
65	0.9128	-8.1847	0	0	0	0	3
66	0.9194	-10.1909	0	0	0	0	3
67	0.928	-11.4299	0	0	3.2	1.53	3
68	0.9483	-10.0712	0	0	3.29	0.32	3

Table 1: Bus results from loadflow using MATLAB

II. SMALL-SIGNAL STABILITY ASSESSMENT USING EIGENVALUES

A. Dynamic model of a Power System

A modern power system consists of various components such as the generators, their excitation systems, power system stabilizers (PSS), FACTS devices such as TCSC, loads, transmission network etc. An accurate modeling of each of these components is fundamental to the study of the dynamic behavior of a power system. As described earlier, the dynamic behavior of the components is modeled using a set of DAEs. A brief description of each component and its governing equations are presented as follows.

1. Generators

All the generators of the power system are represented using the sub-transient models with four equivalent rotor coils as per the IEEE convention. The slow-dynamics of the governors are ignored, and the mechanical torques to the generators are taken as constant inputs. Standard notations ae followed in the following differential equations which represent the dynamic behavior of the i^{th}generator [2]:

$$\frac{d\delta_i}{dt} = \omega_B(\omega_i - \omega_s) = \omega_B S_{mi} \tag{1}$$

$$2H_i\frac{dS_{mi}}{dt} = (T_{mi} - T_{ei}) - D_i S_{mi} \tag{2}$$

$$where, T_{ei} = E'_{di} I_{di} \frac{(X''_{qi} - X_{lsi})}{(X'_{qi} - X_{lsi})} + E'_{qi} I_{qi} \frac{(X''_{di} - X_{lsi})}{(X'_{di} - X_{lsi})} - I_{di} I_{qi} (X''_{di} - X''_{qi})$$
$$+ \psi_{1di} I_{qi} \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{lsi})} - \psi_{2qi} I_{di} \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{lsi})};$$
$$and, I_{qi} + jI_{di} = 1/(R_{ai} + jX''_{di}) \left\{ E'_{qi} \frac{(X''_{di} - X_{lsi})}{(X'_{di} - X_{lsi})} + \psi_{1di} \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{lsi})} - V_{qi} \right.$$
$$\left. + j \left[E'_{di} \frac{(X''_{qi} - X_{lsi})}{(X'_{qi} - X_{lsi})} - \psi_{2qi} \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{lsi})} - V_{di} + E'_{dci} \right] \right\};$$
$$and, T_{ci} \frac{dE'_{dci}}{dt} = I_{qi}(X''_{di} - X''_{qi}) - E'_{dci}$$

$$T'_{q0i} \frac{dE'_{di}}{dt} = -E'_{di} + (X_{qi} - X'_{qi}) \left\{ -I_{qi} + \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{lsi})^2} \left((X'_{qi} - X_{lsi}) I_{qi} - E'_{di} - \psi_{2qi} \right) \right\} \quad (3)$$

$$T'_{d0i} \frac{dE'_{qi}}{dt} = E_{fdi} - E'_{qi} + (X_{di} - X'_{di}) \left\{ I_{di} + \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{lsi})^2} \left(\psi_{1di} - (X'_{di} - X_{lsi}) I_{di} - E'_{qi} \right) \right\} \quad (4)$$

$$T''_{d0i} \frac{d\psi_{1di}}{dt} = E'_{qi} + (X'_{di} - X_{lsi}) I_{di} - \psi_{1di} \quad (5)$$

$$T''_{q0i} \frac{d\psi_{2qi}}{dt} = -E'_{di} + (X'_{qi} - X_{lsi}) I_{qi} - \psi_{2qi} \quad (6)$$

Here, the subscript i refers to the i^{th} generator; δ is the rotor angle in radians; ω_B is the rotor base angular speed in radians per second; H is the inertia constant in seconds; T'_{d0} and T''_{d0} are the d-axis open circuit transient and sub-transient time constants, respectively; T'_{q0} and T''_{q0} are the q-axis open circuit transient and sub-transient time constants, respectively; and T_c is the time constant for the dummy rotor coil (which is usually taken as 0.01 seconds). Rest of the variables are in per unit (p.u.): ω is rotor angular velocity; ω_s is the synchronous angular velocity; S_{mi} is the slip; T_m is the mechanical torque; T_e is the electrical torque; D is the machine rotor damping; E'_q is the transient emf due to field flux linkages; E'_d is the transient emf due to flux linkage in q-axis damper coil; ψ_{1d} and ψ_{2q} are the sub-transient emfs due to d-axis and q-axis damper coils, respectively; E_{fd} is the field excitation voltage; E'_{dc} is the transient emf across the dummy rotor coil; I_d and I_q are the d-axis and q-axis components of the stator current, respectively; V_d and V_q are the d-axis and q-axis components of the stator terminal voltage, respectively; X_d, X'_d and X''_d are the synchronous, transient and sub-transient reactances, respectively, along the d-axis; X_q, X'_q and X''_q are the synchronous, transient and sub-transient reactances, respectively, along the q-axis; R_a is the armature resistance and X_{ls} is the armature leakage reactance.

2. Excitation Systems (AVRs)

We use two types of automatic voltage regulators (AVRs) for the excitation of the generators. The first type is an IEEE standard DC exciter (DC4B) and the second type is the standard static exciter (ST1A). The differential equations governing the operation of the IEEE-DC4B excitation system are given by (7), while for the IEEE-ST1A are given by (8):

$$T_e \frac{dE_{fd}}{dt} = V_a - \left(K_e E_{fd} + E_{fd} A_{ex} e^{B_{ex} E_{fd}}\right) \tag{7}$$

$$where, Efdmin \le V_a \le Efdmax;\ T_r \frac{dV_r}{dt} = V_t - V_r;\ T_f \frac{dV_f}{dt} = E_{fd} - V_f;$$

$$T_a \frac{dV_a}{dt} = K_a V_{PID} - V_a;\ Efdmin/K_a \le V_{PID} \le Efdmax/K_a;$$

$$V_{PID} = \left(V_{ref} + V_{ss} - V_r - \frac{K_f}{T_f}\left[E_{fd} - V_f\right]\right)\left[K_p + \frac{K_i}{s} + \frac{sK_d}{sT_d + 1}\right]$$

$$E_{fd} = K_A\left(V_{ref} + V_{ss} - V_r\right);\ Efdmin \le E_{fd} \le Efdmax;\ T_r \frac{dV_r}{dt} = V_t - V_r \tag{8}$$

Here, E_{fd} is the field excitation voltage, K_e is the exciter gain, T_e is the exciter time constant, T_r is the input filter time constant, V_r is the input filter emf, K_f is the stabilizer gain, T_f is the stabilizer time constant, V_f is the stabilizer emf, A_{ex} and B_{ex} are the saturation constants, K_a is the dc regulator gain, T_a is the regulator time constant, V_a is the regulator emf, K_p, K_i, K_d and T_d are the PID-controller parameters, K_A is the static regulator gain, $V_{ref}$is the reference voltage and V_{ss} is output reference voltage from the PSS.

3. Power System Stabilizers (PSSs)

Besides the excitation control of generators, we use PSSs as supplements to damp the local modes. The feedback signal to a PSS may be the rotor speed (or slip), terminal voltage of the generator, real power or reactive power generated etc. and that signal is chosen which has maximum controllability and observability in the local mode. If the rotor slip S_m is used as the feedback signal then the dynamic equation of the PSS is given by (9).

$$V_{ss} = K_{pss} \frac{sT_w}{(1 + sT_w)} \frac{(1 + sT_{11})}{(1 + sT_{12})} \frac{(1 + sT_{21})}{(1 + sT_{22})} \frac{(1 + sT_{31})}{(1 + sT_{32})} S_m \tag{9}$$

Here K_{pss} is the PSS gain, T_w is the washout time constant, T_{i1} and T_{i2} are the i^{th} stage lead and lag time constants, respectively.

4. Loads, Network Interface and Network Equations

While writing the algebraic network balance equations, we need to work in a common reference frame of the network, instead of the rotating reference frame of the generators. Therefore the generator currents and voltages need to be rotated by the rotor phase angle δ, the resulting equations are given by (10).

$$I_{Qi} + jI_{Di} = \left(I_{qi} + jI_{di}\right)e^{j\delta_i}; \text{and } V_{gi} = V_{Qi} + jV_{Di} = \left(V_{qi} + jV_{di}\right)e^{j\delta_i} \tag{10}$$

In order to club the generator admittances with the network admittance, we represent the generator as a current injection source, given by $I_{gi} = \left(I_{Qi} + jI_{Di}\right) + Y_{gi}V_{gi}$, where $Y_{gi} = 1/(R_{ai} + jX''_{di})$. The generator admittance matrix $\boldsymbol{Y}_G$ is given by $diag(Y_{G1}, Y_{G2}, \dots, Y_{GN})$, where N is the total number of bus nodes, and $Y_{Gj} = Y_{gi}$ if j^{th} node is conected to the i^{th} generator, and $Y_{Gj} = 0$ otherwise. Similarly, the load admittance matrix $\boldsymbol{Y}_L$ is defined, where loads are taken as constant shunt impedances. Assuming that the current injections take place only at the generator buses, the current injection column vector $\boldsymbol{I}$ is such that its j^{th} element $I^j = I_{gi}$ if the j^{th} node is connected to the i^{th} generator, and $I^j = 0$ otherwise, for $j = 1$ to N. The network shunt admittance matrix $\boldsymbol{Y}_N$ is formed using the line impedances, and augmented with $\boldsymbol{Y}_G$ and $\boldsymbol{Y}_L$ to give $\boldsymbol{Y}_{Aug} = \boldsymbol{Y}_N + \boldsymbol{Y}_G + \boldsymbol{Y}_L$, and the column vector of bus voltages $\mathbf{V}$ is given by by (11).

$$\mathbf{V} = \mathbf{Z}_{Aug}\boldsymbol{I}, \text{ where } \mathbf{Z}_{Aug} = \left(\mathbf{Y}_{Aug}\right)^{-1} \tag{11}$$

B. Description of the dynamic elements in the 68-bus system

The 68-bus system is a reduced order equivalent of the interconnected New England test system (NETS) (containing G1 to G9) and New York power system (NYPS) (containing G10 to G13), with five geographical regions out of which NETS and NYPS are represented by a group of generators whereas, the power import from each of the three other neighboring areas are approximated by equivalent generator models (G14 to G16). G13 also represents a small sub-area within NYPS.

There are three major tie-lines between NETS and NYPS (connecting buses 60-61, 53-54 and 27-53). All the three are double-circuit tie-lines. Generators G1 to G8, and G10 to G12 have DC excitation systems (DC4B); G9 has fast static excitation (ST1A), while the rest of the generators (G13 to G16) have manual excitation as they are area equivalents instead of being physical generators.

C. Linearization and calculation of eigen-values

The DAEs (differential and algebraic equations) explained in Section II.A are implemented in the simulink model 'Benchmark_IEEE_standard.mdl'. The 'Init_MultiMachine.m' not only runs load flow, finds the initial steady state values, but it also linearizes the system at t=0, thereby finding the state space matrices and eigenvalues for the linearized system. The eigenvalues are stored in variable 'lambda' in Workspace. They have been calculated for three cases; 1) without any PSS, 2) with PSS only on machine G9; and 3) with PSSs on machines G1 to G12:

1) Without any PSS

The PSSs can be removed from the model by editing the original 'data16m_benchmark.m' file and rewriting the PSSs' parameters as '`pss_con = [];'` Figure 2 shows the plot of the eigenvalues for the case when there are not any PSS in the system. The system is unstable as three pairs of eigenvalues have positive real parts, and also a lot of eigenvalues are outside the 10% damping line.

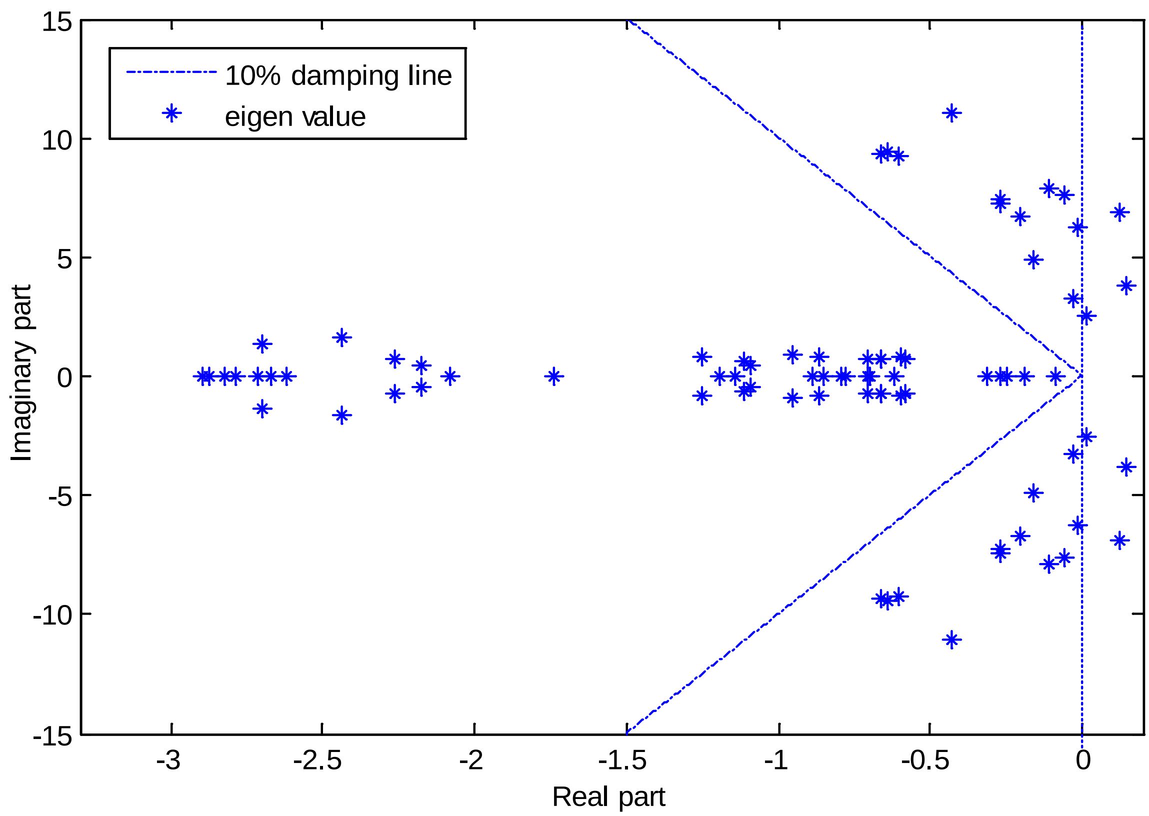

Figure 2- Eigen-value plot for the system without any PSS

2) With PSS only on machine G9

For this case, the PSSs' parameters in the original 'data16m_benchmark.m' file are edited so that only the line corresponding to machine G9 remains:

```
 pss_con = [
           9   9   12  10  0.09  0.02  0.09  0.02  1     1    0.2 -0.05 ;
           ];
```

An easy way of doing this is by commenting out all the lines in 'pss_con', except the line corresponding to machine G9. Fig. 3 shows the plot of the eigenvalues for this case with PSS only on G9. The system is unstable as one pair of eigenvalues still has positive real parts, and also a lot of eigenvalues are outside the 10% damping line. The role of PSS in damping the local modes of G9 will be analyzed in the next section.

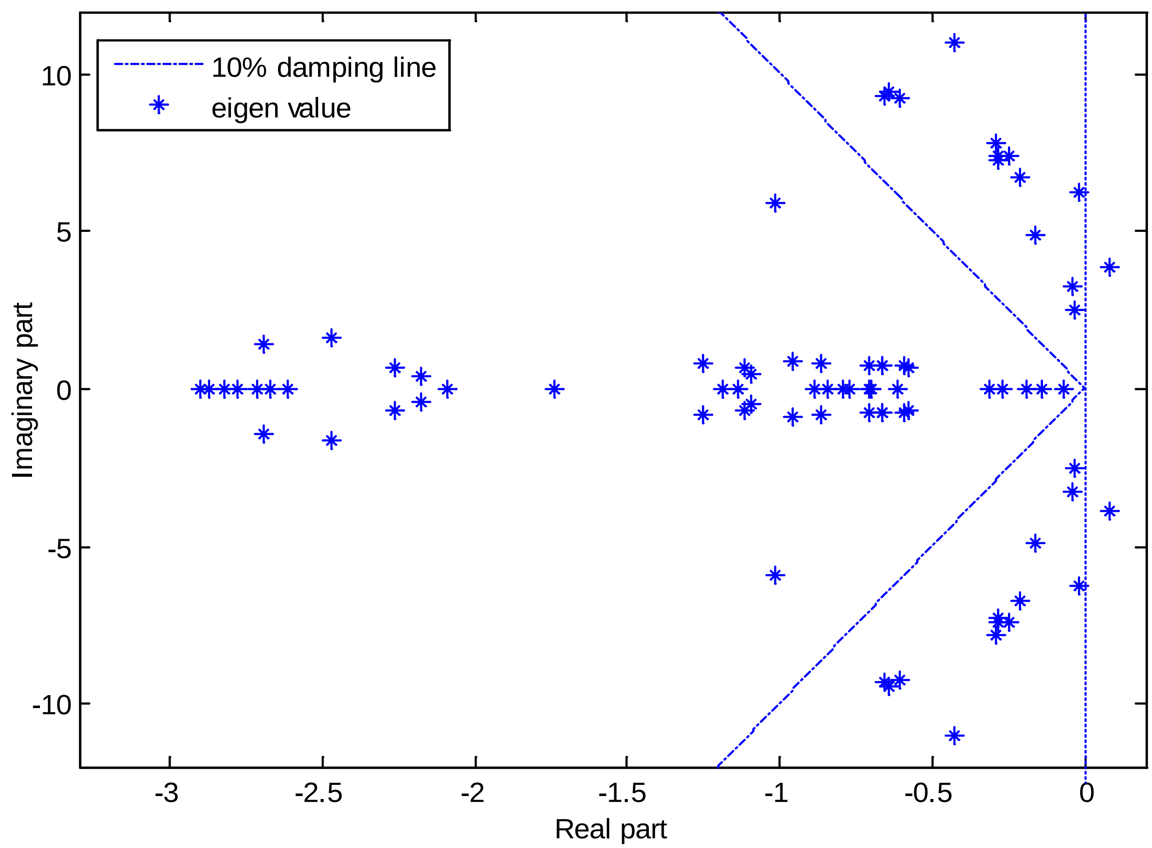

Figure 3- Eigen-value plot for the system with PSS only on machine G9

3) With PSSs on machines G1 to G12

Fig. 4 shows the plot of the eigenvalues for the case with PSSs installed on machines G1 to G12. The system is stable, and only three pairs of eigenvalues are outside the 10% damping line. Thus the PSSs successfully stabilize the system, and damp all of the local modes. The three poorly-damped modes are inter-area modes.

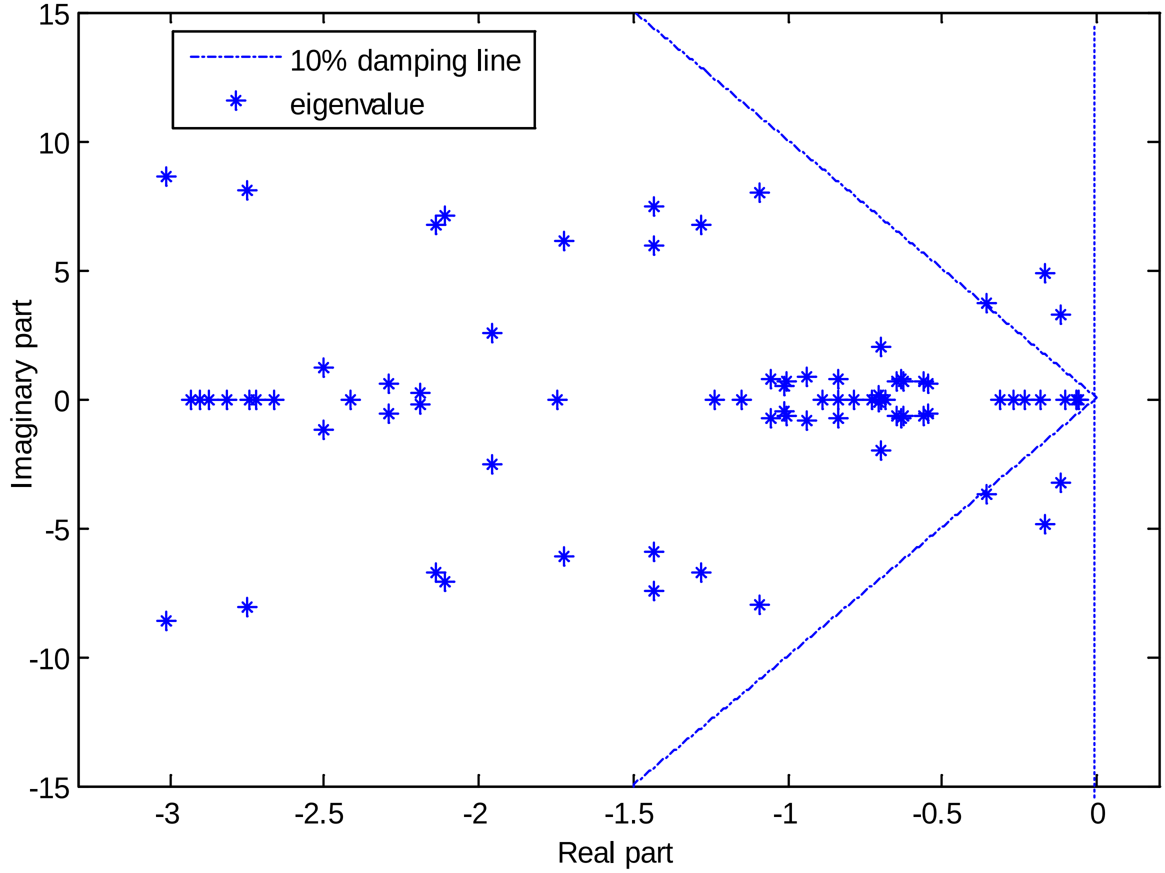

Figure 4- Eigen-value plot for the system with PSSs on machines G1 to G12

D. Electro-mechanical modes

It was observed that all the poorly damped (damping ratio less than 10%) or unstable modes had high participation from rotor angles, and rotor slips of various machines. Such modes are called as electromechanical modes. The electromechanical modes with frequencies in the range 0.1 to 1 Hz are the inter-area modes, while the rest of them are local machine modes. Table 2 - Table 4 show the electromechanical modes for the aforementioned three cases. In each case, the top four modes are inter-area modes while the rest are local modes. A detailed modal analysis of these modes is required to find which machines have high participations in them. The variables ‘st’ and ‘pfac’, which are obtained in the Matlab Workspace after running ‘Init_MultiMachine.m’, give the top ten states and their corresponding normalized participation factors in all the eigen-values of the system. In Table 2 - Table 4, for each mode the highest participating states are also tabulated, and arranged in the increasing order of the normalized participation factors.

1) Without any PSS

Without any PSS in the system, all of the electromechanical modes are either unstable, or poorly damped, as can be seen in Table 2. All the inter-area modes have high participation from machines G13 to G16. The local modes have high participation from the corresponding local machine, for example, the seventh mode in Table 2, with damping ratio -1.803% and frequency 1.093 Hz, has high participation from G9.

No.	Damping Ratio (%)	Frequency (Hz)	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor
1	-0.438	0.404	'Delta(13)'	1	'SM(13)'	0.741	'SM(15)'	0.556	'SM(14)'	0.524
2	0.937	0.526	'Delta(14)'	1	'SM(16)'	0.738	'SM(14)'	0.5	'Delta(13)'	0.114
3	-3.855	0.61	'Delta(13)'	1	'SM(13)'	0.83	'Delta(12)'	0.137	'SM(6)'	0.136
4	3.321	0.779	'Delta(15)'	1	'SM(15)'	0.755	'SM(14)'	0.305	'Delta(14)'	0.149
5	0.256	0.998	'Delta(2)'	1	'SM(2)'	0.992	'Delta(3)'	0.913	'SM(3)'	0.905
6	3.032	1.073	'Delta(12)'	1	'SM(12)'	0.985	'SM(13)'	0.193	'Delta(13)'	0.179
7	-1.803	1.093	'Delta(9)'	1	'SM(9)'	0.996	'Delta(1)'	0.337	'SM(1)'	0.333
8	3.716	1.158	'Delta(5)'	1	'SM(5)'	1	'SM(6)'	0.959	'Delta(6)'	0.958
9	3.588	1.185	'SM(2)'	1	'Delta(2)'	1	'Delta(3)'	0.928	'SM(3)'	0.928
10	0.762	1.217	'Delta(10)'	1	'SM(10)'	0.991	'SM(9)'	0.426	'Delta(9)'	0.42
11	1.347	1.26	'SM(1)'	1	'Delta(1)'	0.996	'Delta(10)'	0.761	'SM(10)'	0.756
12	6.487	1.471	'Delta(8)'	1	'SM(8)'	1	'SM(1)'	0.435	'Delta(1)'	0.435
13	7.033	1.487	'Delta(4)'	1	'SM(4)'	1	'SM(5)'	0.483	'Delta(5)'	0.483
14	6.799	1.503	'Delta(7)'	1	'SM(7)'	1	'SM(6)'	0.557	'Delta(6)'	0.557
15	3.904	1.753	'Delta(11)'	1	'SM(11)'	0.993	'Psi_1d(11)'	0.056	'SM(10)'	0.033

Table 2 Electromechanical modes with normalized participation factors of highest participating states

2) With PSS only on G9

When PSS is present only on G9, the system gets a little more damped, and there is only one unstable mode, as compared to the three unstable modes in the previous case. Comparing the seventh mode in Table 2 and Table 3, the effect of local damping provide by the PSS is quite evident as the unstable mode with -1.8% damping gets damped to 16.9%. This damping can be directly attributed to the added participation of the PSS states in the seventh mode.

No.	Damping Ratio (%)	Frequency (Hz)	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor
1	1.425	0.399	'Delta(13)'	1	'SM(13)'	0.715	'SM(15)'	0.604	'SM(14)'	0.563
2	1.308	0.525	'Delta(14)'	1	'SM(16)'	0.72	'SM(14)'	0.504	'Delta(15)'	0.087
3	-2.151	0.614	'Delta(13)'	1	'SM(13)'	0.826	'Delta(12)'	0.137	'SM(6)'	0.132
4	3.324	0.779	'Delta(15)'	1	'SM(15)'	0.755	'SM(14)'	0.305	'Delta(14)'	0.149
5	0.256	0.998	'Delta(2)'	1	'SM(2)'	0.992	'Delta(3)'	0.914	'SM(3)'	0.905
6	3.186	1.072	'Delta(12)'	1	'SM(12)'	0.985	'SM(13)'	0.197	'Delta(13)'	0.184
7	16.915	0.944	'Delta(9)'	1	'SM(9)'	0.67	'PSS2'	0.221	'PSS3'	0.221
8	3.877	1.155	'SM(5)'	1	'Delta(5)'	0.997	'Delta(6)'	0.761	'SM(6)'	0.758
9	3.788	1.184	'SM(2)'	1	'Delta(2)'	0.994	'Delta(3)'	0.515	'SM(3)'	0.511
10	3.352	1.182	'SM(3)'	1	'Delta(3)'	0.989	'Delta(10)'	0.886	'SM(10)'	0.876
11	3.69	1.249	'Delta(10)'	1	'SM(10)'	0.993	'SM(1)'	0.641	'Delta(1)'	0.636
12	6.561	1.47	'Delta(8)'	1	'SM(8)'	1	'SM(1)'	0.455	'Delta(1)'	0.454
13	7.034	1.487	'Delta(4)'	1	'SM(4)'	1	'SM(5)'	0.483	'Delta(5)'	0.483
14	6.799	1.503	'Delta(7)'	1	'SM(7)'	1	'SM(6)'	0.557	'Delta(6)'	0.557
15	3.908	1.753	'Delta(11)'	1	'SM(11)'	0.993	'Psi_1d(11)'	0.056	'SM(10)'	0.033

Table 3 Electromechanical modes with normalized participation factors for the case with PSS only on G9

3) *With PSS on G1 to G12*

When PSSs are present on machines G1 to G12, the system becomes stable, and much more damped than the previous case; and there are only three poorly damped modes, all of which are inter-area modes. All the local modes get properly damped. The next section analyzes how the inter-area modes may be properly damped.

No.	Damping Ratio (%)	Frequency (Hz)	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor	State	Partici-pation Factor
1	33.537	0.314	'SM(15)'	1	'SM(16)'	0.982	'Delta(7)'	0.884	'Delta(4)'	0.857
2	3.621	0.52	'Delta(14)'	1	'SM(16)'	0.645	'SM(14)'	0.521	'Delta(15)'	0.149
3	9.625	0.591	'Delta(13)'	1	'SM(13)'	0.83	'SM(16)'	0.111	'SM(12)'	0.096
4	3.381	0.779	'Delta(15)'	1	'SM(15)'	0.755	'SM(14)'	0.304	'Delta(14)'	0.148
5	27.136	0.972	'SM(3)'	1	'Delta(3)'	0.962	'SM(2)'	0.924	'Delta(2)'	0.849
6	18.566	1.08	'SM(12)'	1	'Delta(12)'	0.931	'Eqd(12)'	0.335	'PSS4(11)'	0.196
7	23.607	0.939	'Delta(9)'	1	'SM(9)'	0.684	'PSS3(9)'	0.231	'PSS2(9)'	0.231
8	30.111	1.078	'Delta(5)'	1	'SM(5)'	0.928	'SM(6)'	0.909	'Delta(6)'	0.883
9	28.306	1.136	'SM(2)'	1	'Delta(2)'	0.961	'Delta(3)'	0.798	'SM(3)'	0.777
10	13.426	1.278	'SM(1)'	1	'Delta(1)'	0.969	'Eqd(1)'	0.135	'SM(8)'	0.111
11	18.83	1.188	'Delta(10)'	1	'SM(10)'	0.994	'Eqd(10)'	0.324	'PSS3(10)'	0.19
12	32.062	1.292	'Delta(8)'	1	'SM(8)'	0.935	'Eqd(8)'	0.61	'PSS2(8)'	0.422
13	39.542	1.288	'Delta(4)'	1	'SM(4)'	0.888	'Eqd(4)'	0.706	'PSS4(4)'	0.537
14	33.119	1.367	'Delta(7)'	1	'SM(7)'	0.909	'Delta(6)'	0.653	'SM(6)'	0.612
15	23.87	5.075	'SM(11)'	1	'Eqd(11)'	0.916	'Psi_1d(11)'	0.641	'PSS4(10)'	0.464

Table 4 -Electromechanical modes with normalized participation factors for the case with PSS on G1-G12

E. Inter-area modes

The inter-area modes are named so because in these modes the participating machines divide into two groups, and the two groups oscillate against each other. If the inter-area modes are poorly damped, or unstable, then the two groups may lose synchronism completely and this leads to system breakdown. Also, as the inter-area modes have low frequencies as compared to other modes, for a given damping ratio they take much more time to die down than the other modes. A 10% or more damping ratio for all the inter-area modes gives an acceptable system performance, and hence control methods are designed to give at least 10% damping ratio to all the inter-area modes.

The phenomenon of all the machines dividing into two groups may be better understood by the help of mode shapes. Mode shapes are the polar plots of the eigenvectors of a mode corresponding to the desired states. In Matlab, ‘feather’ or ‘compass’ functions may be used for plotting the mode shapes. Figure 5 shows the mode shapes of the inter-area modes, in which the eigenvectors (corresponding to each machine’s slip) of all the inter-area modes are plotted. Figure 5 shows the mode shapes for case when there is no PSS in the system, while Figure 6 shows the mode shapes with PSSs included on G1-G12. The division of machines into two opposing groups is evident in both the cases. It can also be observed that the mode shapes of Modes 2 to 4 (the poorly damped modes) do not change much, and hence PSSs have small effect on these poorly damped modes.

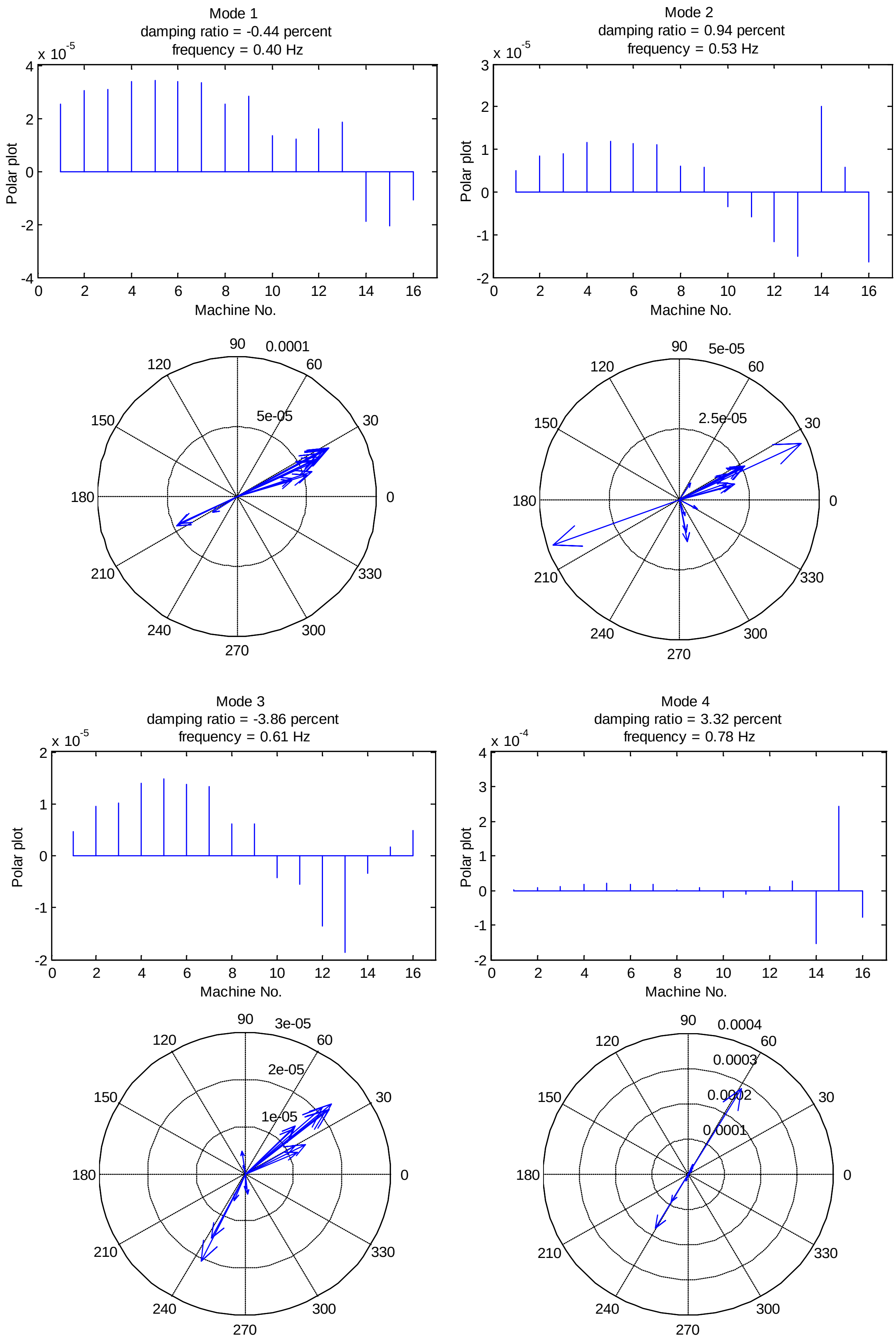

Figure 5 - Mode shapes for inter-area modes, without PSS

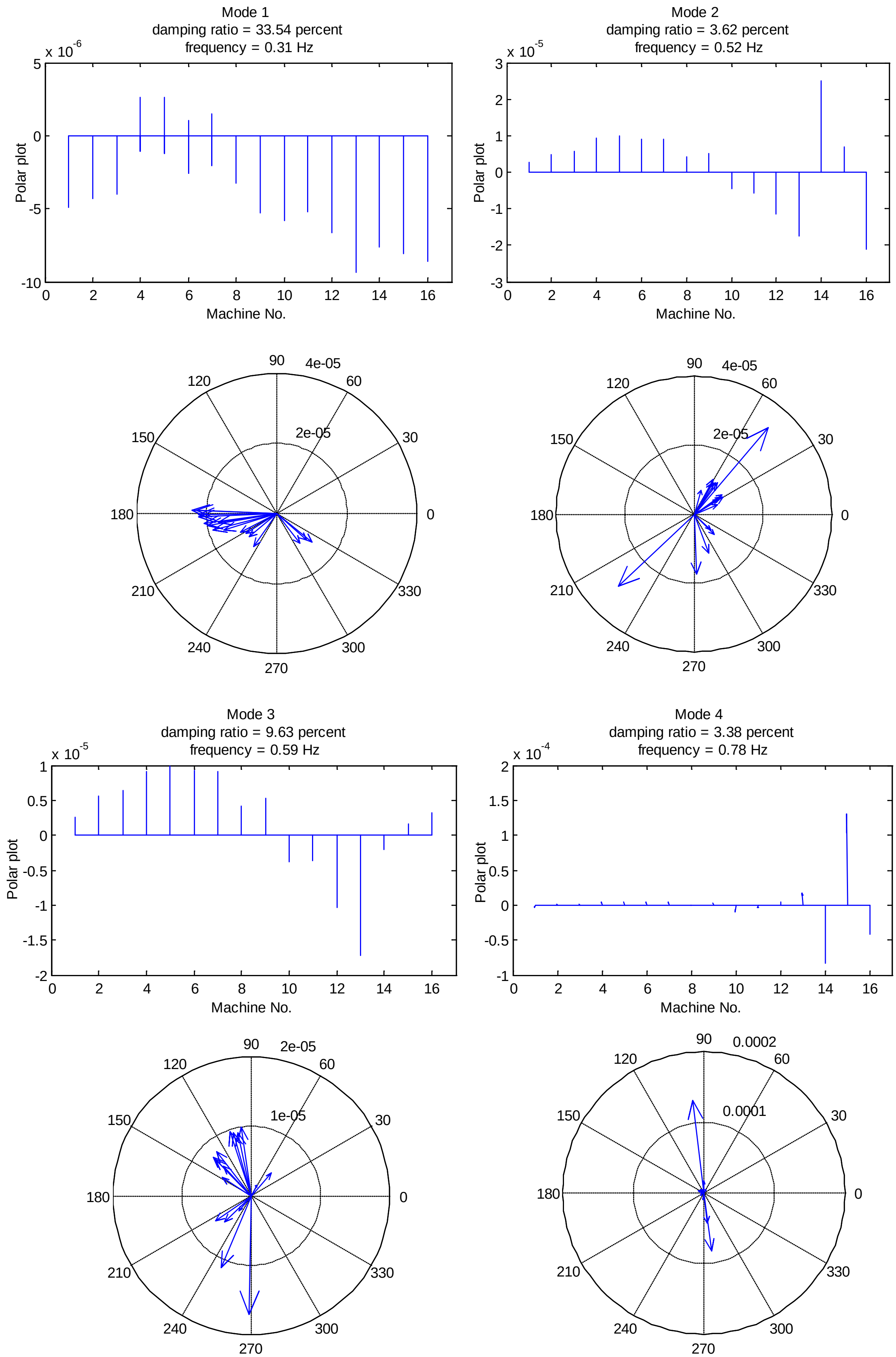

Figure 6 - Mode shapes for inter-area modes, with PSS on G1-G12

A way to assess the controllability of electromechanical modes is participation factor analysis. Table 5 gives a complete list of normalized participation factors (in increasing order) of all the speed-deviations (rotor slips) in all the four inter-area modes. It can be observed that the three poorly damped modes have very little participation from the rotor slips of all the machines with PSSs. Hence these three modes cannot be damped just by PSSs and we need wide area measurement systems (WAMS) and wide area control-device(s), such as Flexible AC transmission system (FACTS) devices, to damp them. This is one of the challenges offered by the system under study, and provides wide scope for research. Study of WAMS and FACTS is outside the scope of this task force, and the interested readers are referred to [2].

Mode 1 damping ratio=33.54% frequency=0.314 Hz		**Mode 2** damping ratio=3.62% frequency=0.520 Hz		**Mode 3** damping ratio=9.63% frequency=0.591 Hz		**Mode 4** damping ratio=3.38% frequency=0.779 Hz	
State	Participation factor	State	Participation factor	State	Participation factor	State	Participation factor
'SM(15)'	1	'SM(16)'	0.83	'SM(13)'	0.6448	'SM(15)'	0.7551
'SM(16)'	0.9817	'SM(14)'	0.1114	'SM(16)'	0.5214	'SM(14)'	0.304
'SM(14)'	0.843	'SM(13)'	0.0959	'SM(12)'	0.1112	'SM(16)'	0.1211
'SM(13)'	0.68	'SM(15)'	0.0366	'SM(6)'	0.0295	'SM(13)'	0.0083
'SM(6)'	0.2602	'SM(12)'	0.0286	'SM(7)'	0.0151	'SM(10)'	0.0004
'SM(5)'	0.2252	'SM(6)'	0.0278	'SM(4)'	0.0059	'SM(9)'	0.0002
'SM(3)'	0.2199	'SM(3)'	0.0277	'SM(3)'	0.005	'SM(12)'	0.0002
'SM(4)'	0.2148	'SM(7)'	0.0271	'SM(14)'	0.0046	'SM(6)'	0.0002
'SM(7)'	0.1995	'SM(4)'	0.0255	'SM(5)'	0.0046	'SM(7)'	0.0001
'SM(9)'	0.191	'SM(9)'	0.0245	'SM(9)'	0.0045	'SM(5)'	0.0001
'SM(2)'	0.17	'SM(5)'	0.0211	'SM(2)'	0.0041	'SM(1)'	0.0001
'SM(1)'	0.1594	'SM(2)'	0.0156	'SM(15)'	0.004	'SM(4)'	0.0001
'SM(12)'	0.1394	'SM(1)'	0.0122	'SM(1)'	0.003	'SM(3)'	0.0001
'SM(8)'	0.0999	'SM(11)'	0.0093	'SM(11)'	0.0022	'SM(2)'	0.0001
'SM(11)'	0.0531	'SM(8)'	0.0085	'SM(10)'	0.0019	'SM(11)'	0.0001
'SM(10)'	0.0506	'SM(10)'	0.0085	'SM(8)'	0.0017	'SM(8)'	0

Table 5 - Normalized participation factors of all the rotor-slips in the four inter-area modes

III. NONLINEAR SIMULATION OF THE MODEL USING SIMULINK

In the nonlinear simulations using Simulink, two cases are considered:

a) A 20 s simulation, with a 2% step in Vref of the test machine at t = 1.0 s and a -2% step in the same Vref at t = 11.0 s;
b) The connection, at t = 1.0 s, of a 50 MVAr shunt reactor to the bus connected to the test machine, removing this reactor at t = 11.0 s and simulating until t = 20 s.

As we are interested in observing the damping provided by the PSS, any one of G1 to G12 can be chosen as the test machine. In this report, machine 3 is chosen as the test machine.

Simulations are carried out for the above two cases, for both with and without PSSs, to observe the damping provided by the PSSs. Simulations may be viewed by running 'Benchmark_IEEE_standard.mdl' (shown in Figure 7), after initialization using 'Init_MultiMachine.m', using the 'Run' button (green circle with black triangle) in the graphical interface of Simulink. Different cases may be chosen using the manual switch, as shown in the following figure. The graphs of the rotor slips of machines G3, G9 and G15, with respect to the slip of the swing machine (G16), are shown for the two cases in Figure 8 and Figure 9, respectively.

As expected, without PSS the oscillations are highly unstable. As in both the cases the disturbances are created close to machine G3, their effect is maximum on the oscillations of G3, and the effect decreases as we move away from G3, that is, the effect is lesser on the oscillations of G9 and least on G15. Also, the damping provided by the PSS is maximum for G3, lesser for G9 and least for G15 (as it does not have a PSS). The oscillations for all the machines are poorly damped, especially for G15, due to the presence of three pairs of poorly-damped inter-area modes.

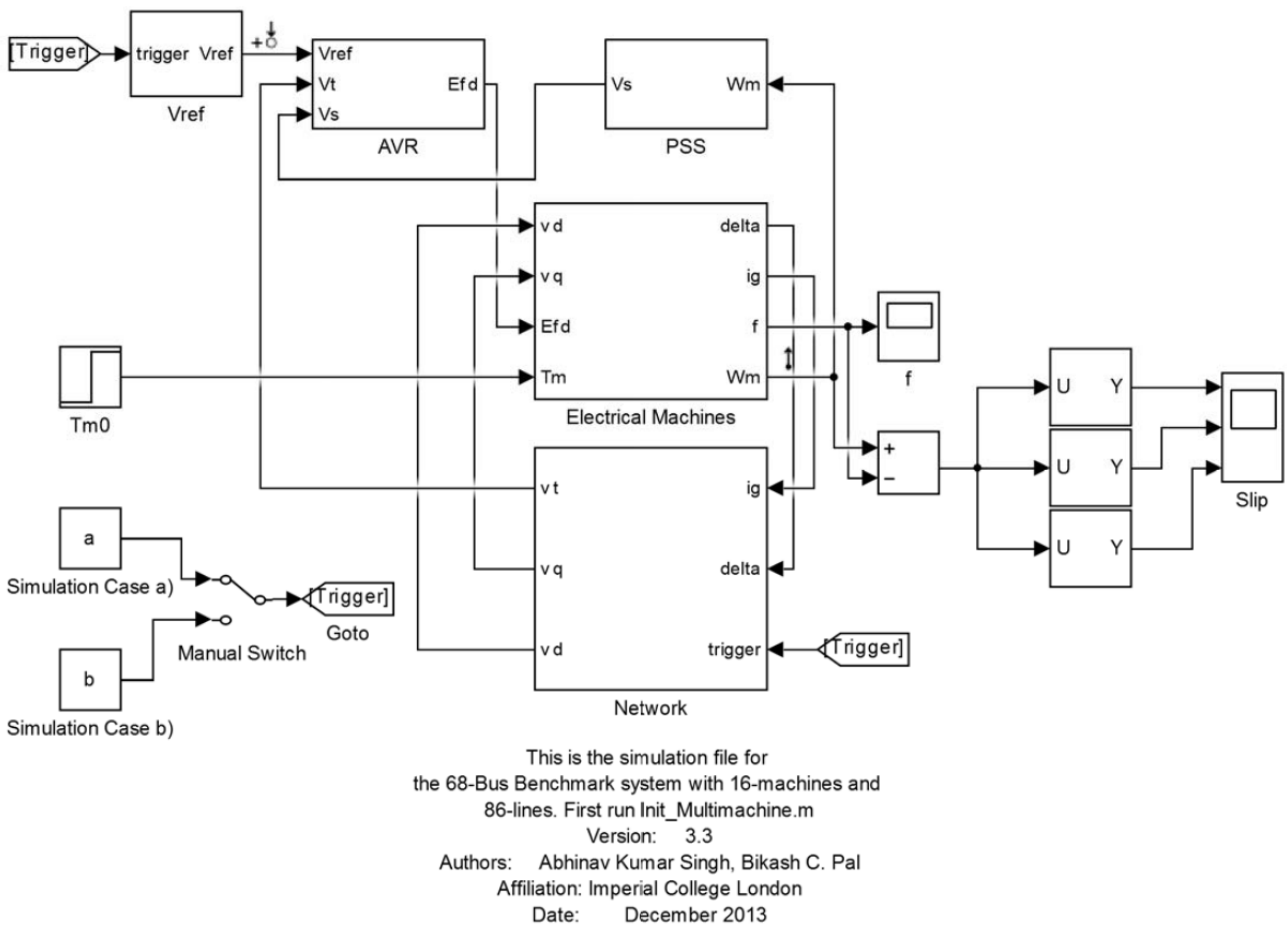

Figure 7 - Simulink model

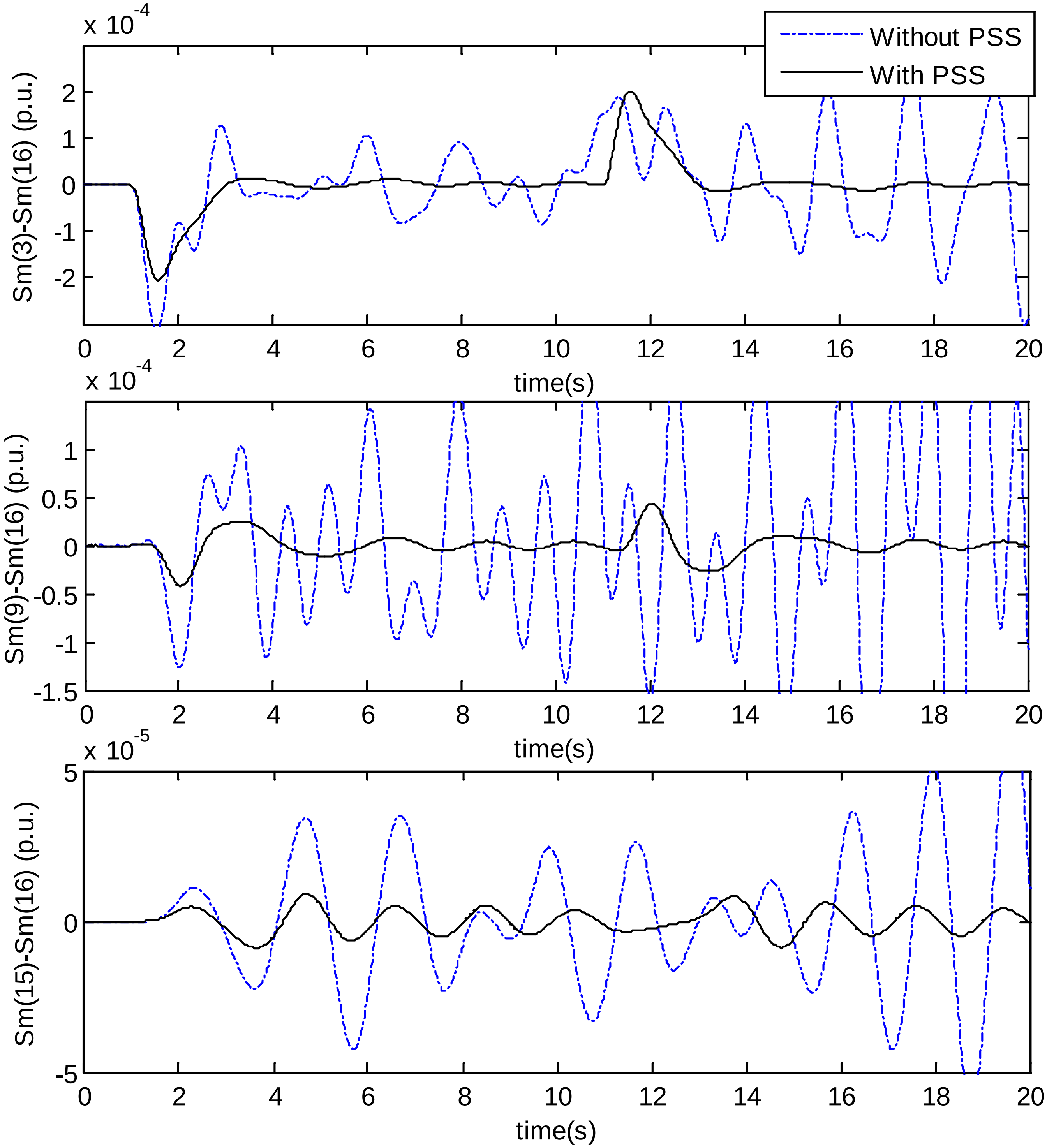

Figure 8 - Relative rotor slips for G3, G9 and G15 for case (a) (2% disturbance in Vref of G3)

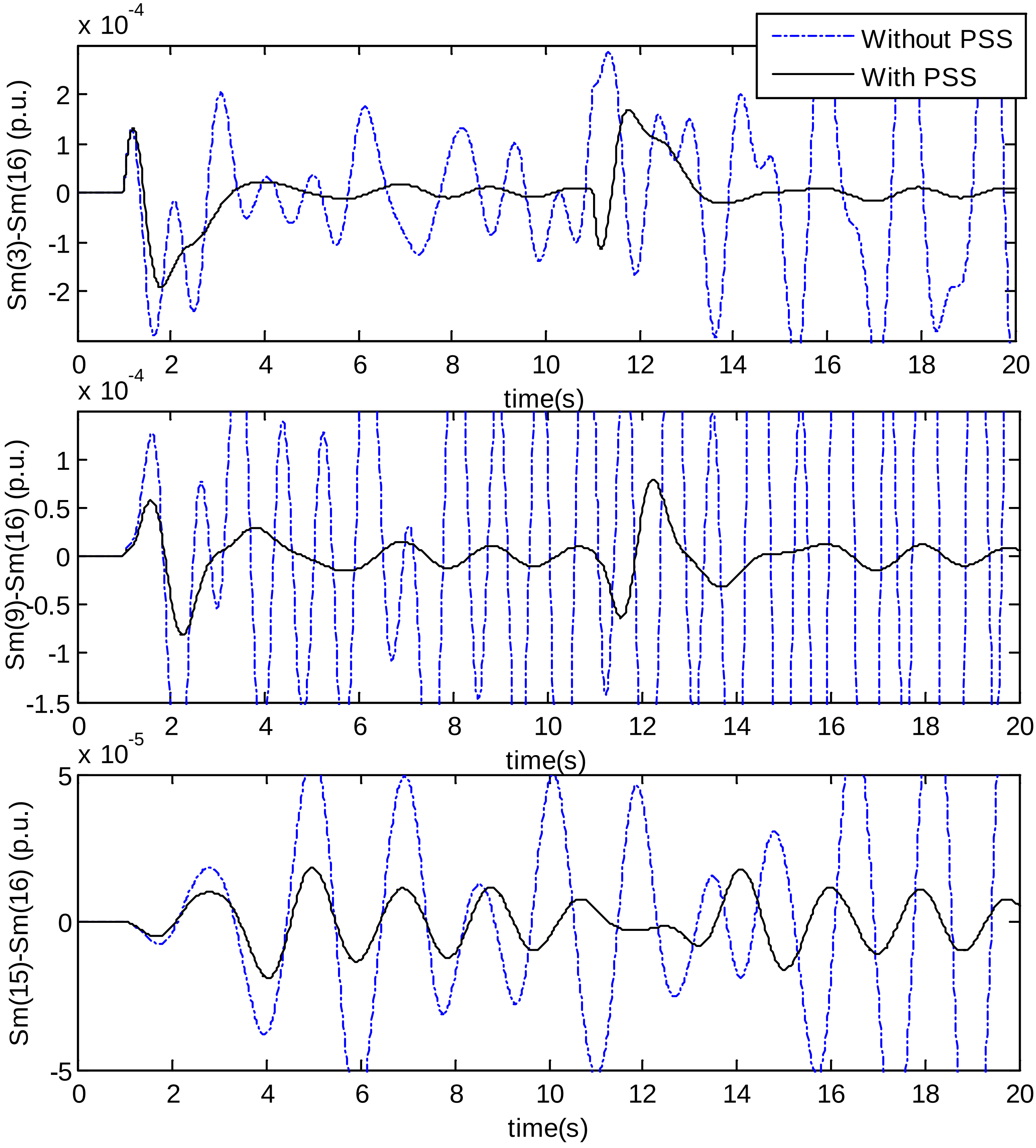

Figure 9 -Relative rotor slips for G3, G9 and G15 for case (b) (addition of 50 MVAr shunt reactor to bus 3)

IV. APPENDIX-A: MATLAB INITIALIZATION FILE 'INIT_MULTIMACHINE.M'

```
% This is the initialization and modal analysis file for
% the 68-Bus Benchmark system with 16-machines and
% 86-lines. It requires following files to successfully run:
% 1. calc.m
% 2. chq_lim.m
% 3. data16m_benchmark.m
% 4. form_jac.m
% 5. loadflow.m
% 6. y_sparse.m
% 7. Benchmark_IEEE_standard.mdl
% Version:     3.3
% Authors:     Abhinav Kumar Singh, Bikash C. Pal
% Affiliation: Imperial College London
% Date:        December 2013
clear all;
clc;
MVA_Base=100.0;
f=60.0;
deg_rad = pi/180.0;    % degree to radian,
rad_deg = 180.0/pi;  % radian to degree.
j=sqrt(-1);

%%%%%%%%%%Load Data%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
data16m_benchmark;
N_Machine=size(mac_con,1);
N_Bus=size(bus,1);
N_Line=size(line,1);
%%%%%%%%%%Loading Data Ends%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%Run Load Flow%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tol = 1e-12;iter_max = 50; vmin = 0.5; vmax = 1.5; acc = 1.0;
disply='y';flag = 2;
svolt = bus(:,2); stheta = bus(:,3)*deg_rad;
bus_type = round(bus(:,10));
swing_index=find(bus_type==1);
    [bus_sol,line_sol,line_flow,Y1,y,tps,chrg] = ...
        loadflow(bus,line,tol,iter_max,acc,disply,flag);
clc;
display('Running..');
%%%%%%%%%%Load Flow End%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%Initialize Machine Variables%%%%%%%%%%%%%%%%%%%%%
BM=mac_con(:,3)/MVA_Base;%uniform base conversion matrix
xls=mac_con(:,4)./BM;
Ra=mac_con(:,5)./BM;
xd=mac_con(:,6)./BM;
xdd=mac_con(:,7)./BM;
xddd=mac_con(:,8)./BM;
Td0d=mac_con(:,9);
Td0dd=mac_con(:,10);
xq=mac_con(:,11)./BM;
xqd=mac_con(:,12)./BM;
xqdd=mac_con(:,13)./BM;
Tq0d=mac_con(:,14);
Tq0dd=mac_con(:,15);
H=mac_con(:,16).*BM;
D=mac_con(:,17).*BM;
```

```
M=2*H;
wB=2*pi*f;
Tc=0.01*ones(N_Machine,1);
Vs=0.0*ones(N_Machine,1);
if xddd~=0 %#ok<BDSCI>
    Zg= Ra + j*xddd;%Zg for sub-transient model
else
    Zg= Ra + j*xdd;
end
Yg=1./Zg;
%%%%%%%%%Machine Variables initialization ends%%%%%%%%

%%%%%%%%%%%%%%%AVR Initialization%%%%%%%%%%%%%%%%%%%%%
Tr=ones(N_Machine,1);
KA=zeros(N_Machine,1);
Kp=zeros(N_Machine,1);
Ki=zeros(N_Machine,1);
Kd=zeros(N_Machine,1);
Td=ones(N_Machine,1);
Ka=ones(N_Machine,1);
Kad=zeros(N_Machine,1);%for DC4B Efd0 initialization
Ta=ones(N_Machine,1);
Ke=ones(N_Machine,1);
Aex=zeros(N_Machine,1);
Bex=zeros(N_Machine,1);
Te=ones(N_Machine,1);
Kf=zeros(N_Machine,1);
Tf=ones(N_Machine,1);
Efdmin=zeros(N_Machine,1);
Efdmax=zeros(N_Machine,1);
Efdmin_dc=zeros(N_Machine,1);
Efdmax_dc=zeros(N_Machine,1);
len_exc=size(exc_con);
Vref_Manual=ones(N_Machine,1);
for i=1:1:len_exc(1)
    Exc_m_indx=exc_con(i,2);%present machine index
    Vref_Manual(Exc_m_indx)=0;
    if(exc_con(i,3)~=0)
        Tr(Exc_m_indx)=exc_con(i,3);
    end;
    if(exc_con(i,5)~=0)
        Ta(Exc_m_indx)=exc_con(i,5);
    end;
    if(exc_con(i,1)==1)
        Kp(Exc_m_indx)=exc_con(i,16);
        Kd(Exc_m_indx)=exc_con(i,17);
        Ki(Exc_m_indx)=exc_con(i,18);
        Td(Exc_m_indx)=exc_con(i,19);
        Ke(Exc_m_indx)=exc_con(i,8);
        Te(Exc_m_indx)=exc_con(i,9);
        Kf(Exc_m_indx)=exc_con(i,14);
        Tf(Exc_m_indx)=exc_con(i,15);
        Bex(Exc_m_indx)=log(exc_con(i,11)/exc_con(i,13))/(exc_con(i,10)-
exc_con(i,12));
        Aex(Exc_m_indx)=exc_con(i,11)*exp(-Bex(Exc_m_indx)*exc_con(i,10));
        Ka(Exc_m_indx)=exc_con(i,4);
        Kad(Exc_m_indx)=1;
        Efdmin_dc(Exc_m_indx)=exc_con(i,7);
        Efdmax_dc(Exc_m_indx)=exc_con(i,6);
    else
        KA(Exc_m_indx)=exc_con(i,4);
```

```
        Efdmin(Exc_m_indx)=exc_con(i,7);
        Efdmax(Exc_m_indx)=exc_con(i,6);
    end;
end;
%%%%%%%%%%%%%%%%AVR initialization ends%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%PSS initialization%%%%%%%%%%%%%%%%%
Ks=zeros(N_Machine,1);
Tw=ones(N_Machine,1);
T11=ones(N_Machine,1);
T12=ones(N_Machine,1);
T21=ones(N_Machine,1);
T22=ones(N_Machine,1);
T31=ones(N_Machine,1);
T32=ones(N_Machine,1);
Vs_max=zeros(N_Machine,1);
Vs_min=zeros(N_Machine,1);
len_pss=size(pss_con);
for i=1:1:len_pss(1)
    Pss_m_indx=pss_con(i,2);%present machine index
    Ks(Pss_m_indx)=pss_con(i,3);%pssgain
    Tw(Pss_m_indx)=pss_con(i,4);%washout time constant
    T11(Pss_m_indx)=pss_con(i,5);%first lead time constant
    T12(Pss_m_indx)=pss_con(i,6);%first lag time constant
    T21(Pss_m_indx)=pss_con(i,7);%second lead time constant
    T22(Pss_m_indx)=pss_con(i,8);%second lag time constant
    T31(Pss_m_indx)=pss_con(i,9);%third lead time constant
    T32(Pss_m_indx)=pss_con(i,10);%third lag time constant
    Vs_max(Pss_m_indx)=pss_con(i,11);%maximum output limit
    Vs_min(Pss_m_indx)=pss_con(i,12);%minimum output limit
end
%%%%%%%%%%%%%%%%%%%%PSS initialization ends%%%%%%%%%%%%%

%%%%%%%%%%Initialize Network Variables%%%%%%%%%%%%%%%%%%
Y=full(Y1);%sparse to full matrix
MM=zeros(N_Bus,N_Machine,'double');%Multiplying matrix to convert Ig into vector of
correct length
for i=1:1:N_Machine
    MM(mac_con(i,2),mac_con(i,1))=1;
end
YG=MM*Yg;
YG=diag(YG);
V=bus_sol(:,2);
theta=bus_sol(:,3)*pi/180;
YL=diag((bus(:,6)-j*bus(:,7))./V.^2);%Constant Impedence Load model
Y_Aug=Y+YL+YG;
Z=inv(Y_Aug);
Y_Aug_dash=Y_Aug;
%%%%%%%%%%Network Variables initialization ends%%%%%%%%

%%%%%%%%%%%%%%%%%%Initial Conditions%%%%%%%%%%%%%%%%%%%%
Vg=MM'*V;thg=MM'*theta;
P=MM'*bus_sol(:,4);Q=MM'*bus_sol(:,5);
mp=2;
mq=2;
kp=bus_sol(:,6)./V.^mp;
kq=bus_sol(:,7)./V.^mq;
V0=V.*(cos(theta)+ j*sin(theta));
y_dash=y;
from_bus = line(:,1);
```

```
to_bus = line(:,2);
MW_s = V0(from_bus).*conj((V0(from_bus) - tps.*V0(to_bus)).*y_dash ...
    + V0(from_bus).*(j*chrg/2))./(tps.*conj(tps));
P_s = real(MW_s);      % active power sent out by from_bus
Q_s = imag(MW_s);
voltage = Vg.*(cos(thg) + j*sin(thg));
current = conj((P+j*Q)./voltage);
Eq0 = voltage + (Ra+j.*xq).*current;
id0 = -abs(current) .* (sin(angle(Eq0) - angle(current)));
iq0 = abs(current) .* cos(angle(Eq0) - angle(current));
vd0 = -abs(voltage) .* (sin(angle(Eq0) - angle(voltage)));
vq0 = abs(voltage) .* cos(angle(Eq0) - angle(voltage));
Efd0 = abs(Eq0) - (xd-xq).*id0;
Eq_dash0 = Efd0 + (xd - xdd) .* id0;
Ed_dash0 = -(xq-xqd) .* iq0;
Psi1d0=Eq_dash0+(xdd-xls).*id0;
Psi2q0=-Ed_dash0+(xqd-xls).*iq0;
Edc_dash0=(xddd-xqdd).*iq0;
Te0 = Eq_dash0.*iq0.*(xddd-xls)./(xdd-xls) + Ed_dash0.*id0.*(xqdd-xls)./(xqd-
xls)+(xddd-xqdd).*id0.*iq0 - Psi2q0.*id0.*(xqd-xqdd)./(xqd-xls) + Psi1d0.*iq0.*(xdd-
xddd)./(xdd-xls);
delta0 = angle(Eq0);
IG0 = (Yg.*(vq0+j.*vd0)+ (iq0+j.*id0)).*exp(j.*delta0);
IQ0 = real((iq0+j.*id0).*exp(j.*delta0));
ID0 = imag((iq0+j.*id0).*exp(j.*delta0));
VDQ=MM'*(Y_Aug_dash\(MM*IG0));
VD0=imag(VDQ);
VQ0=real(VDQ);
Vref=Efd0;
V_Ka0=zeros(N_Machine,1);
V_Ki0=zeros(N_Machine,1);
for i=1:1:len_exc(1)
    Exc_m_indx=exc_con(i,2);
    if(exc_con(i,1)==1)

V_Ka0(Exc_m_indx)=Efd0(Exc_m_indx)*(Ke(Exc_m_indx)+Aex(Exc_m_indx)*exp(Efd0(Exc_m_ind
x)*Bex(Exc_m_indx)));
        V_Ki0(Exc_m_indx)=V_Ka0(Exc_m_indx)/Ka(Exc_m_indx);
        Vref(Exc_m_indx)=Vg(Exc_m_indx);
    else
        Vref(Exc_m_indx)=(Efd0(Exc_m_indx)/KA(Exc_m_indx))+Vg(Exc_m_indx);
    end
end
Tm0=Te0;
Pm0=Tm0;
Sm0=0.0*ones(N_Machine,1);
%%%%%%%%%%%%%%%%%%Initial Conditions end%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%Creating simulation cases' disturbances%%%%%%%%
% a) A 20 s simulation, with a 2% step in Vref of test machine at
%    t = 1.0 s and a -2% step in the same Vref at t = 11.0 s;
% b) The connection, at t = 1.0 s, of a 50 MVAr shunt reactor to the
%    bus corresponding to the test machine, removing this reactor
%    at t = 11.0 s and simulating until t = 20 s.
%%%%%%%%%%%%%%Case a)%%%%%%%%%%%%%%%
a=0; %for case a
c=3; %test machine number
dVref=zeros(N_Machine,1);
dVref(c,1)=.02*Vref(c,1);
%%%%%%%%%%%%%%Case b)%%%%%%%%%%%%%%%
b=1;  %for case b
```

```
Yd=Y_Aug;
Yd(c,c)=Yd(c,c)+(-j*50/100)/(V(c)^2);%adding 50 MVAr to the bus corresponding to the
swing machine
Zd=inv(Yd);
%%%%%%%%%%%%%%%%%%Ending simulation cases' disturbances%%%%%%%%%

%%%%%%%%%%%%%%Linearization%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
timeop=0;
io = getlinio('Benchmark_IEEE_standard');
op = findop('Benchmark_IEEE_standard',timeop);
sys=linearize('Benchmark_IEEE_standard',op,io);
A=sys.a;
B=sys.b;
C=sys.c;
x_state=sys.StateName;
[rv,lambda]=eig(A,'nobalance');
lambda=diag(lambda);
lv = inv(rv);
freq=abs(imag(lambda))/(2*pi);%in hz
omega=abs(imag(lambda));%in rad/s
damping_ratio=-real(lambda)./abs(lambda);
N_State=size(A,1);
for i=1:1:N_State
    if(abs(lambda(i))<=1e-10)
    damping_ratio(i)=1;
    end
end
[Dr,Idx]=sort(damping_ratio*100);
%%%%%%%%%%%%%%Linearization ends%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%Particicipation Factor Analysis%%%%%%%%%%%
pf1=zeros(N_State);
pf=zeros(N_State);
for i=1:N_State
    pf1(i,:)=(rv(:,i).*(lv(i,:)'))';
    pf(i,:)=abs(pf1(i,:)/max(pf1(i,:)));
end
[PF,Indx]=sort(pf,2,'descend');
st=[x_state(Indx(:,1)) x_state(Indx(:,2)) x_state(Indx(:,3)) x_state(Indx(:,4))
x_state(Indx(:,5)) x_state(Indx(:,6)) x_state(Indx(:,7)) x_state(Indx(:,8))
x_state(Indx(:,9)) x_state(Indx(:,10))];
pfac=[PF(:,1) PF(:,2) PF(:,3) PF(:,4) PF(:,5) PF(:,6) PF(:,7) PF(:,8) PF(:,9)
PF(:,10)];
%%%%%%%%%%%%%%%%%%Particicipation Factor Analysis ends%%%%%%

%%%arranging modes to have interarea modes at top, then local, then rest%%%
np=0;
for i=1:N_State
    ch1=char(x_state(Indx(i,1)));
    ch2=char(x_state(Indx(i,2)));
    if ((ch1(1)=='S' || ch1(1)=='D') && imag(lambda(i)) < 0)
        np=np+1;
    end
end
freq_red=zeros(np,2);
l=0;
for i=1:N_State
    ch1=char(x_state(Indx(i,1)));
    ch2=char(x_state(Indx(i,2)));
    if ((ch1(1)=='S' || ch1(1)=='D') && imag(lambda(i)) < 0)
```

```
        l=l+1;
        freq_red(l,1)=freq(i);
        freq_red(l,2)=i;
    end
end
[fr,idx]=sort(freq_red);
mode_sort=cell(np,10);
l=0;
for i=1:np
    mode_sort(i,:)=[round(100000*damping_ratio(freq_red(idx(i),2)))/1000
round(1000*freq(freq_red(idx(i),2)))/1000 ...
        st(freq_red(idx(i),2),1) round(1000*pfac(freq_red(idx(i),2),1))/1000
st(freq_red(idx(i),2),2) ...
        round(1000*pfac(freq_red(idx(i),2),2))/1000 st(freq_red(idx(i),2),3)
round(1000*pfac(freq_red(idx(i),2),3))/1000 ...
        st(freq_red(idx(i),2),4) round(1000*pfac(freq_red(idx(i),2),4))/1000];
end
%%%arranging modes ends%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%Finding poorly damped modes and PFs of speed deviations in
them%%%%%%
n_inter_area=0;
for i=1:N_State
    ch2=char(x_state(Indx(i,1)));
    if ((ch2(1)=='S' || ch2(1)=='D') && freq(i) < .9 && imag(lambda(i)) > 0)
        n_inter_area=n_inter_area+1;
    end
end
pf2=zeros(n_inter_area,N_Machine);
ms2=zeros(n_inter_area,N_Machine);
ms3=zeros(n_inter_area,N_Machine);
st2=cell(n_inter_area,N_Machine);
eig2=zeros(n_inter_area,2);
m=0;
for i=1:N_State
    ch2=char(x_state(Indx(i,1)));
    if ((ch2(1)=='S' || ch2(1)=='D') && freq(i) < .9 && imag(lambda(i)) > 0)
        m=m+1;
        eig2(m,1)=damping_ratio(i);
        eig2(m,2)=freq(i);
        l=0;
        for k=1:N_State
            ch=char(x_state(Indx(i,k)));
            if ch(1)=='S'
                l=l+1;
                st2(m,l)=x_state(Indx(i,k));
                pf2(m,l)=PF(i,k);
            end
        end
        l=0;
        for k=1:N_State
            ch=char(x_state(k));
            if ch(1)=='S'
                l=l+1;
                ms2(m,l)=angle(rv(k,i));%mode shape
                ms3(m,l)=rv(k,i);
            end
        end
    end
end
st2=st2';
```

```
pf2=(round(10000*pf2)/10000)';
eig2=eig2';
%%%%%%%%%%%%%%%%%%Ending of finding poorly damped modes and PFs of speed deviations
in them%%%%%%

%%%%%%%%%%%%%%%%%%%%%%Eigenvalue Plot%%%%%%%%%%%%%%%%%%%%%%
figure(1);
hold off;
y=-12:.01:12;
x=-.1*abs(y);
plot(x,y, '-.');
hold on;
grid on;
plot(lambda, '*');
xlim([-3.3 0.2]);
ylim([-12 12]);
xlabel 'Real part'
ylabel 'Imaginary part'
%%%%%%%%%%%%%%%%%%%%%%Eigenvalue Plot ends%%%%%%%%%%%%%%%%%%

% %%%%%%%%%%%%%%%%%%%%%%Mode shape Plot%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
for i=2:-1:1
    subplot(2,2,4-i+1);
    compass(ms3(i,:))
    subplot(2,2,2-i+1);
    feather(ms3(i,:))
    xlim([0 N_Machine+1]);
    xlabel 'Machine No.'
    ylabel 'Polar plot'
    title(sprintf('Mode %d \n damping ratio = %3.2f percent \n frequency = %3.2f
Hz',4-i+1,round(10000*eig2(1,i))/100,round(1000*eig2(2,i))/1000));
end

figure(3);
for i=4:-1:3
    subplot(2,2,6-i+1);
    compass(ms3(i,:))
    subplot(2,2,4-i+1);
    feather(ms3(i,:))
    xlim([0 N_Machine+1]);
    xlabel 'Machine No.'
    ylabel 'Polar plot'
    title(sprintf('Mode %d \n damping ratio = %3.2f percent \n frequency = %3.2f
Hz',4-i+1,round(10000*eig2(1,i))/100,round(1000*eig2(2,i))/1000));
end
%%%%%%%%%%%%%%%%%%%%%%Mode shape Plot ends%%%%%%%%%%%%%%%%%%
clc;
```

V. APPENDIX-B: 68-BUS SYSTEM DATA FILE 'DATA16M_BENCHMARK.M'

```
% This is the data file for the 68-Bus Benchmark system with 16-machines
% and 86-lines. The data is partially taken from the book "Robust Control
% in Power Systems" by B. Pal and B. Chaudhuri, with some of the parameters
% modified to account for a more realistic model.
% Version:     3.3
% Authors:     Abhinav Kumar Singh, Bikash C. Pal
```

```
% Affiliation: Imperial College London
% Date:        December 2013

%************************ BUS DATA STARTS ********************************
% bus data format
% bus: number, voltage(pu), angle(degree), p_gen(pu), q_gen(pu),
%      p_load(pu), q_load(pu),G-shunt (p.u), B shunt (p.u); bus_type
%      bus_type - 1, swing bus
%               - 2, generator bus (PV bus)
%               - 3, load bus (PQ bus)
system_base_mva = 100.0;
bus = [...
    01 1.045   0.00   2.50   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    02 0.98    0.00   5.45   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    03 0.983   0.00   6.50   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    04 0.997   0.00   6.32   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    05 1.011   0.00   5.05   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    06 1.050   0.00   7.00   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    07 1.063   0.00   5.60   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    08 1.03    0.00   5.40   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    09 1.025   0.00   8.00   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    10 1.010   0.00   5.00   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    11 1.000   0.00  10.000  0.00   0.00   0.00    0.00 0.00  2  999  -999;
    12 1.0156  0.00  13.50   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    13 1.011   0.00  35.91   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    14 1.00    0.00  17.85   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    15 1.000   0.00  10.00   0.00   0.00   0.00    0.00 0.00  2  999  -999;
    16 1.000   0.00  40.00   0.00   0.00   0.00    0.00 0.00  1  0   0;
    17 1.00    0.00   0.00   0.00  60.00   3.00    0.00 0.00  3  0   0;
    18 1.00    0.00   0.00   0.00  24.70   1.23    0.00 0.00  3  0   0;
    19 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    20 1.00    0.00   0.00   0.00   6.800  1.03    0.00 0.00  3  0   0;
    21 1.00    0.00   0.00   0.00   2.740  1.15    0.00 0.00  3  0   0;
    22 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    23 1.00    0.00   0.00   0.00   2.480  0.85    0.00 0.00  3  0   0;
    24 1.00    0.00   0.00   0.00   3.09  -0.92    0.00 0.00  3  0   0;
    25 1.00    0.00   0.00   0.00   2.24   0.47    0.00 0.00  3  0   0;
    26 1.00    0.00   0.00   0.00   1.39   0.17    0.00 0.00  3  0   0;
    27 1.00    0.00   0.00   0.00   2.810  0.76    0.00 0.00  3  0   0;
    28 1.00    0.00   0.00   0.00   2.060  0.28    0.00 0.00  3  0   0;
    29 1.00    0.00   0.00   0.00   2.840  0.27    0.00 0.00  3  0   0;
    30 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    31 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    32 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    33 1.00    0.00   0.00   0.00   1.12   0.00    0.00 0.00  3  0   0;
    34 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    35 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    36 1.00    0.00   0.00   0.00   1.02  -0.1946  0.00 0.00  3  0   0;
    37 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    38 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    39 1.00    0.00   0.00   0.00   2.67   0.126   0.00 0.00  3  0   0;
    40 1.00    0.00   0.00   0.00   0.6563 0.2353  0.00 0.00  3  0   0;
    41 1.00    0.00   0.00   0.00  10.00   2.50    0.00 0.00  3  0   0;
    42 1.00    0.00   0.00   0.00  11.50   2.50    0.00 0.00  3  0   0;
    43 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0   0;
    44 1.00    0.00   0.00   0.00   2.6755 0.0484  0.00 0.00  3  0   0;
    45 1.00    0.00   0.00   0.00   2.08   0.21    0.00 0.00  3  0   0;
    46 1.00    0.00   0.00   0.00   1.507  0.285   0.00 0.00  3  0   0;
    47 1.00    0.00   0.00   0.00   2.0312 0.3259  0.00 0.00  3  0   0;
    48 1.00    0.00   0.00   0.00   2.4120 0.022   0.00 0.00  3  0   0;
    49 1.00    0.00   0.00   0.00   1.6400 0.29    0.00 0.00  3  0   0;
```

```
    50 1.00    0.00   0.00   0.00   1.00  -1.47    0.00 0.00  3  0  0;
    51 1.00    0.00   0.00   0.00   3.37  -1.22    0.00 0.00  3  0  0;
    52 1.00    0.00   0.00   0.00   1.58   0.30    0.00 0.00  3  0  0;
    53 1.00    0.00   0.00   0.00   2.527  1.1856  0.00 0.00  3  0  0;
    54 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    55 1.00    0.00   0.00   0.00   3.22   0.02    0.00 0.00  3  0  0;
    56 1.00    0.00   0.00   0.00   2.00  0.736    0.00 0.00  3  0  0;
    57 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    58 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    59 1.00    0.00   0.00   0.00   2.34   0.84    0.00 0.00  3  0  0;
    60 1.00    0.00   0.00   0.00  2.088  0.708    0.00 0.00  3  0  0;
    61 1.00    0.00   0.00   0.00   1.04   1.25    0.00 0.00  3  0  0;
    62 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    63 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    64 1.00    0.00   0.00   0.00   0.09   0.88    0.00 0.00  3  0  0;
    65 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    66 1.00    0.00   0.00   0.00   0.00   0.00    0.00 0.00  3  0  0;
    67 1.00    0.00   0.00   0.00   3.200  1.5300  0.00 0.00  3  0  0;
    68 1.00    0.00   0.00   0.00   3.290  0.32    0.00 0.00  3  0  0];
%************************ BUS DATA ENDS ********************************

%************************ LINE DATA STARTS ******************************
line =[...
    01   54        0    0.0181         0    1.0250         0;
    02   58        0    0.0250         0    1.0700         0;
    03   62        0    0.0200         0    1.0700         0;
    04   19    0.0007   0.0142         0    1.0700         0;
    05   20    0.0009   0.0180         0    1.0090         0;
    06   22        0    0.0143         0    1.0250         0;
    07   23    0.0005   0.0272         0         0         0;
    08   25    0.0006   0.0232         0    1.0250         0;
    09   29    0.0008   0.0156         0    1.0250         0;
    10   31        0    0.0260         0    1.0400         0;
    11   32        0    0.0130         0    1.0400         0;
    12   36        0    0.0075         0    1.0400         0;
    13   17        0    0.0033         0    1.0400         0;
    14   41        0    0.0015         0    1.0000         0;
    15   42        0    0.0015         0    1.0000         0;
    16   18        0    0.0030         0    1.0000         0;
    17   36    0.0005   0.0045    0.3200         0         0;
    18   49    0.0076   0.1141    1.1600         0         0;
    18   50    0.0012   0.0288    2.0600         0         0;
    19   68    0.0016   0.0195    0.3040         0         0;
    20   19    0.0007   0.0138         0    1.0600         0;
    21   68    0.0008   0.0135    0.2548         0         0;
    22   21    0.0008   0.0140    0.2565         0         0;
    23   22    0.0006   0.0096    0.1846         0         0;
    24   23    0.0022   0.0350    0.3610         0         0;
    24   68    0.0003   0.0059    0.0680         0         0;
    25   54    0.0070   0.0086    0.1460         0         0;
    26   25    0.0032   0.0323    0.5310         0         0;
    27   37    0.0013   0.0173    0.3216         0         0;
    27   26    0.0014   0.0147    0.2396         0         0;
    28   26    0.0043   0.0474    0.7802         0         0;
    29   26    0.0057   0.0625    1.0290         0         0;
    29   28    0.0014   0.0151    0.2490         0         0;
    30   53    0.0008   0.0074    0.4800         0         0;
    30   61    0.00095  0.00915   0.5800         0         0;
    31   30    0.0013   0.0187    0.3330         0         0;
    31   53    0.0016   0.0163    0.2500         0         0;
    32   30    0.0024   0.0288    0.4880         0         0;
```

```
    33   32    0.0008    0.0099    0.1680         0         0;
    34   33    0.0011    0.0157    0.2020         0         0;
    34   35    0.0001    0.0074         0    0.9460         0;
    36   34    0.0033    0.0111    1.4500         0         0;
    36   61    0.0011    0.0098    0.6800         0         0;
    37   68    0.0007    0.0089    0.1342         0         0;
    38   31    0.0011    0.0147    0.2470         0         0;
    38   33    0.0036    0.0444    0.6930         0         0;
    40   41    0.0060    0.0840    3.1500         0         0;
    40   48    0.0020    0.0220    1.2800         0         0;
    41   42    0.0040    0.0600    2.2500         0         0;
    42   18    0.0040    0.0600    2.2500         0         0;
    43   17    0.0005    0.0276         0         0         0;
    44   39         0    0.0411         0         0         0;
    44   43    0.0001    0.0011         0         0         0;
    45   35    0.0007    0.0175    1.3900         0         0;
    45   39         0    0.0839         0         0         0;
    45   44    0.0025    0.0730         0         0         0;
    46   38    0.0022    0.0284    0.4300         0         0;
    47   53    0.0013    0.0188    1.3100         0         0;
    48   47    0.00125   0.0134    0.8000         0         0;
    49   46    0.0018    0.0274    0.2700         0         0;
    51   45    0.0004    0.0105    0.7200         0         0;
    51   50    0.0009    0.0221    1.6200         0         0;
    52   37    0.0007    0.0082    0.1319         0         0;
    52   55    0.0011    0.0133    0.2138         0         0;
    54   53    0.0035    0.0411    0.6987         0         0;
    55   54    0.0013    0.0151    0.2572         0         0;
    56   55    0.0013    0.0213    0.2214         0         0;
    57   56    0.0008    0.0128    0.1342         0         0;
    58   57    0.0002    0.0026    0.0434         0         0;
    59   58    0.0006    0.0092    0.1130         0         0;
    60   57    0.0008    0.0112    0.1476         0         0;
    60   59    0.0004    0.0046    0.0780         0         0;
    61   60    0.0023    0.0363    0.3804         0         0;
    63   58    0.0007    0.0082    0.1389         0         0;
    63   62    0.0004    0.0043    0.0729         0         0;
    63   64    0.0016    0.0435         0    1.0600         0;
    65   62    0.0004    0.0043    0.0729         0         0;
    65   64    0.0016    0.0435         0    1.0600         0;
    66   56    0.0008    0.0129    0.1382         0         0;
    66   65    0.0009    0.0101    0.1723         0         0;
    67   66    0.0018    0.0217    0.3660         0         0;
    68   67    0.0009    0.0094    0.1710         0         0;
    27   53    0.0320    0.3200    0.4100         0         0;
    ];
%*********************** LINE DATA ENDS ******************************

% ********************** MACHINE DATA STARTS ***************************
% 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
%       20. saturation factor S(1.0)
%       21. saturation factor S(1.2)
% note: all the following machines use subtransient reactance model
mac_con = [
    01 01  100 0.0125 0.0  0.1    0.031   0.025    10.2  0.05 0.069  0.0416667
0.025   1.5   0.035 42.   0  0  01   0 0;
    02 02  100 0.035  0.0  0.295  0.0697  0.05     6.56  0.05 0.282  0.0933333   0.05
1.5   0.035 30.2  0  0  02   0 0;
    03 03  100 0.0304 0.0  0.2495 0.0531  0.045    5.7   0.05 0.237  0.0714286
0.045   1.5   0.035 35.8  0  0  03   0 0;
    04 04  100 0.0295 0.0  0.262  0.0436  0.035    5.69  0.05 0.258  0.0585714
0.035   1.5   0.035 28.6  0  0  04   0 0;
    05 05  100 0.027  0.0  0.33   0.066   0.05     5.4   0.05 0.31   0.0883333   0.05
0.44  0.035 26.   0  0  05   0 0;
    06 06  100 0.0224 0.0  0.254  0.05    0.04     7.3   0.05 0.241  0.0675000   0.04
0.4   0.035 34.8  0  0  06   0 0;
    07 07  100 0.0322 0.0  0.295  0.049   0.04     5.66  0.05 0.292  0.0666667   0.04
1.5   0.035 26.4  0  0  07   0 0;
    08 08  100 0.028  0.0  0.29   0.057   0.045    6.7   0.05 0.280  0.0766667
0.045   0.41  0.035 24.3  0  0  08   0 0;
    09 09  100 0.0298 0.0  0.2106 0.057   0.045    4.79  0.05 0.205  0.0766667
0.045   1.96  0.035 34.5  0  0  09   0 0;
    10 10  100 0.0199 0.0  0.169  0.0457  0.04     9.37  0.05 0.115  0.0615385   0.04
1.5   0.035 31.0  0  0  10   0 0;
    11 11  100 0.0103 0.0  0.128  0.018   0.012    4.1   0.05 0.123  0.0241176
0.012   1.5   0.035 28.2  0  0  11   0 0;
    12 12  100 0.022  0.0  0.101  0.031   0.025    7.4   0.05 0.095  0.0420000
0.025   1.5   0.035 92.3  0  0  12   0 0;
    13 13  200 0.0030 0.0  0.0296 0.0055  0.004    5.9   0.05 0.0286 0.0074000
0.004   1.5   0.035 248.0 0  0  13   0 0;
    14 14  100 0.0017 0.0  0.018  0.00285 0.0023   4.1   0.05 0.0173 0.0037931
0.0023  1.5   0.035 300.0 0  0  14   0 0;
    15 15  100 0.0017 0.0  0.018  0.00285 0.0023   4.1   0.05 0.0173 0.0037931
0.0023  1.5   0.035 300.0 0  0  15   0 0;
    16 16  200 0.0041 0.0  0.0356 0.0071  0.0055   7.8   0.05 0.0334 0.0095000
0.0055  1.5   0.035 225.0 0  0  16   0 0;
                            ] ;
% ********************** MACHINE DATA ENDS *************************

% *********************** EXCITER DATA STARTS ***********************
% Description of Exciter data starts
% exciter data DC4B,ST1A model
%      1 - exciter type (1 for DC4B, 0 for ST1A)
%      2 - machine number
%      3 - input filter time constant T_R
%      4 - voltage regulator gain K_A
%      5 - voltage regulator time constant T_A
%      6 - maximum voltage regulator output V_Rmax
%      7 - minimum voltage regulator output V_Rmin
%      8 - exciter constant K_E
%      9 - exciter time constant T_E
```

```
%    10 - E_1
%    11 - S(E_1)
%    12 - E_2
%    13 - S(E_2)
%    14 - stabilizer gain K_F
%    15 - stabilizer time constant T_F
%    16 - K_P
%    17 - K_I
%    18 - K_D
%    19 - T_D
exc_con = [...
    1 1  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 2  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 3  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 4  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 5  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 6  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 7  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 8  0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    0 9  0.01 200. 0.00 5.0  -5.0  0.0     0   0      0     0      0     0     0   0
0  0    0;
    1 10 0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 11 0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
    1 12 0.01 1.   0.02 10.  -10.  1.0    .785 3.9267 0.070 5.2356 0.910 0.030 1.0
200 50 50 .01;
          ];
 %************************ EXCITER DATA ENDS ************************

% ************************ PSS DATA STARTS ************************
%1-S. No.
%2-present machine index
%3-pssgain
%4-washout time constant
%5-first lead time constant
%6-first lag time constant
%7-second lead time constant
%8-second lag time constant
%9-third lead time constant
%10-third lag time constant
%11-maximum output limit
%12-minimum output limit
pss_con = [
    1   1   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    2   2   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    3   3   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    4   4   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    5   5   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    6   6   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    7   7   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    8   8   20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    9   9   12  10  0.09  0.02  0.09  0.02  1     1    0.2 -0.05 ;
```

```
    10  10  20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    11  11  20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    12  12  20  15  0.15  0.04  0.15  0.04  0.15  0.04 0.2 -0.05 ;
    ];
%*********************** PSS DATA ENDS ***********************
```

VI. APPENDIX-C: LOADFLOW FUNCTION FILE 'LOADFLOW.M'

```
function [bus_sol,line_sol,line_flow,Y,y,tps,chrg] = ...
   loadflow(bus,line,tol,iter_max,acc,display,flag)
% Syntax:    [bus_sol,line_sol,line_flow] =
% loadflow(bus,line,tol,iter_max,acc,display,flag)
% 8/12/97
% Purpose:   solve the load-flow equations of power systems
%            modified to eliminate do loops and improve the use
%            sparse matices
% Input:    bus       - bus data
%           line      - line data
%           tol       - tolerance for convergence
%           iter_max  - maximum number of iterations
%           acc       - acceleration factor
%           display   - 'y', generate load-flow study report
%                        else, no load-flow study report
%           flag      - 1, form new Jacobian every iteration
%                       2, form new Jacobian every other
%                           iteration
%
% Output:   bus_sol   - bus solution (see report for the
%                         solution format)
%           line_sol  - modified line matrix
%           line_flow - line flow solution (see report)
%
% See also:
%
% Algorithm: Newton-Raphson method using the polar form of
%   the equations for P(real power) and Q(reactive power).
%
% Calls:     Y_sparse, calc, form_jac chq_lim
%
%
% (c) Copyright 1991 Joe H. Chow - All Rights Reserved
%
% History (in reverse chronological order)
% Modification to correct generator var error on output
% Graham Rogers November 1997
% Version:   2.1
% Author:     Graham Rogers
% Date:      October 1996
% Purpose:   To add generator var limits and on-load tap changers
% Version:   2.0
% Author:    Graham Rogers
% Date:      March 1994
% Version:   1.0
% Authors:   Kwok W. Cheung, Joe H. Chow
% Date:      March 1991
%
% ***********************************************************
global bus_int
global Qg bus_type g_bno PQV_no PQ_no ang_red volt_red
global Q Ql
```

```
global gen_chg_idx
global ac_line n_dcl
tt = clock;     % start the total time clock
jay = sqrt(-1);
load_bus = 3;
gen_bus = 2;
swing_bus = 1;
if exist('flag') == 0
   flag = 1;
end
lf_flag = 1;
% set solution defaults
if isempty(tol);tol = 1e-9;end
if isempty(iter_max);iter_max = 30;end
if isempty(acc);acc = 1.0; end;
if isempty(display);display = 'n';end;

if flag <1 || flag > 2
   error('LOADFLOW: flag not recognized')
end
[nline nlc] = size(line);    % number of lines and no of line cols
[nbus ncol] = size(bus);     % number of buses and number of col
% set defaults
% bus data defaults
if ncol<15
   % set generator var limits
   if ncol<12
      bus(:,11) = 9999*ones(nbus,1);
      bus(:,12) = -9999*ones(nbus,1);
   end
   if ncol<13;bus(:,13) = ones(nbus,1);end
   bus(:,14) = 1.5*ones(nbus,1);
   bus(:,15) = 0.5*ones(nbus,1);
   volt_min = bus(:,15);
   volt_max = bus(:,14);
else
   volt_min = bus(:,15);
   volt_max = bus(:,14);
end
no_vmin_idx = find(volt_min==0);
if ~isempty(no_vmin_idx)
   volt_min(no_vmin_idx) = 0.5*ones(length(no_vmin_idx),1);
end
no_vmax_idx = find(volt_max==0);
if ~isempty(no_vmax_idx)
   volt_max(no_vmax_idx) = 1.5*ones(length(no_vmax_idx),1);
end
no_mxv = find(bus(:,11)==0);
no_mnv = find(bus(:,12)==0);
if ~isempty(no_mxv);bus(no_mxv,11)=9999*ones(length(no_mxv),1);end
if ~isempty(no_mnv);bus(no_mnv,12) = -9999*ones(length(no_mnv),1);end
no_vrate = find(bus(:,13)==0);
if ~isempty(no_vrate);bus(no_vrate,13) = ones(length(no_vrate),1);end
tap_it = 0;
tap_it_max = 10;
no_taps = 0;
% line data defaults, sets all tap ranges to zero - this fixes taps
if nlc < 10
   line(:,7:10) = zeros(nline,4);
   no_taps = 1;
   % disable tap changing
```

```
end

% outer loop for on-load tap changers

mm_chk=1;
while (tap_it<tap_it_max&&mm_chk)
   tap_it = tap_it+1;
   % build admittance matrix Y
   [Y,nSW,nPV,nPQ,SB] = y_sparse(bus,line);
   % process bus data
   bus_no = bus(:,1);
   V = bus(:,2);
   ang = bus(:,3)*pi/180;
   Pg = bus(:,4);
   Qg = bus(:,5);
   Pl = bus(:,6);
   Ql = bus(:,7);
   Gb = bus(:,8);
   Bb = bus(:,9);
   bus_type = round(bus(:,10));
   qg_max = bus(:,11);
   qg_min = bus(:,12);
   sw_bno=ones(nbus,1);
   g_bno=sw_bno;
   % set up index for Jacobian calculation
   %% form PQV_no and PQ_no
   bus_zeros=zeros(nbus,1);
   swing_index=find(bus_type==1);
   sw_bno(swing_index)=bus_zeros(swing_index);
   PQV_no=find(bus_type >=2);
   PQ_no=find(bus_type==3);
   gen_index=find(bus_type==2);
   g_bno(gen_index)=bus_zeros(gen_index);
   %sw_bno is a vector having ones everywhere but the swing bus locations
   %g_bno is a vector having ones everywhere but the generator bus locations

   % construct sparse angle reduction matrix
   il = length(PQV_no);
   ii = (1:1:il)';
   ang_red = sparse(ii,PQV_no,ones(il,1),il,nbus);

   % construct sparse voltage reduction matrix
   il = length(PQ_no);
   ii = (1:1:il)';
   volt_red = sparse(ii,PQ_no,ones(il,1),il,nbus);

   iter = 0;     % initialize iteration counter

   % calculate the power mismatch and check convergence

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


   st = clock;     % start the iteration time clock
   %% start iteration process for main Newton_Raphson solution
   while (conv_flag == 1 && iter < iter_max)
      iter = iter + 1;
      % Form the Jacobean matrix
```

```
      clear Jac
      Jac=form_jac(V,ang,Y,ang_red,volt_red);
      % reduced real and reactive power mismatch vectors
      red_delP = ang_red*delP;
      red_delQ = volt_red*delQ;
      clear delP delQ
      % solve for voltage magnitude and phase angle increments
      temp = Jac\[red_delP; red_delQ];
      % expand solution vectors to all buses
      delAng = ang_red'*temp(1:length(PQV_no),:);
      delV = volt_red'*temp(length(PQV_no)+1:length(PQV_no)+length(PQ_no),:);
      % update voltage magnitude and phase angle
      V = V + acc*delV;
      V = max(V,volt_min);  % voltage higher than minimum
      V = min(V,volt_max);  % voltage lower than maximum
      ang = ang + acc*delAng;
      % calculate the power mismatch and check convergence
      [delP,delQ,P,Q,conv_flag] =...
         calc(V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol);
      % check if Qg is outside limits
      gen_index=find(bus_type==2);
      Qg(gen_index) = Q(gen_index) + Ql(gen_index);
      lim_flag = chq_lim(qg_max,qg_min);
      if lim_flag == 1;
         disp('Qg at var limit');
      end
   end
    if iter == iter_max
      imstr = int2str(iter_max);
      disp(['inner ac load flow failed to converge after ', imstr,' iterations'])
      tistr = int2str(tap_it);
      disp(['at tap iteration number ' tistr])
   else
      disp('inner load flow iterations')
      disp(iter)
   end
   if no_taps == 0
      lftap
   else
      mm_chk = 0;
   end
end
if tap_it >= tap_it_max
   titstr = int2str(tap_it_max);
   disp(['tap iteration failed to converge after',titstr,' iterations'])
else
   disp(' tap iterations ')
   disp(tap_it)
end
ste = clock;     % end the iteration time clock
vmx_idx = find(V==volt_max);
vmn_idx = find(V==volt_min);
if ~isempty(vmx_idx)
   disp('voltages at')
   bus(vmx_idx,1)'
   disp('are at the max limit')
end
if ~isempty(vmn_idx)
   disp('voltages at')
   bus(vmn_idx,1)'
   disp('are at the min limit');
```

```
end
gen_index=find(bus_type==2);
load_index = find(bus_type==3);
Pg(gen_index) = P(gen_index) + Pl(gen_index);
Qg(gen_index) = Q(gen_index) + Ql(gen_index);
gend_idx = find((bus(:,10)==2)&(bus_type~=2));
if ~isempty(gend_idx)
   disp('the following generators are at their var limits')
   disp('    bus#    Qg')
   disp([bus(gend_idx,1)  Q(gend_idx)])
   Qlg = Ql(gend_idx)-bus(gend_idx,7);% the generator var part of the load
   Qg(gend_idx)=Qg(gend_idx)-Qlg;% restore the generator vars
   Ql(gend_idx)=bus(gend_idx,7);% restore the original load vars
end
Pl(load_index) = Pg(load_index) - P(load_index);
Ql(load_index) = Qg(load_index) - Q(load_index);

Pg(SB) = P(SB) + Pl(SB); Qg(SB) = Q(SB) + Ql(SB);
VV = V.*exp(jay*ang);  % solution voltage
% calculate the line flows and power losses
tap_index = find(abs(line(:,6))>0);
tap_ratio = ones(nline,1);
tap_ratio(tap_index)=line(tap_index,6);
phase_shift(:,1) = line(:,7);
tps = tap_ratio.*exp(jay*phase_shift*pi/180);
from_bus = line(:,1);
from_int = bus_int(round(from_bus));
to_bus = line(:,2);
to_int = bus_int(round(to_bus));
r = line(:,3);
rx = line(:,4);
chrg = line(:,5);
z = r + jay*rx;
y = ones(nline,1)./z;



MW_s = VV(from_int).*conj((VV(from_int) - tps.*VV(to_int)).*y ...
   + VV(from_int).*(jay*chrg/2))./(tps.*conj(tps));
P_s = real(MW_s);     % active power sent out by from_bus
% to to_bus
Q_s = imag(MW_s);     % reactive power sent out by
% from_bus to to_bus
MW_r = VV(to_int).*conj((VV(to_int) ...
   - VV(from_int)./tps).*y ...
   + VV(to_int).*(jay*chrg/2));
P_r = real(MW_r);     % active power received by to_bus
% from from_bus
Q_r = imag(MW_r);     % reactive power received by
% to_bus from from_bus
iline = (1:1:nline)';
line_ffrom = [iline from_bus to_bus P_s Q_s];
line_fto   = [iline to_bus from_bus P_r Q_r];
% keyboard
P_loss = sum(P_s) + sum(P_r) ;
Q_loss = sum(Q_s) + sum(Q_r) ;
bus_sol=[bus_no  V  ang*180/pi Pg Qg Pl Ql Gb Bb...
      bus_type qg_max qg_min bus(:,13) volt_max volt_min];
line_sol = line;
line_flow(1:nline, :)  =[iline from_bus to_bus P_s Q_s];
line_flow(1+nline:2*nline,:) = [iline to_bus from_bus P_r Q_r];
```

```
% Give warning of non-convergence
if conv_flag == 1
   disp('ac load flow failed to converge')
   error('stop')
end

% display results
if display == 'y',
   clc
   disp('                             LOAD-FLOW STUDY')
   disp('                    REPORT OF POWER FLOW CALCULATIONS ')
   disp(' ')
   disp(date)
   fprintf('SWING BUS                  : BUS %g \n', SB)
   fprintf('NUMBER OF ITERATIONS       : %g \n', iter)
   fprintf('SOLUTION TIME              : %g sec.\n',etime(ste,st))
   fprintf('TOTAL TIME                 : %g sec.\n',etime(clock,tt))
   fprintf('TOTAL REAL POWER LOSSES    : %g.\n',P_loss)
   fprintf('TOTAL REACTIVE POWER LOSSES: %g.\n\n',Q_loss)
   if conv_flag == 0,
      disp('                                      GENERATION             LOAD')
      disp('       BUS     VOLTS     ANGLE      REAL   REACTIVE      REAL   REACTIVE ')
      disp(bus_sol(:,1:7))

      disp('                      LINE FLOWS                     ')
      disp('      LINE   FROM BUS    TO BUS      REAL   REACTIVE   ')
      disp(line_ffrom)
      disp(line_fto)
   end
end; %
if iter > iter_max,
   disp('Note: Solution did not converge in %g iterations.\n', iter_max)
   lf_flag = 0
end

return
```

VII. APPENDIX-D: LOADFLOW FUNCTION FILE 'CALC.M'

```
function [delP,delQ,P,Q,conv_flag] = ...
                 calc(V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol)
% Syntax:  [delP,delQ,P,Q,conv_flag] =
%                calc(V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol)
%
% Purpose: calculates power mismatch and checks convergence
%          also determines the values of P and Q based on the
%          supplied values of voltage magnitude and angle
% Version: 2.0 eliminates do loop
% Input:   nbus      - total number of buses
%          bus_type  - load_bus(3), gen_bus(2), swing_bus(1)
%          V         - magnitude of bus voltage
%          ang       - angle(rad) of bus voltage
%          Y         - admittance matrix
%          Pg        - real power of generation
%          Qg        - reactive power  of generation
%          Pl        - real power of load
%          Ql        - reactive power of load
%    sw_bno - a vector having zeros at all  swing_bus locations ones otherwise
```

```
%    g_bno  - a vector having zeros at all  generator bus locations ones otherwise
%          tol       - a tolerance of computational error
%
% Output:  delP      - real power mismatch
%          delQ      - reactive power mismatch
%          P         - calculated real power
%          Q         - calculated reactive power
%          conv_flag - 0, converged
%                      1, not yet converged
%
% See also:
%
% Calls:
%
% Called By:   loadflow

% (c) Copyright 1991 Joe H. Chow - All Rights Reserved
%
% History (in reverse chronological order)
% Version:   2.0
% Author:    Graham Rogers
% Date:      July 1994
%
% Version:   1.0
% Author:    Kwok W. Cheung, Joe H. Chow
% Date:      March 1991
%
% ***********************************************************
jay = sqrt(-1);
swing_bus = 1;
gen_bus = 2;
load_bus = 3;
% voltage in rectangular coordinate
V_rect = V.*exp(jay*ang);
% bus current injection
cur_inj = Y*V_rect;
% power output based on voltages
S = V_rect.*conj(cur_inj);
P = real(S); Q = imag(S);
delP = Pg - Pl - P;
delQ = Qg - Ql - Q;
% zero out mismatches on swing bus and generation bus
delP=delP.*sw_bno;
delQ=delQ.*sw_bno;
delQ=delQ.*g_bno;
%  total mismatch
[pmis,ip]=max(abs(delP));
[qmis,iq]=max(abs(delQ));
mism = pmis+qmis;
if mism > tol,
    conv_flag = 1;
  else
    conv_flag = 0;
end
return
```

VIII. APPENDIX-E: LOADFLOW FUNCTION FILE 'CHQ_LIM.M'

```
function f = chq_lim(qg_max,qg_min)
%Syntax:
%       f = chq_lim(qg_max,qg_min)
% function for detecting generator vars outside limit
% sets Qg to zero if limit exceded, sets Ql to negative of limit
% sets bus_type to 3, and recalculates ang_red and volt_red
% changes  generator bus_type to type 3
% recalculates the generator index
% inputs: qg_max and qg_min are the last two clumns of the bus matrix
% outputs:f is set to zero if no limit reached, or to 1 if a limit is reached
% Version:  1.1
% Author: Graham Rogers
% Date:   May 1997
% Purpose: Addition of var limit index
% Version:  1.0
% Author:   Graham Rogers
% Date:     October 1996
%
% (c) copyright Joe Chow 1996
global Qg bus_type g_bno PQV_no PQ_no ang_red volt_red
global Ql
global gen_chg_idx
%         gen_chg_idx indicates those generators changed to PQ buses
%         gen_cgq_idx = ones(n of bus,1) if no gen at vars limits
%                     = 0 at the corresponding bus if generator at var limit

f = 0;
lim_flag = 0;% indicates whether limit has been reached
gen_idx = find(bus_type ==2);
qg_max_idx = find(Qg(gen_idx)>qg_max(gen_idx));
qg_min_idx = find(Qg(gen_idx)<qg_min(gen_idx));
if ~isempty(qg_max_idx)
  %some q excedes maximum
  %set Qg to zero
  Qg(gen_idx(qg_max_idx)) = zeros(length(qg_max_idx),1);
  % modify Ql
  Ql(gen_idx(qg_max_idx)) = Ql(gen_idx(qg_max_idx))...
                          - qg_max(gen_idx(qg_max_idx));
  % modify bus_type to PQ bus
  bus_type(gen_idx(qg_max_idx)) = 3*ones(length(qg_max_idx),1);
  gen_chg_idx(gen_idx(qg_max_idx)) = zeros(length(qg_max_idx),1);
  lim_flag = 1;
end
if ~isempty(qg_min_idx)
  %some q less than minimum
  %set Qg to zero
  Qg(gen_idx(qg_min_idx)) = zeros(length(qg_min_idx),1);
  % modify Ql
  Ql(gen_idx(qg_min_idx)) = Ql(gen_idx(qg_min_idx))...
                          - qg_min(gen_idx(qg_min_idx));
  % modify bus_type to PQ bus
  bus_type(gen_idx(qg_min_idx)) = 3*ones(length(qg_min_idx),1);
  gen_chg_idx(gen_idx(qg_min_idx)) = zeros(length(qg_min_idx),1);
  lim_flag = 1;
end
if lim_flag == 1
  %recalculate g_bno
  nbus = length(bus_type);
  g_bno = ones(nbus,1);
  bus_zeros=zeros(nbus,1);
  bus_index=(1:1:nbus)';
```

```
  PQV_no=find(bus_type >=2);
  PQ_no=find(bus_type==3);
  gen_index=find(bus_type==2);
  g_bno(gen_index)=bus_zeros(gen_index);
  % construct sparse angle reduction matrix
  il = length(PQV_no);
  ii = (1:1:il)';
  ang_red = sparse(ii,PQV_no,ones(il,1),il,nbus);
  % construct sparse voltage reduction matrix
  il = length(PQ_no);
  ii = (1:1:il)';
  volt_red = sparse(ii,PQ_no,ones(il,1),il,nbus);
end
f = lim_flag;
return
```

IX. APPENDIX-F: LOADFLOW FUNCTION FILE 'FORM_JAC.M'

```
function [Jac11,Jac12,Jac21,Jac22]=form_jac(V,ang,Y,ang_red,volt_red)
% Syntax:  [Jac] = form_jac(V,ang,Y,ang_red,volt_red)
%          [Jac11,Jac12,Jac21,Jac22] = form_jac(V,ang,Y,...
%                                       ang_red,volt_red)
%
% Purpose: form the Jacobian matrix using sparse matrix techniques
%
% Input:   V        - magnitude of bus voltage
%          ang      - angle(rad) of bus voltage
%          Y        - admittance matrix
%          ang_red  - matrix to eliminate swing bus  voltage magnitude and angle
%                     entries
%          volt_red - matrix to eliminate generator bus voltage magnitude
%                     entries
% Output:  Jac      - jacobian matrix
%          Jac11,Jac12,Jac21,Jac22 - submatrices of
%                                       jacobian matrix
% See also:
%
% Calls:
%
% Called By:   vsdemo loadflow

% (c) Copyright 1991-1996 Joe H. Chow - All Rights Reserved
%
% History (in reverse chronological order)
% Version:   2.0
% Author:    Graham Rogers
% Date:      March 1994
% Purpose:   eliminated do loops to improve speed
% Version:   1.0
% Author:    Kwok W. Cheung, Joe H. Chow
% Date:      March 1991
%
% ***********************************************************
jay = sqrt(-1);
exp_ang = exp(jay*ang);
% Voltage rectangular coordinates
V_rect = V.*exp_ang;
CV_rect=conj(V_rect);
```

```
Y_con = conj(Y);
%vector of conjugate currents
i_c=Y_con*CV_rect;
% complex power vector
S=V_rect.*i_c;
S=sparse(diag(S));
Vdia=sparse(diag(V_rect));
CVdia=conj(Vdia);
Vmag=sparse(diag(abs(V)));
S1=Vdia*Y_con*CVdia;
t1=((S+S1)/Vmag)*volt_red';
t2=(S-S1)*ang_red';
J11=-ang_red*imag(t2);
J12=ang_red*real(t1);
J21=volt_red*real(t2);
J22=volt_red*imag(t1);
if nargout > 3
   Jac11 = J11; clear J11
   Jac12 = J12; clear J12
   Jac21 = J21; clear J21
   Jac22 = J22; clear J22
else
   Jac11 = [J11 J12;
            J21 J22];
end
```

X. APPENDIX-G: LOADFLOW FUNCTION FILE 'Y_SPARSE.M'

```
function     [Y,nSW,nPV,nPQ,SB] = y_sparse(bus,line)
% Syntax:    [Y,nSW,nPV,nPQ,SB] = y_sparse(bus,line)
%
% Purpose:   build sparse admittance matrix Y from the line data
%
% Input:     bus  - bus data
%            line - line data
%
% Output:    Y    - admittance matrix
%            nSW  - total number of swing buses
%            nPV  - total number generator buses
%            nPQ  - total number of load buses
%            SB   - internal bus numbers of swing bus
%
% See also:
%
% Calls:
%
% Called By:   loadflow, form_j, calc

% (c) Copyright 1994-1996 Joe Chow - All Rights Reserved
%
% History (in reverse chronological order)
%
% Version:   2.0
% Author:    Graham Rogers
% Date:      April 1994
% Version:   1.0
% Author:    Kwok W. Cheung, Joe H. Chow
% Date:      March 1991
```

```
%
% ************************************************************
global bus_int

jay = sqrt(-1);
swing_bus = 1;
gen_bus = 2;
load_bus = 3;

nline = length(line(:,1));     % number of lines
nbus = length(bus(:,1));     % number of buses
r=zeros(nline,1);
rx=zeros(nline,1);
chrg=zeros(nline,1);
z=zeros(nline,1);
y=zeros(nline,1);

Y = sparse(1,1,0,nbus,nbus);

% set up internal bus numbers for second indexing of buses
busmax = max(bus(:,1));
bus_int = zeros(busmax,1);
ibus = (1:nbus)';
bus_int(round(bus(:,1))) = ibus;


% process line data and build admittance matrix Y
  r = line(:,3);
  rx = line(:,4);
  chrg =jay*sparse(diag( 0.5*line(:,5)));
  z = r + jay*rx;     % line impedance
  y = sparse(diag(ones(nline,1)./z));

% determine connection matrices including tap changers and phase shifters
  from_bus = round(line(:,1));
  from_int = bus_int(from_bus);
  to_bus = round(line(:,2));
  to_int = bus_int(to_bus);
  tap_index = find(abs(line(:,6))>0);
  tap=ones(nline,1);
  tap(tap_index)=1. ./line(tap_index,6);
  phase_shift = line(:,7);
  tap = tap.*exp(-jay*phase_shift*pi/180);

  % sparse matrix formulation
  iline = [1:1:nline]';
  C_from = sparse(from_int,iline,tap,nbus,nline,nline);
  C_to = sparse(to_int,iline,ones(nline,1),nbus,nline,nline);
  C_line = C_from - C_to;

% form Y matrix from primative line ys and connection matrices
  Y=C_from*chrg*C_from' + C_to*chrg*C_to' ;
  Y = Y + C_line*y*C_line';
  Gb = bus(:,8);     % bus conductance
  Bb = bus(:,9);     % bus susceptance

% add diagonal shunt admittances
  Y = Y + sparse(ibus,ibus,Gb+jay*Bb,nbus,nbus);
```

```
if nargout > 1
  % count buses of different types
  nSW = 0;
  nPV = 0;
  nPQ = 0;
  bus_type=round(bus(:,10));
  load_index=find(bus_type==3);
  gen_index=find(bus_type==2);
  SB=find(bus_type==1);
  nSW=length(SB);
  nPV=length(gen_index);
  nPQ=length(load_index);
end

return
```

XI. REFERENCES

[1] http://www.sel.eesc.usp.br/ieee/

[2] B. Pal and B. Chaudhuri, Robust Control in Power Systems. New York, U.S.A.: Springer, 2005, chapter 4.