

เอกสารอ้างอิง

1. Task force on bulk power system reliability of the IEEE-PES application of probability method subcommittee. Bulk power system reliability concepts and applications. IEEE Trans Power Systems., Vol.3, No.1, Feb 1988.
2. Billinton,R., and Allan, R.N. Reliability evaluation of power systems. 1st published :Pitman,1984.
3. Eua-arporn,B. Determination of power system reserves based on reliability indices. Master's Degree thesis ,Chulalongkorn University, 1989.
4. Billinton,R.,Bhavaraju,M.P.,Albrecht,P.F.,Reppen,N.D.,and Ringlee, R.J., Requirements for composite system reliability evaluation models. IEEE Trans. Power Systems.,Vol.3,No.1,Feb 1988.
5. Billinton,R., and Agarwal,S.K. Examination of severe contingencies in a small area of a large composite power system using adequacy equivalent. IEE Proceedings.,Vol.137, Pt.C, No.2, Mar 1990.
6. Billinton,R., and Kumar,S. Adequacy equivalents in composite power system evaluation. IEEE Trans. Power Systems.,Vol.3, No.3, Aug 1988.
7. Billinton,R., Koglin,H.J., and Roos,E. Reliability equivalents in composite system reliability evaluation. IEE Proceedings, Vol.134, Pt.C, No.3, May 1987.
8. Transmission reliability task force* : Pennsylvania - New Jersey - Maryland(PJM). Bulk power area reliability evaluation considering probabilistic transfer capacity. IEEE Trans. Power Apparatus and System.,Vol.PAS-101, No.9, Sep 1982.

9. Billinton,R., and Allan,R.N. Reliability evaluation of engineering system:concept and techniques. reprinted:Pitman Publishing Inc.,1985.
10. IEEE Task Force. IEEE-Reliability Test System. IEEE Trans. Power Apparatus and System., Vol.PAS-98, No.6, Nov/Dec 1979.
11. Billinton,R.,and Kumar,S. A comparative study of system versus load point indices for bulk power systems. IEEE Trans.on Power Systems., Vol.PWRS-1, No.3 ,Aug 1986.
12. Sullivan,R.L, Power system planning. McGraw-Hill Inc.,1977.
13. Kreyszig,E. Advanced engineering mathematics. 6th ed. John Wiley & Son Inc.,1988.
14. Billinton,R., Kumar,S., Chowdhury,N., Chu,K., Debnath,K., Goel,L., Khan,E.,Kos,P.,NourbakhsK,G., and Oteng-Adjei,J. A reliability test system for educational proposes - basic data. IEEE Winter Meeting Paper No.89 WM 035-7 PWRS , Jan 1989.
15. Billinton,R., Kumar,S., Chowdhury,N., Chu,K., Debnath,K., Goel,L., Khan,E.,Kos,P.,NourbakhsK,G., and Oteng-Adjei,J. A reliability test system for educational proposes-basic results. IEEE Trans. Power Systems, Vol.5, No.1, Feb 1990.
16. Billinton,R., and Wenyuan,L. Hybrid approach for reliability evaluation of composit generation and transmission systems using Monte-Carlo simulation and enumeration technique. IEE Proceedings.,Pt.C,Vol.138,No.3, May 1991.
17. Billinton,R., and Salvaderi,L. A comparison between two fundamentally different approaches to composite system reliability evaluation. IEEE Trans. Power Apparatus and system., Vol. PAS-104, No.12, pp 3486-3492, Dec 1985.

ภาคผนวก

ภาคผนวก ก.

โมเดลแอดดิแควอ์คิววาเล็นตส์ ของบัส 11-24 ระบบทดสอบ IEEE-RTS
(Not approximate in available capacity margin)

STATE	PROB	FREQ	Dpt rate. to DOWN	Dpt rate. to UP	Avail Bus 11	Capacity 12	Margin 24	Connected Line in service
1	0.43699267	63.170	7.03	137.53	481.20	481.20	240.60	7,14,15,16,17,
2	0.06896925	21.394	181.89	128.31	402.40	402.40	201.20	7,14,15,16,17,
3	0.04450061	12.721	151.31	134.55	476.40	476.40	238.20	7,14,15,16,17,
4	0.07281670	25.796	225.86	128.40	419.20	419.20	209.60	7,14,15,16,17,
5	0.12148322	24.749	74.23	129.49	321.20	321.20	160.60	7,14,15,16,17,
6	0.02648295	15.311	445.03	133.10	461.20	461.20	230.60	7,14,15,16,17,
7	0.03798133	8.516	94.29	129.91	341.20	341.20	170.60	7,14,15,16,17,
8	0.00362687	1.725	356.42	119.09	323.60	323.60	161.80	7,14,15,16,17,
9	0.00180672	0.767	292.99	131.57	471.60	471.60	235.80	7,14,15,16,17,
10	0.00066282	0.666	876.00	128.68	441.20	441.20	220.60	7,14,15,16,17,
11	0.00451388	2.517	438.36	119.28	357.20	357.20	178.60	7,14,15,16,17,
12	0.00703864	3.190	327.89	125.33	397.60	397.60	198.80	7,14,15,16,17,
13	0.01149731	5.981	401.06	119.18	340.40	340.40	170.20	7,14,15,16,17,
14	0.01248056	5.158	294.34	118.91	259.20	259.20	129.60	7,14,15,16,17,
15	0.04456098	17.804	279.47	120.08	259.20	259.20	129.60	7,14,15,16,17,
16	0.00418150	3.111	620.23	123.88	382.40	382.40	191.20	7,14,15,16,17,
17	0.00599493	2.337	269.22	120.69	262.40	262.40	131.20	7,14,15,16,17,
18	0.00368822	1.812	365.99	125.43	414.40	414.40	207.20	7,14,15,16,17,
19	0.01239046	4.290	219.70	126.51	316.40	316.40	158.20	7,14,15,16,17,
20	0.00269686	1.940	589.31	130.13	456.40	456.40	228.20	7,14,15,16,17,
21	0.00371953	1.853	372.69	125.42	414.40	414.40	207.20	7,14,15,16,17,
22	0.00387618	1.423	240.29	126.93	336.40	336.40	168.20	7,14,15,16,17,
23	0.00438688	3.436	659.18	123.98	399.20	399.20	199.60	7,14,15,16,17,
24	0.00630669	2.719	310.35	120.79	279.20	279.20	139.60	7,14,15,16,17,
25	0.01640546	6.868	300.87	117.77	259.20	259.20	129.60	7,14,15,16,17,
26	0.00728760	4.613	507.83	125.11	301.20	301.20	150.60	7,14,15,16,17,
27	0.00003683	0.021	438.00	128.59	466.80	466.80	233.40	7,14,15,16,17,
28	0.00000893	0.013	1314.00	124.26	421.20	421.20	210.60	7,14,15,16,17,
29	0.00036734	0.225	496.40	116.11	318.80	318.80	159.40	7,14,15,16,17,
30	0.00059991	0.408	569.40	109.96	261.60	261.60	130.80	7,14,15,16,17,
31	0.00021814	0.197	788.40	114.67	303.60	303.60	151.80	7,14,15,16,17,
32	0.00028492	0.168	467.20	122.35	392.80	392.80	196.40	7,14,15,16,17,
33	0.00015037	0.095	511.00	122.45	409.60	409.60	204.80	7,14,15,16,17,
34	0.00050166	0.242	359.26	123.53	311.60	311.60	155.80	7,14,15,16,17,
35	0.00010936	0.094	730.00	127.15	451.60	451.60	225.80	7,14,15,16,17,
36	0.00015037	0.095	511.00	122.45	409.60	409.60	204.80	7,14,15,16,17,
37	0.00015690	0.079	379.60	123.95	331.60	331.60	165.80	7,14,15,16,17,
38	0.00010466	0.123	1051.20	119.46	362.40	362.40	181.20	7,14,15,16,17,
39	0.00006764	0.078	1022.00	125.70	436.40	436.40	218.20	7,14,15,16,17,
40	0.00011047	0.134	1095.00	119.56	379.20	379.20	189.60	7,14,15,16,17,
41	0.00018078	0.191	934.40	120.72	281.20	281.20	140.60	7,14,15,16,17,
42	0.00059366	0.429	613.20	110.06	278.40	278.40	139.20	7,14,15,16,17,
43	0.00038371	0.269	584.00	116.30	352.40	352.40	176.20	7,14,15,16,17,
44	0.00012533	0.096	657.00	110.16	295.20	295.20	147.60	7,14,15,16,17,
45	0.00187361	1.062	451.60	115.03	259.20	259.20	129.60	7,14,15,16,17,
46	0.00027343	0.271	876.00	114.86	337.20	337.20	168.60	7,14,15,16,17,
47	0.00116325	0.764	540.20	116.21	335.60	335.60	167.80	7,14,15,16,17,
48	0.00042298	0.372	759.20	120.91	377.60	377.60	188.80	7,14,15,16,17,
49	0.00011873	0.086	613.20	110.06	278.40	278.40	139.20	7,14,15,16,17,
50	0.00059990	0.350	472.57	111.21	259.20	259.20	129.60	7,14,15,16,17,

51	0.00069077	0.654	832.20	114.76	320.40	320.40	160.20	7,14,15,16,17,
52	0.00137011	0.600	325.52	112.09	203.84	203.84	101.92	7,14,15,16,17,
53	0.00057527	0.457	677.87	115.99	259.20	259.20	129.60	7,14,15,16,17,
54	0.00162188	0.704	321.20	112.73	223.84	223.84	111.92	7,14,15,16,17,
55	0.00007674	0.054	584.00	116.30	352.40	352.40	176.20	7,14,15,16,17,
56	0.00022324	0.206	803.00	121.00	394.40	394.40	197.20	7,14,15,16,17,
57	0.00064060	0.365	452.60	117.81	274.40	274.40	137.20	7,14,15,16,17,
58	0.00073651	0.566	646.13	122.13	296.40	296.40	148.20	7,14,15,16,17,
59	0.00050234	0.272	423.40	117.46	259.20	259.20	129.60	7,14,15,16,17,
60	0.00022324	0.206	803.00	121.00	394.40	394.40	197.20	7,14,15,16,17,
61	0.00072312	0.348	369.32	112.19	220.64	220.64	110.32	7,14,15,16,17,
62	0.00021400	0.102	365.00	112.83	240.64	240.64	120.32	7,14,15,16,17,
63	0.00042800	0.205	365.00	112.83	240.64	240.64	120.32	7,14,15,16,17,
64	0.00021400	0.102	365.00	112.83	240.64	240.64	120.32	7,14,15,16,17,
65	0.00067124	0.301	335.80	112.48	220.64	220.64	110.32	7,14,15,16,17,
66	0.00070039	0.223	204.40	113.99	147.97	147.97	73.98	7,14,15,16,17,
67	0.00085600	0.409	365.00	112.83	240.64	240.64	120.32	7,14,15,16,17,
68	0.00000861	0.006	584.00	101.90	206.24	206.24	103.12	7,14,15,16,17,
69	0.00000861	0.006	584.00	101.90	206.24	206.24	103.12	7,14,15,16,17,
70	0.00000549	0.004	613.20	102.25	226.24	226.24	113.12	7,14,15,16,17,
71	0.00000582	0.004	613.20	119.37	388.00	388.00	194.00	7,14,15,16,17,
72	0.00000614	0.005	657.00	119.47	404.80	404.80	202.40	7,14,15,16,17,
73	0.00001005	0.006	496.40	120.63	306.80	306.80	153.40	7,14,15,16,17,
74	0.00000223	0.002	876.00	124.17	446.80	446.80	223.40	7,14,15,16,17,
75	0.00000320	0.002	525.60	120.97	326.80	326.80	163.40	7,14,15,16,17,
76	0.00000141	0.002	1489.20	115.04	342.40	342.40	171.20	7,14,15,16,17,
77	0.00000243	0.004	1372.40	116.30	261.20	261.20	130.60	7,14,15,16,17,
78	0.00002226	0.023	934.40	111.69	298.80	298.80	149.40	7,14,15,16,17,
79	0.00012271	0.090	627.80	102.00	223.04	223.04	111.52	7,14,15,16,17,
80	0.00002608	0.020	657.00	102.35	243.04	243.04	121.52	7,14,15,16,17,
81	0.00006694	0.038	467.20	103.16	139.74	139.74	69.87	7,14,15,16,17,
82	0.00004090	0.030	627.80	102.00	223.04	223.04	111.52	7,14,15,16,17,
83	0.00008536	0.051	496.40	103.51	149.09	149.09	74.54	7,14,15,16,17,
84	0.00002608	0.020	657.00	102.35	243.04	243.04	121.52	7,14,15,16,17,
85	0.00004749	0.038	686.20	113.23	330.80	330.80	165.40	7,14,15,16,17,
86	0.00001727	0.018	905.20	117.93	372.80	372.80	186.40	7,14,15,16,17,
87	0.00000313	0.003	730.00	113.32	347.60	347.60	173.80	7,14,15,16,17,
88	0.00000911	0.010	949.00	118.02	389.60	389.60	194.80	7,14,15,16,17,
89	0.00001253	0.011	730.00	113.32	347.60	347.60	173.80	7,14,15,16,17,
90	0.00002615	0.019	598.60	114.83	269.60	269.60	134.80	7,14,15,16,17,
91	0.00002983	0.027	788.40	119.18	291.60	291.60	145.80	7,14,15,16,17,
92	0.00000911	0.010	949.00	118.02	389.60	389.60	194.80	7,14,15,16,17,
93	0.00001068	0.014	1197.20	116.48	357.60	357.60	178.80	7,14,15,16,17,
94	0.00001744	0.024	1270.20	110.34	300.40	300.40	150.20	7,14,15,16,17,
95	0.00000564	0.008	1241.00	116.58	374.40	374.40	187.20	7,14,15,16,17,
96	0.00001845	0.022	1080.40	117.74	276.40	276.40	138.20	7,14,15,16,17,
97	0.00000564	0.008	1241.00	116.58	374.40	374.40	187.20	7,14,15,16,17,
98	0.00000575	0.008	1314.00	110.43	317.20	317.20	158.60	7,14,15,16,17,
99	0.00001233	0.014	992.80	112.76	242.64	242.64	121.32	7,14,15,16,17,
100	0.00006059	0.052	759.20	107.08	273.60	273.60	136.80	7,14,15,16,17,

101	0.00016191	0.125	671.60	102.10	239.84	239.84	119.92	7,14,15,16,17,
102	0.00001279	0.012	803.00	107.18	290.40	290.40	145.20	7,14,15,16,17,
103	0.00002790	0.032	1022.00	111.88	332.40	332.40	166.20	7,14,15,16,17,
104	0.00002136	0.017	715.40	102.19	256.64	256.64	128.32	7,14,15,16,17,
105	0.00000760	0.009	1095.00	105.73	275.20	275.20	137.60	7,14,15,16,17,
106	0.00001282	0.010	715.40	102.19	256.64	256.64	128.32	7,14,15,16,17,
107	0.00001398	0.009	554.80	103.35	158.64	158.64	79.32	7,14,15,16,17,
108	0.00001783	0.012	584.00	103.70	178.64	178.64	89.32	7,14,15,16,17,
109	0.00001212	0.010	759.20	107.08	273.60	273.60	136.80	7,14,15,16,17,
110	0.00007050	0.077	978.20	111.78	315.60	315.60	157.80	7,14,15,16,17,
111	0.00012980	0.071	438.00	109.40	199.04	199.04	99.52	7,14,15,16,17,
112	0.00016552	0.095	467.20	109.75	219.04	219.04	109.52	7,14,15,16,17,
113	0.00015898	0.098	511.00	103.26	147.59	147.59	73.80	7,14,15,16,17,
114	0.00023652	0.152	540.20	103.60	161.84	161.84	80.92	7,14,15,16,17,
115	0.00005299	0.033	511.00	103.26	147.59	147.59	73.80	7,14,15,16,17,
116	0.00003238	0.025	671.60	102.10	239.84	239.84	119.92	7,14,15,16,17,
117	0.00003379	0.022	540.20	103.60	161.84	161.84	80.92	7,14,15,16,17,
118	0.00007708	0.065	730.00	107.96	183.84	183.84	91.92	7,14,15,16,17,
119	0.00011059	0.054	379.60	104.77	111.14	111.14	55.57	7,14,15,16,17,
120	0.00013701	0.081	481.80	109.50	215.84	215.84	107.92	7,14,15,16,17,
121	0.00015288	0.095	511.00	109.85	235.84	235.84	117.92	7,14,15,16,17,
122	0.00002184	0.014	511.00	109.85	235.84	235.84	117.92	7,14,15,16,17,
123	0.00004982	0.041	700.80	114.20	257.84	257.84	128.92	7,14,15,16,17,
124	0.00007148	0.033	350.40	111.01	145.72	145.72	72.86	7,14,15,16,17,
125	0.00004195	0.028	554.80	103.35	158.64	158.64	79.32	7,14,15,16,17,
126	0.00000892	0.006	584.00	103.70	178.64	178.64	89.32	7,14,15,16,17,
127	0.00000621	0.006	934.40	106.89	259.20	259.20	129.60	7,14,15,16,17,
128	0.00004458	0.031	584.00	103.70	178.64	178.64	89.32	7,14,15,16,17,
129	0.00004068	0.036	773.80	108.05	200.64	200.64	100.32	7,14,15,16,17,
130	0.00002918	0.015	423.40	104.86	118.99	118.99	59.49	7,14,15,16,17,
131	0.00001297	0.012	803.00	108.40	220.64	220.64	110.32	7,14,15,16,17,
132	0.00001783	0.012	584.00	103.70	178.64	178.64	89.32	7,14,15,16,17,
133	0.00002594	0.024	803.00	108.40	220.64	220.64	110.32	7,14,15,16,17,
134	0.00002797	0.018	554.80	103.35	158.64	158.64	79.32	7,14,15,16,17,
135	0.00002918	0.015	423.40	104.86	118.99	118.99	59.49	7,14,15,16,17,
136	0.00001297	0.012	803.00	108.40	220.64	220.64	110.32	7,14,15,16,17,
137	0.00001783	0.012	584.00	103.70	178.64	178.64	89.32	7,14,15,16,17,
138	0.00000793	0.008	963.60	107.24	259.20	259.20	129.60	7,14,15,16,17,
139	0.00004068	0.036	773.80	108.05	200.64	200.64	100.32	7,14,15,16,17,
140	0.00004245	0.032	642.40	109.56	138.62	138.62	69.31	7,14,15,16,17,
141	0.00005837	0.031	423.40	104.86	118.99	118.99	59.49	7,14,15,16,17,
142	0.00042424	0.396	796.36	137.14	306.00	656.40	240.60	7,14,15,16,17,
143	0.00003435	0.038	971.56	127.91	306.00	498.80	201.20	7,14,15,16,17,
144	0.00004330	0.047	942.36	134.16	306.00	646.80	238.20	7,14,15,16,17,
145	0.00001813	0.021	1015.36	128.01	306.00	532.40	209.60	7,14,15,16,17,
146	0.00011571	0.114	854.76	129.18	306.00	336.40	160.60	7,14,15,16,17,
147	0.00002571	0.035	1234.36	132.72	306.00	616.40	230.60	7,14,15,16,17,
148	0.00003535	0.040	1015.36	128.02	306.00	532.40	209.60	7,14,15,16,17,
149	0.00001892	0.019	883.96	129.51	306.00	376.40	170.60	7,14,15,16,17,
150	0.00000181	0.002	1146.76	118.69	306.00	341.20	161.80	7,14,15,16,17,

151	0.00000351	0.004	1117.56	124.93	306.00	489.20	198.80	7,14,15,16,17,
152	0.00000286	0.004	1190.56	118.79	306.00	374.80	170.20	7,14,15,16,17,
153	0.00000406	0.006	1409.56	123.50	306.00	458.80	191.20	7,14,15,16,17,
154	0.00000558	0.007	1190.56	118.80	306.00	374.80	170.20	7,14,15,16,17,
155	0.00000606	0.007	1000.76	126.19	306.00	326.80	158.20	7,14,15,16,17,
156	0.00000262	0.004	1380.36	129.74	306.00	606.80	228.20	7,14,15,16,17,
157	0.00000185	0.002	1161.36	125.03	306.00	522.80	207.20	7,14,15,16,17,
158	0.00000193	0.002	1029.96	126.53	306.00	366.80	168.20	7,14,15,16,17,
159	0.00000214	0.003	1453.36	123.59	306.00	492.40	199.60	7,14,15,16,17,
160	0.00000115	0.002	1321.96	125.09	306.00	336.40	160.60	7,14,15,16,17,
161	0.00009658	0.104	942.36	134.10	476.40	476.40	238.20	7,14,15,16,17,
162	0.00000590	0.007	1000.76	126.20	316.40	316.40	158.20	7,14,15,16,17,
163	0.00000585	0.009	1380.36	129.68	456.40	456.40	228.20	7,14,15,16,17,
164	0.00000252	0.003	942.36	121.56	259.20	259.20	129.60	7,14,15,16,17,
165	0.00000229	0.003	1088.36	131.07	471.60	471.60	235.80	7,14,15,16,17,
166	0.00003263	0.036	971.56	127.93	306.00	480.00	220.00	7,14,15,16,17,
167	0.00001722	0.020	1015.36	128.03	306.00	522.00	220.00	7,14,15,16,17,
168	0.00001797	0.018	883.96	129.53	291.80	341.20	220.00	7,14,15,16,17,
169	0.00000325	0.004	1146.76	118.73	265.40	323.60	220.00	7,14,15,16,17,
170	0.00000333	0.004	1117.56	124.95	306.00	480.00	208.00	7,14,15,16,17,
171	0.00000272	0.004	1190.56	118.81	290.60	340.40	220.00	7,14,15,16,17,
172	0.00000284	0.003	1059.16	120.31	173.60	262.40	220.00	7,14,15,16,17,
173	0.00000575	0.006	1000.76	126.21	306.00	326.80	158.20	7,14,15,16,17,
174	0.00000176	0.002	1161.36	125.05	306.00	522.00	208.00	7,14,15,16,17,
175	0.00000183	0.002	1029.96	126.55	296.60	336.40	208.00	7,14,15,16,17,
176	0.00000109	0.002	1321.96	125.11	306.00	327.00	170.00	7,14,15,16,17,
177	0.00000150	0.002	1102.96	120.41	198.80	279.20	220.00	7,14,15,16,17,
178	0.00000307	0.004	1146.76	118.75	320.40	323.60	165.00	7,14,15,16,17,
179	0.00000508	0.006	1059.16	120.35	228.60	262.40	165.00	7,14,15,16,17,
180	0.00000268	0.003	1102.96	120.45	279.20	253.80	165.00	7,14,15,16,17,
181	0.00022300	0.208	796.36	137.12	601.50	601.50	0.00	7,14,15,16,17,
182	0.00003521	0.039	971.56	127.90	503.00	503.00	0.00	7,14,15,16,17,
183	0.00002276	0.024	942.36	134.14	595.50	595.50	0.00	7,14,15,16,17,
184	0.00003717	0.042	1015.36	128.00	524.00	524.00	0.00	7,14,15,16,17,
185	0.00006199	0.062	863.62	129.08	401.50	401.50	0.00	7,14,15,16,17,
186	0.00001351	0.018	1234.36	132.70	576.50	576.50	0.00	7,14,15,16,17,
187	0.00001939	0.020	883.96	129.50	426.50	426.50	0.00	7,14,15,16,17,
188	0.00000185	0.002	1146.76	118.68	404.50	404.50	0.00	7,14,15,16,17,
189	0.00000359	0.004	1117.56	124.92	497.00	497.00	0.00	7,14,15,16,17,
190	0.00000293	0.004	1190.56	118.78	425.50	425.50	0.00	7,14,15,16,17,
191	0.00002503	0.029	1029.19	120.31	324.00	324.00	0.00	7,14,15,16,17,
192	0.00000213	0.003	1409.56	123.48	478.00	478.00	0.00	7,14,15,16,17,
193	0.00000293	0.004	1190.56	118.78	425.50	425.50	0.00	7,14,15,16,17,
194	0.00000306	0.004	1059.16	120.28	328.00	328.00	0.00	7,14,15,16,17,
195	0.00000621	0.007	1000.76	126.18	395.50	395.50	0.00	7,14,15,16,17,
196	0.00000138	0.002	1380.36	129.72	570.50	570.50	0.00	7,14,15,16,17,
197	0.00000190	0.002	1161.36	125.02	518.00	518.00	0.00	7,14,15,16,17,
198	0.00000198	0.002	1029.96	126.52	420.50	420.50	0.00	7,14,15,16,17,
199	0.00000415	0.004	913.16	121.19	324.00	324.00	0.00	7,14,15,16,17,
200	0.00000369	0.005	1292.76	124.73	376.50	376.50	0.00	7,14,15,16,17,

201	0.00000113	0.002	1453.36	123.57	499.00	499.00	0.00	7,14,15,16,17,
202	0.00000162	0.002	1102.96	120.38	349.00	349.00	0.00	7,14,15,16,17,
203	0.00002920	0.032	971.56	127.97	402.40	383.60	220.00	7,14,15,16,17,
204	0.00001541	0.018	1015.36	128.07	419.20	408.80	220.00	7,14,15,16,17,
205	0.00001608	0.016	883.96	129.57	341.20	291.80	220.00	7,14,15,16,17,
206	0.00000298	0.004	1117.56	124.99	397.60	388.40	208.00	7,14,15,16,17,
207	0.00000243	0.003	1190.56	118.85	340.40	290.60	220.00	7,14,15,16,17,
208	0.00000254	0.003	1059.16	120.35	262.40	173.60	220.00	7,14,15,16,17,
209	0.00000157	0.002	1161.36	125.09	414.40	413.60	208.00	7,14,15,16,17,
210	0.00000164	0.002	1029.96	126.59	336.40	296.60	208.00	7,14,15,16,17,
211	0.00000134	0.002	1102.96	120.45	279.20	198.80	220.00	7,14,15,16,17,
212	0.00000890	0.010	1029.96	119.97	259.20	259.20	129.60	7,14,15,16,17,
213	0.00000490	0.005	942.36	121.57	259.20	259.20	129.60	7,14,15,16,17,
214	0.00075949	0.113	11.41	137.51	601.50	601.50	0.00	14,15,16,17,
215	0.00011992	0.038	186.61	128.29	503.00	503.00	0.00	14,15,16,17,
216	0.00007751	0.023	157.41	134.53	595.50	595.50	0.00	14,15,16,17,
217	0.00012658	0.045	230.41	128.39	524.00	524.00	0.00	14,15,16,17,
218	0.00021114	0.044	78.66	129.47	401.50	401.50	0.00	14,15,16,17,
219	0.00004603	0.027	449.41	133.09	576.50	576.50	0.00	14,15,16,17,
220	0.00006604	0.015	99.01	129.89	426.50	426.50	0.00	14,15,16,17,
221	0.00000631	0.003	361.81	119.07	404.50	404.50	0.00	14,15,16,17,
222	0.00000316	0.001	303.41	131.55	589.50	589.50	0.00	14,15,16,17,
223	0.00000116	0.001	887.41	128.66	551.50	551.50	0.00	14,15,16,17,
224	0.00000791	0.004	449.41	119.26	446.50	446.50	0.00	14,15,16,17,
225	0.00001224	0.006	332.61	125.31	497.00	497.00	0.00	14,15,16,17,
226	0.00001999	0.010	405.61	119.17	425.50	425.50	0.00	14,15,16,17,
227	0.00008691	0.032	247.05	120.64	324.00	324.00	0.00	14,15,16,17,
228	0.00000727	0.005	624.61	123.87	478.00	478.00	0.00	14,15,16,17,
229	0.00001043	0.004	274.21	120.67	328.00	328.00	0.00	14,15,16,17,
230	0.00001292	0.006	376.41	125.41	518.00	518.00	0.00	14,15,16,17,
231	0.00002114	0.007	215.81	126.57	395.50	395.50	0.00	14,15,16,17,
232	0.00000470	0.003	595.41	130.11	570.50	570.50	0.00	14,15,16,17,
233	0.00000674	0.003	245.01	126.91	420.50	420.50	0.00	14,15,16,17,
234	0.00000767	0.006	668.41	123.96	499.00	499.00	0.00	14,15,16,17,
235	0.00001101	0.005	318.01	120.77	349.00	349.00	0.00	14,15,16,17,
236	0.00002102	0.007	217.39	119.50	324.00	324.00	0.00	14,15,16,17,
237	0.00001255	0.008	507.81	125.12	376.50	376.50	0.00	14,15,16,17,
238	0.00000102	0.001	551.61	116.19	419.50	419.50	0.00	14,15,16,17,
239	0.00000223	0.001	303.41	112.36	254.80	254.80	0.00	14,15,16,17,
240	0.00000284	0.001	332.61	112.71	279.80	279.80	0.00	14,15,16,17,
241	0.00000118	0.001	347.21	112.46	275.80	275.80	0.00	14,15,16,17,
242	0.00000123	0.000	215.81	113.97	184.96	184.96	0.00	14,15,16,17,
243	0.00075949	0.113	11.41	137.51	300.75	601.50	300.75	7,15,16,17,
244	0.00011992	0.038	186.61	128.29	251.50	503.00	251.50	7,15,16,17,
245	0.00007751	0.023	157.41	134.53	297.75	595.50	297.75	7,15,16,17,
246	0.00012658	0.045	230.41	128.39	262.00	524.00	262.00	7,15,16,17,
247	0.00021114	0.044	78.66	129.47	200.75	401.50	200.75	7,15,16,17,
248	0.00004603	0.027	449.41	133.09	288.25	576.50	288.25	7,15,16,17,
249	0.00006604	0.015	99.01	129.89	213.25	426.50	213.25	7,15,16,17,
250	0.00000631	0.003	361.81	119.07	202.25	404.50	202.25	7,15,16,17,

251	0.00000316	0.001	303.41	131.55	294.75	589.50	294.75	7,15,16,17,
252	0.00000116	0.001	887.41	128.66	275.75	551.50	275.75	7,15,16,17,
253	0.00000791	0.004	449.41	119.26	223.25	446.50	223.25	7,15,16,17,
254	0.00001224	0.006	332.61	125.31	248.50	497.00	248.50	7,15,16,17,
255	0.00001999	0.010	405.61	119.17	212.75	425.50	212.75	7,15,16,17,
256	0.00008691	0.032	247.05	120.64	162.00	324.00	162.00	7,15,16,17,
257	0.00000727	0.005	624.61	123.87	239.00	478.00	239.00	7,15,16,17,
258	0.00001043	0.004	274.21	120.67	164.00	328.00	164.00	7,15,16,17,
259	0.00001292	0.006	376.41	125.41	259.00	518.00	259.00	7,15,16,17,
260	0.00002114	0.007	215.81	126.57	197.75	395.50	197.75	7,15,16,17,
261	0.00000470	0.003	595.41	130.11	285.25	570.50	285.25	7,15,16,17,
262	0.00000674	0.003	245.01	126.91	210.25	420.50	210.25	7,15,16,17,
263	0.00000767	0.006	668.41	123.96	249.50	499.00	249.50	7,15,16,17,
264	0.00001101	0.005	318.01	120.77	174.50	349.00	174.50	7,15,16,17,
265	0.00002102	0.007	217.39	119.50	162.00	324.00	162.00	7,15,16,17,
266	0.00001255	0.008	507.81	125.12	188.25	376.50	188.25	7,15,16,17,
267	0.00000102	0.001	551.61	116.19	209.75	419.50	209.75	7,15,16,17,
268	0.00000223	0.001	303.41	112.36	127.40	254.80	127.40	7,15,16,17,
269	0.00000284	0.001	332.61	112.71	139.90	279.80	139.90	7,15,16,17,
270	0.00000118	0.001	347.21	112.46	137.90	275.80	137.90	7,15,16,17,
271	0.00000123	0.000	215.81	113.97	92.48	184.96	92.48	7,15,16,17,
272	0.00075949	0.113	11.41	137.51	601.50	300.75	300.75	7,14,16,17,
273	0.00011992	0.038	186.61	128.29	503.00	251.50	251.50	7,14,16,17,
274	0.00007751	0.023	157.41	134.53	595.50	297.75	297.75	7,14,16,17,
275	0.00012658	0.045	230.41	128.39	524.00	262.00	262.00	7,14,16,17,
276	0.00021114	0.044	78.66	129.47	401.50	200.75	200.75	7,14,16,17,
277	0.00004603	0.027	449.41	133.09	576.50	288.25	288.25	7,14,16,17,
278	0.00006604	0.015	99.01	129.89	426.50	213.25	213.25	7,14,16,17,
279	0.00000631	0.003	361.81	119.07	404.50	202.25	202.25	7,14,16,17,
280	0.00000316	0.001	303.41	131.55	589.50	294.75	294.75	7,14,16,17,
281	0.00000116	0.001	887.41	128.66	551.50	275.75	275.75	7,14,16,17,
282	0.00000791	0.004	449.41	119.26	446.50	223.25	223.25	7,14,16,17,
283	0.00001224	0.006	332.61	125.31	497.00	248.50	248.50	7,14,16,17,
284	0.00001999	0.010	405.61	119.17	425.50	212.75	212.75	7,14,16,17,
285	0.00001635	0.006	245.01	120.33	324.00	162.00	162.00	7,14,16,17,
286	0.00007056	0.026	247.53	120.71	324.00	162.00	162.00	7,14,16,17,
287	0.00000727	0.005	624.61	123.87	478.00	239.00	239.00	7,14,16,17,
288	0.00001043	0.004	274.21	120.67	328.00	164.00	164.00	7,14,16,17,
289	0.00001292	0.006	376.41	125.41	518.00	259.00	259.00	7,14,16,17,
290	0.00002114	0.007	215.81	126.57	395.50	197.75	197.75	7,14,16,17,
291	0.00000470	0.003	595.41	130.11	570.50	285.25	285.25	7,14,16,17,
292	0.00000674	0.003	245.01	126.91	420.50	210.25	210.25	7,14,16,17,
293	0.00000767	0.006	668.41	123.96	499.00	249.50	249.50	7,14,16,17,
294	0.00001101	0.005	318.01	120.77	349.00	174.50	174.50	7,14,16,17,
295	0.00002102	0.007	217.39	119.50	324.00	162.00	162.00	7,14,16,17,
296	0.00001255	0.008	507.81	125.12	376.50	188.25	188.25	7,14,16,17,
297	0.00000102	0.001	551.61	116.19	419.50	209.75	209.75	7,14,16,17,
298	0.00000223	0.001	303.41	112.36	254.80	127.40	127.40	7,14,16,17,
299	0.00000284	0.001	332.61	112.71	279.80	139.90	139.90	7,14,16,17,
300	0.00000118	0.001	347.21	112.46	275.80	137.90	137.90	7,14,16,17,

301	0.00000123	0.000	215.81	113.97	184.96	92.48	92.48	7,14,16,17,
302	0.00075949	0.113	11.41	137.51	300.75	601.50	300.75	7,14,15,17,
303	0.00011992	0.038	186.61	128.29	251.50	503.00	251.50	7,14,15,17,
304	0.00007751	0.023	157.41	134.53	297.75	595.50	297.75	7,14,15,17,
305	0.00012658	0.045	230.41	128.39	262.00	524.00	262.00	7,14,15,17,
306	0.00021114	0.044	78.66	129.47	200.75	401.50	200.75	7,14,15,17,
307	0.00004603	0.027	449.41	133.09	288.25	576.50	288.25	7,14,15,17,
308	0.00006604	0.015	99.01	129.89	213.25	426.50	213.25	7,14,15,17,
309	0.00000631	0.003	361.81	119.07	202.25	404.50	202.25	7,14,15,17,
310	0.00000316	0.001	303.41	131.55	294.75	589.50	294.75	7,14,15,17,
311	0.00000116	0.001	887.41	128.66	275.75	551.50	275.75	7,14,15,17,
312	0.00000791	0.004	449.41	119.26	223.25	446.50	223.25	7,14,15,17,
313	0.00001224	0.006	332.61	125.31	248.50	497.00	248.50	7,14,15,17,
314	0.00001999	0.010	405.61	119.17	212.75	425.50	212.75	7,14,15,17,
315	0.00008691	0.032	247.05	120.64	162.00	324.00	162.00	7,14,15,17,
316	0.00000727	0.005	624.61	123.87	239.00	478.00	239.00	7,14,15,17,
317	0.00001043	0.004	274.21	120.67	164.00	328.00	164.00	7,14,15,17,
318	0.00001292	0.006	376.41	125.41	259.00	518.00	259.00	7,14,15,17,
319	0.00002114	0.007	215.81	126.57	197.75	395.50	197.75	7,14,15,17,
320	0.00000470	0.003	595.41	130.11	285.25	570.50	285.25	7,14,15,17,
321	0.00000674	0.003	245.01	126.91	210.25	420.50	210.25	7,14,15,17,
322	0.00000767	0.006	668.41	123.96	249.50	499.00	249.50	7,14,15,17,
323	0.00001101	0.005	318.01	120.77	174.50	349.00	174.50	7,14,15,17,
324	0.00002102	0.007	217.39	119.50	162.00	324.00	162.00	7,14,15,17,
325	0.00001255	0.008	507.81	125.12	188.25	376.50	188.25	7,14,15,17,
326	0.00000102	0.001	551.61	116.19	209.75	419.50	209.75	7,14,15,17,
327	0.00000223	0.001	303.41	112.36	127.40	254.80	127.40	7,14,15,17,
328	0.00000284	0.001	332.61	112.71	139.90	279.80	139.90	7,14,15,17,
329	0.00000118	0.001	347.21	112.46	137.90	275.80	137.90	7,14,15,17,
330	0.00000123	0.000	215.81	113.97	92.48	184.96	92.48	7,14,15,17,
331	0.00075949	0.113	11.41	137.51	601.50	300.75	300.75	7,14,15,16,
332	0.00011992	0.038	186.61	128.29	503.00	251.50	251.50	7,14,15,16,
333	0.00007751	0.023	157.41	134.53	595.50	297.75	297.75	7,14,15,16,
334	0.00012658	0.045	230.41	128.39	524.00	262.00	262.00	7,14,15,16,
335	0.00021114	0.044	78.66	129.47	401.50	200.75	200.75	7,14,15,16,
336	0.00004603	0.027	449.41	133.09	576.50	288.25	288.25	7,14,15,16,
337	0.00006604	0.015	99.01	129.89	426.50	213.25	213.25	7,14,15,16,
338	0.00000631	0.003	361.81	119.07	404.50	202.25	202.25	7,14,15,16,
339	0.00000316	0.001	303.41	131.55	589.50	294.75	294.75	7,14,15,16,
340	0.00000116	0.001	887.41	128.66	551.50	275.75	275.75	7,14,15,16,
341	0.00000791	0.004	449.41	119.26	446.50	223.25	223.25	7,14,15,16,
342	0.00001224	0.006	332.61	125.31	497.00	248.50	248.50	7,14,15,16,
343	0.00001999	0.010	405.61	119.17	425.50	212.75	212.75	7,14,15,16,
344	0.00001635	0.006	245.01	120.33	324.00	162.00	162.00	7,14,15,16,
345	0.00007056	0.026	247.53	120.71	324.00	162.00	162.00	7,14,15,16,
346	0.00000727	0.005	624.61	123.87	478.00	239.00	239.00	7,14,15,16,
347	0.00001043	0.004	274.21	120.67	328.00	164.00	164.00	7,14,15,16,
348	0.00001292	0.006	376.41	125.41	518.00	259.00	259.00	7,14,15,16,
349	0.00002114	0.007	215.81	126.57	395.50	197.75	197.75	7,14,15,16,
350	0.00000470	0.003	595.41	130.11	570.50	285.25	285.25	7,14,15,16,

351	0.00000674	0.003	245.01	126.91	420.50	210.25	210.25	7,14,15,16,
352	0.00000767	0.006	668.41	123.96	499.00	249.50	249.50	7,14,15,16,
353	0.00001101	0.005	318.01	120.77	349.00	174.50	174.50	7,14,15,16,
354	0.00002102	0.007	217.39	119.50	324.00	162.00	162.00	7,14,15,16,
355	0.00001255	0.008	507.81	125.12	376.50	188.25	188.25	7,14,15,16,
356	0.00000102	0.001	551.61	116.19	419.50	209.75	209.75	7,14,15,16,
357	0.00000223	0.001	303.41	112.36	254.80	127.40	127.40	7,14,15,16,
358	0.00000284	0.001	332.61	112.71	279.80	139.90	139.90	7,14,15,16,
359	0.00000118	0.001	347.21	112.46	275.80	137.90	137.90	7,14,15,16,
360	0.00000123	0.000	215.81	113.97	184.96	92.48	92.48	7,14,15,16,
361	0.00000133	0.000	22.81	137.49	400.00	800.00	0.00	15,16,17,
362	0.00000133	0.000	22.81	137.49	800.00	400.00	0.00	14,16,17,
363	0.00000133	0.000	22.81	137.49	400.00	800.00	0.00	14,15,17,
364	0.00000133	0.000	22.81	137.49	800.00	400.00	0.00	14,15,16,
365	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,16,17,
366	0.00000133	0.000	22.81	137.49	0.00	800.00	400.00	7,15,17,
367	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,15,16,
368	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,14,17,
369	0.00000133	0.000	22.81	137.49	800.00	0.00	400.00	7,14,16,
370	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,14,15,

PROBABILITY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = 0.99829346
FREQUENCY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = 261.3188
TOTAL STATES OF MODEL ADEQUACY EQUIVALENTS = 370
TIME TO DEVELOP MODEL ADEQUACY EQUIVALENTS = 253 Sec

ภาคผนวก ข.

โมเดลแอตติเคาซ็อกวิวาเลนซ์ ของบัส 11-24 ระบบทดสอบ IEEE-RTS

เมื่อประมาณค่าความพร้อมมูลในการจ่ายกำลังไฟฟ้าเป็นจำนวนเต็ม

STATE	PROB	FREQ	Dpt rate. to DOWN	Dpt rate. to UP	Avail Bus 11	Capacity 12	Margin 24	Connected Line in service
1	0.43699267	63.170	7.03	137.53	481.00	481.00	241.00	7,14,15,16,17,
2	0.06896925	21.394	181.89	128.31	402.00	403.00	201.00	7,14,15,16,17,
3	0.04459720	12.825	153.03	134.55	476.00	477.00	238.00	7,14,15,16,17,
4	0.07281670	25.796	225.86	128.40	419.00	419.00	210.00	7,14,15,16,17,
5	0.12148322	24.749	74.23	129.49	321.00	321.00	161.00	7,14,15,16,17,
6	0.02648295	15.311	445.03	133.10	461.00	461.00	231.00	7,14,15,16,17,
7	0.03798133	8.516	94.29	129.91	341.00	341.00	171.00	7,14,15,16,17,
8	0.00362687	1.725	356.42	119.09	323.00	324.00	162.00	7,14,15,16,17,
9	0.00180901	0.770	293.99	131.57	471.00	472.00	236.00	7,14,15,16,17,
10	0.00066282	0.666	876.00	128.68	441.00	441.00	221.00	7,14,15,16,17,
11	0.00451388	2.517	438.36	119.28	357.00	357.00	179.00	7,14,15,16,17,
12	0.00703864	3.190	327.89	125.33	397.00	398.00	199.00	7,14,15,16,17,
13	0.01149731	5.981	401.06	119.18	340.00	341.00	170.00	7,14,15,16,17,
14	0.07702857	32.003	296.31	119.16	259.00	259.00	130.00	7,14,15,16,17,
15	0.00418150	3.111	620.23	123.88	382.00	383.00	191.00	7,14,15,16,17,
16	0.00599493	2.337	269.22	120.69	262.00	263.00	131.00	7,14,15,16,17,
17	0.00740775	3.665	369.35	125.42	414.00	415.00	207.00	7,14,15,16,17,
18	0.01239636	4.296	220.07	126.51	316.00	317.00	158.00	7,14,15,16,17,
19	0.00270271	1.949	591.03	130.12	456.00	457.00	228.00	7,14,15,16,17,
20	0.00387618	1.423	240.29	126.93	336.00	337.00	168.00	7,14,15,16,17,
21	0.00438688	3.436	659.18	123.98	399.00	399.00	200.00	7,14,15,16,17,
22	0.00630669	2.719	310.35	120.79	279.00	279.00	140.00	7,14,15,16,17,
23	0.00728760	4.613	507.83	125.11	301.00	301.00	151.00	7,14,15,16,17,
24	0.00003683	0.021	438.00	128.59	467.00	467.00	233.00	7,14,15,16,17,
25	0.00000893	0.013	1314.00	124.26	421.00	421.00	211.00	7,14,15,16,17,
26	0.00036734	0.225	496.40	116.11	319.00	319.00	159.00	7,14,15,16,17,
27	0.00059991	0.408	569.40	109.96	261.00	262.00	131.00	7,14,15,16,17,
28	0.00021814	0.197	788.40	114.67	303.00	304.00	152.00	7,14,15,16,17,
29	0.00028492	0.168	467.20	122.35	393.00	393.00	196.00	7,14,15,16,17,
30	0.00030075	0.191	511.00	122.45	409.00	410.00	205.00	7,14,15,16,17,
31	0.00050166	0.242	359.26	123.53	311.00	312.00	156.00	7,14,15,16,17,
32	0.00010936	0.094	730.00	127.15	451.00	452.00	226.00	7,14,15,16,17,
33	0.00015690	0.079	379.60	123.95	331.00	332.00	166.00	7,14,15,16,17,
34	0.00010466	0.123	1051.20	119.46	362.00	363.00	181.00	7,14,15,16,17,
35	0.00006764	0.078	1022.00	125.70	436.00	437.00	218.00	7,14,15,16,17,
36	0.00011047	0.134	1095.00	119.56	379.00	379.00	190.00	7,14,15,16,17,
37	0.00018078	0.191	934.40	120.72	281.00	281.00	141.00	7,14,15,16,17,
38	0.00071239	0.515	613.20	110.06	278.00	279.00	139.00	7,14,15,16,17,
39	0.00046045	0.322	584.00	116.30	352.00	353.00	176.00	7,14,15,16,17,
40	0.00012533	0.096	657.00	110.16	295.00	295.00	148.00	7,14,15,16,17,
41	0.00027343	0.271	876.00	114.86	337.00	337.00	169.00	7,14,15,16,17,
42	0.00116325	0.764	540.20	116.21	335.00	336.00	168.00	7,14,15,16,17,
43	0.00042298	0.372	759.20	120.91	377.00	378.00	189.00	7,14,15,16,17,
44	0.00069077	0.654	832.20	114.76	320.00	321.00	160.00	7,14,15,16,17,
45	0.00137011	0.600	325.52	112.09	204.00	204.00	102.00	7,14,15,16,17,
46	0.00162188	0.704	321.20	112.73	224.00	224.00	112.00	7,14,15,16,17,
47	0.00044648	0.413	803.00	121.00	394.00	395.00	197.00	7,14,15,16,17,
48	0.00064060	0.365	452.60	117.81	274.00	275.00	137.00	7,14,15,16,17,
49	0.00073651	0.566	646.13	122.13	296.00	297.00	148.00	7,14,15,16,17,
50	0.00144624	0.696	369.32	112.19	221.00	221.00	110.00	7,14,15,16,17,

51	0.00171199	0.818	365.00	112.83	241.00	241.00	120.00	7,14,15,16,17,
52	0.00070039	0.223	204.40	113.99	148.00	148.00	74.00	7,14,15,16,17,
53	0.00001722	0.012	584.00	101.90	206.00	207.00	103.00	7,14,15,16,17,
54	0.00000549	0.004	613.20	102.25	226.00	227.00	113.00	7,14,15,16,17,
55	0.00000582	0.004	613.20	119.37	388.00	388.00	194.00	7,14,15,16,17,
56	0.00000614	0.005	657.00	119.47	405.00	405.00	202.00	7,14,15,16,17,
57	0.00001005	0.006	496.40	120.63	307.00	307.00	153.00	7,14,15,16,17,
58	0.00000223	0.002	876.00	124.17	447.00	447.00	223.00	7,14,15,16,17,
59	0.00000320	0.002	525.60	120.97	327.00	327.00	163.00	7,14,15,16,17,
60	0.00000141	0.002	1489.20	115.04	342.00	343.00	171.00	7,14,15,16,17,
61	0.00000243	0.004	1372.40	116.30	261.00	261.00	131.00	7,14,15,16,17,
62	0.00002226	0.023	934.40	111.69	299.00	299.00	149.00	7,14,15,16,17,
63	0.00016362	0.119	627.80	102.00	223.00	223.00	112.00	7,14,15,16,17,
64	0.00005216	0.040	657.00	102.35	243.00	243.00	122.00	7,14,15,16,17,
65	0.00006694	0.038	467.20	103.16	139.00	140.00	70.00	7,14,15,16,17,
66	0.00008536	0.051	496.40	103.51	149.00	149.00	75.00	7,14,15,16,17,
67	0.00004749	0.038	686.20	113.23	331.00	331.00	165.00	7,14,15,16,17,
68	0.00001727	0.018	905.20	117.93	373.00	373.00	186.00	7,14,15,16,17,
69	0.00001566	0.013	730.00	113.32	347.00	348.00	174.00	7,14,15,16,17,
70	0.00001823	0.019	949.00	118.02	389.00	390.00	195.00	7,14,15,16,17,
71	0.00002615	0.019	598.60	114.83	269.00	270.00	135.00	7,14,15,16,17,
72	0.00002983	0.027	788.40	119.18	291.00	292.00	146.00	7,14,15,16,17,
73	0.00001068	0.014	1197.20	116.48	357.00	358.00	179.00	7,14,15,16,17,
74	0.00001744	0.024	1270.20	110.34	300.00	301.00	150.00	7,14,15,16,17,
75	0.00001127	0.015	1241.00	116.58	374.00	375.00	187.00	7,14,15,16,17,
76	0.00001845	0.022	1080.40	117.74	276.00	277.00	138.00	7,14,15,16,17,
77	0.00000575	0.008	1314.00	110.43	317.00	317.00	159.00	7,14,15,16,17,
78	0.00001233	0.014	992.80	112.76	243.00	243.00	121.00	7,14,15,16,17,
79	0.00007270	0.063	759.20	107.08	273.00	274.00	137.00	7,14,15,16,17,
80	0.00019430	0.150	671.60	102.10	240.00	240.00	120.00	7,14,15,16,17,
81	0.00001279	0.012	803.00	107.18	290.00	291.00	145.00	7,14,15,16,17,
82	0.00002790	0.032	1022.00	111.88	332.00	333.00	166.00	7,14,15,16,17,
83	0.00003418	0.028	715.40	102.19	257.00	257.00	128.00	7,14,15,16,17,
84	0.00000760	0.009	1095.00	105.73	275.00	275.00	138.00	7,14,15,16,17,
85	0.00008391	0.055	554.80	103.35	159.00	159.00	79.00	7,14,15,16,17,
86	0.00010700	0.074	584.00	103.70	179.00	179.00	89.00	7,14,15,16,17,
87	0.00007050	0.077	978.20	111.78	315.00	316.00	158.00	7,14,15,16,17,
88	0.00012980	0.071	438.00	109.40	199.00	199.00	100.00	7,14,15,16,17,
89	0.00016552	0.095	467.20	109.75	219.00	219.00	110.00	7,14,15,16,17,
90	0.00021197	0.130	511.00	103.26	147.00	148.00	74.00	7,14,15,16,17,
91	0.00027031	0.174	540.20	103.60	162.00	162.00	81.00	7,14,15,16,17,
92	0.00007708	0.065	730.00	107.96	184.00	184.00	92.00	7,14,15,16,17,
93	0.00011059	0.054	379.60	104.77	111.00	111.00	56.00	7,14,15,16,17,
94	0.00013701	0.081	481.80	109.50	216.00	216.00	108.00	7,14,15,16,17,
95	0.00017472	0.108	511.00	109.85	236.00	236.00	118.00	7,14,15,16,17,
96	0.00004982	0.041	700.80	114.20	258.00	258.00	129.00	7,14,15,16,17,
97	0.00007148	0.033	350.40	111.01	145.00	146.00	73.00	7,14,15,16,17,
98	0.00008136	0.072	773.80	108.05	201.00	201.00	100.00	7,14,15,16,17,
99	0.00011673	0.062	423.40	104.86	119.00	119.00	59.00	7,14,15,16,17,
100	0.00004245	0.032	642.40	109.56	139.00	139.00	69.00	7,14,15,16,17,

101	0.00042424	0.396	796.36	137.14	306.00	656.00	241.00	7,14,15,16,17,
102	0.00003435	0.038	971.56	127.91	306.00	499.00	201.00	7,14,15,16,17,
103	0.00004330	0.047	942.36	134.16	306.00	647.00	238.00	7,14,15,16,17,
104	0.00005348	0.061	1015.36	128.01	306.00	532.00	210.00	7,14,15,16,17,
105	0.00011685	0.116	859.35	129.14	306.00	336.00	161.00	7,14,15,16,17,
106	0.00002571	0.035	1234.36	132.72	306.00	616.00	231.00	7,14,15,16,17,
107	0.00001892	0.019	883.96	129.51	306.00	376.00	171.00	7,14,15,16,17,
108	0.00000181	0.002	1146.76	118.69	306.00	341.00	162.00	7,14,15,16,17,
109	0.00000351	0.004	1117.56	124.93	306.00	489.00	199.00	7,14,15,16,17,
110	0.00000844	0.011	1190.56	118.79	306.00	375.00	170.00	7,14,15,16,17,
111	0.00000406	0.006	1409.56	123.50	306.00	459.00	191.00	7,14,15,16,17,
112	0.00001181	0.013	1000.76	126.20	306.00	327.00	158.00	7,14,15,16,17,
113	0.00000262	0.004	1380.36	129.74	306.00	607.00	228.00	7,14,15,16,17,
114	0.00000185	0.002	1161.36	125.03	306.00	523.00	207.00	7,14,15,16,17,
115	0.00000193	0.002	1029.96	126.53	306.00	367.00	168.00	7,14,15,16,17,
116	0.00000214	0.003	1453.36	123.59	306.00	492.00	200.00	7,14,15,16,17,
117	0.00003263	0.036	971.56	127.93	306.00	480.00	220.00	7,14,15,16,17,
118	0.00001722	0.020	1015.36	128.03	306.00	522.00	220.00	7,14,15,16,17,
119	0.00001797	0.018	883.96	129.53	292.00	341.00	220.00	7,14,15,16,17,
120	0.00000325	0.004	1146.76	118.73	265.00	324.00	220.00	7,14,15,16,17,
121	0.00000333	0.004	1117.56	124.95	306.00	480.00	208.00	7,14,15,16,17,
122	0.00000272	0.004	1190.56	118.81	291.00	340.00	220.00	7,14,15,16,17,
123	0.00000284	0.003	1059.16	120.31	174.00	262.00	220.00	7,14,15,16,17,
124	0.00000176	0.002	1161.36	125.05	306.00	522.00	208.00	7,14,15,16,17,
125	0.00000183	0.002	1029.96	126.55	297.00	336.00	208.00	7,14,15,16,17,
126	0.00000109	0.002	1321.96	125.11	306.00	327.00	170.00	7,14,15,16,17,
127	0.00000150	0.002	1102.96	120.41	199.00	279.00	220.00	7,14,15,16,17,
128	0.00000307	0.004	1146.76	118.75	320.00	324.00	165.00	7,14,15,16,17,
129	0.00000508	0.006	1059.16	120.35	229.00	262.00	165.00	7,14,15,16,17,
130	0.00000268	0.003	1102.96	120.45	279.00	254.00	165.00	7,14,15,16,17,
131	0.00022300	0.208	796.36	137.12	601.00	602.00	0.00	7,14,15,16,17,
132	0.00003521	0.039	971.56	127.90	503.00	503.00	0.00	7,14,15,16,17,
133	0.00002276	0.024	942.36	134.14	595.00	596.00	0.00	7,14,15,16,17,
134	0.00003717	0.042	1015.36	128.00	524.00	524.00	0.00	7,14,15,16,17,
135	0.00006199	0.062	863.62	129.08	401.00	402.00	0.00	7,14,15,16,17,
136	0.00001351	0.018	1234.36	132.70	576.00	577.00	0.00	7,14,15,16,17,
137	0.00001939	0.020	883.96	129.50	426.00	427.00	0.00	7,14,15,16,17,
138	0.00000185	0.002	1146.76	118.68	404.00	405.00	0.00	7,14,15,16,17,
139	0.00000359	0.004	1117.56	124.92	497.00	497.00	0.00	7,14,15,16,17,
140	0.00000587	0.008	1190.56	118.78	425.00	426.00	0.00	7,14,15,16,17,
141	0.00002918	0.033	1012.70	120.44	324.00	324.00	0.00	7,14,15,16,17,
142	0.00000213	0.003	1409.56	123.48	478.00	478.00	0.00	7,14,15,16,17,
143	0.00000306	0.004	1059.16	120.28	328.00	328.00	0.00	7,14,15,16,17,
144	0.00000621	0.007	1000.76	126.18	395.00	396.00	0.00	7,14,15,16,17,
145	0.00000138	0.002	1380.36	129.72	570.00	571.00	0.00	7,14,15,16,17,
146	0.00000190	0.002	1161.36	125.02	518.00	518.00	0.00	7,14,15,16,17,
147	0.00000198	0.002	1029.96	126.52	420.00	421.00	0.00	7,14,15,16,17,
148	0.00000369	0.005	1292.76	124.73	376.00	377.00	0.00	7,14,15,16,17,
149	0.00000113	0.002	1453.36	123.57	499.00	499.00	0.00	7,14,15,16,17,
150	0.00000162	0.002	1102.96	120.38	349.00	349.00	0.00	7,14,15,16,17,

151	0.00002920	0.032	971.56	127.97	402.00	384.00	220.00	7,14,15,16,17,
152	0.00001541	0.018	1015.36	128.07	419.00	409.00	220.00	7,14,15,16,17,
153	0.00001608	0.016	883.96	129.57	341.00	292.00	220.00	7,14,15,16,17,
154	0.00000298	0.004	1117.56	124.99	398.00	388.00	208.00	7,14,15,16,17,
155	0.00000243	0.003	1190.56	118.85	340.00	291.00	220.00	7,14,15,16,17,
156	0.00000254	0.003	1059.16	120.35	262.00	174.00	220.00	7,14,15,16,17,
157	0.00000157	0.002	1161.36	125.09	414.00	414.00	208.00	7,14,15,16,17,
158	0.00000164	0.002	1029.96	126.59	336.00	297.00	208.00	7,14,15,16,17,
159	0.00000134	0.002	1102.96	120.45	279.00	199.00	220.00	7,14,15,16,17,
160	0.00075949	0.113	11.41	137.51	601.00	602.00	0.00	14,15,16,17,
161	0.00011992	0.038	186.61	128.29	503.00	503.00	0.00	14,15,16,17,
162	0.00007751	0.023	157.41	134.53	595.00	596.00	0.00	14,15,16,17,
163	0.00012658	0.045	230.41	128.39	524.00	524.00	0.00	14,15,16,17,
164	0.00021114	0.044	78.66	129.47	401.00	402.00	0.00	14,15,16,17,
165	0.00004603	0.027	449.41	133.09	576.00	577.00	0.00	14,15,16,17,
166	0.00006604	0.015	99.01	129.89	426.00	427.00	0.00	14,15,16,17,
167	0.00000631	0.003	361.81	119.07	404.00	405.00	0.00	14,15,16,17,
168	0.00000316	0.001	303.41	131.55	589.00	590.00	0.00	14,15,16,17,
169	0.00000116	0.001	887.41	128.66	551.00	552.00	0.00	14,15,16,17,
170	0.00000791	0.004	449.41	119.26	446.00	447.00	0.00	14,15,16,17,
171	0.00001224	0.006	332.61	125.31	497.00	497.00	0.00	14,15,16,17,
172	0.00001999	0.010	405.61	119.17	425.00	426.00	0.00	14,15,16,17,
173	0.00010793	0.039	241.28	120.42	324.00	324.00	0.00	14,15,16,17,
174	0.00000727	0.005	624.61	123.87	478.00	478.00	0.00	14,15,16,17,
175	0.00001043	0.004	274.21	120.67	328.00	328.00	0.00	14,15,16,17,
176	0.00001292	0.006	376.41	125.41	518.00	518.00	0.00	14,15,16,17,
177	0.00002114	0.007	215.81	126.57	395.00	396.00	0.00	14,15,16,17,
178	0.00000470	0.003	595.41	130.11	570.00	571.00	0.00	14,15,16,17,
179	0.00000674	0.003	245.01	126.91	420.00	421.00	0.00	14,15,16,17,
180	0.00000767	0.006	668.41	123.96	499.00	499.00	0.00	14,15,16,17,
181	0.00001101	0.005	318.01	120.77	349.00	349.00	0.00	14,15,16,17,
182	0.00001255	0.008	507.81	125.12	376.00	377.00	0.00	14,15,16,17,
183	0.00000102	0.001	551.61	116.19	419.00	420.00	0.00	14,15,16,17,
184	0.00000223	0.001	303.41	112.36	255.00	255.00	0.00	14,15,16,17,
185	0.00000284	0.001	332.61	112.71	280.00	280.00	0.00	14,15,16,17,
186	0.00000118	0.001	347.21	112.46	276.00	276.00	0.00	14,15,16,17,
187	0.00000123	0.000	215.81	113.97	185.00	185.00	0.00	14,15,16,17,
188	0.00075949	0.113	11.41	137.51	301.00	601.00	301.00	7,15,16,17,
189	0.00011992	0.038	186.61	128.29	251.00	503.00	252.00	7,15,16,17,
190	0.00007751	0.023	157.41	134.53	298.00	595.00	298.00	7,15,16,17,
191	0.00012658	0.045	230.41	128.39	262.00	524.00	262.00	7,15,16,17,
192	0.00021114	0.044	78.66	129.47	201.00	401.00	201.00	7,15,16,17,
193	0.00004603	0.027	449.41	133.09	288.00	577.00	288.00	7,15,16,17,
194	0.00006604	0.015	99.01	129.89	213.00	427.00	213.00	7,15,16,17,
195	0.00000631	0.003	361.81	119.07	202.00	405.00	202.00	7,15,16,17,
196	0.00000316	0.001	303.41	131.55	295.00	589.00	295.00	7,15,16,17,
197	0.00000116	0.001	887.41	128.66	276.00	551.00	276.00	7,15,16,17,
198	0.00000791	0.004	449.41	119.26	223.00	447.00	223.00	7,15,16,17,
199	0.00001224	0.006	332.61	125.31	248.00	497.00	249.00	7,15,16,17,
200	0.00001999	0.010	405.61	119.17	213.00	425.00	213.00	7,15,16,17,

201	0.00010793	0.039	241.28	120.42	162.00	324.00	162.00	7,15,16,17,
202	0.00000727	0.005	624.61	123.87	239.00	478.00	239.00	7,15,16,17,
203	0.00001043	0.004	274.21	120.67	164.00	328.00	164.00	7,15,16,17,
204	0.00001292	0.006	376.41	125.41	259.00	518.00	259.00	7,15,16,17,
205	0.00002114	0.007	215.81	126.57	198.00	395.00	198.00	7,15,16,17,
206	0.00000470	0.003	595.41	130.11	285.00	571.00	285.00	7,15,16,17,
207	0.00000674	0.003	245.01	126.91	210.00	421.00	210.00	7,15,16,17,
208	0.00000767	0.006	668.41	123.96	249.00	499.00	250.00	7,15,16,17,
209	0.00001101	0.005	318.01	120.77	174.00	349.00	175.00	7,15,16,17,
210	0.00001255	0.008	507.81	125.12	188.00	377.00	188.00	7,15,16,17,
211	0.00000102	0.001	551.61	116.19	210.00	419.00	210.00	7,15,16,17,
212	0.00000223	0.001	303.41	112.36	128.00	255.00	127.00	7,15,16,17,
213	0.00000284	0.001	332.61	112.71	140.00	280.00	140.00	7,15,16,17,
214	0.00000118	0.001	347.21	112.46	138.00	276.00	138.00	7,15,16,17,
215	0.00000123	0.000	215.81	113.97	92.00	185.00	93.00	7,15,16,17,
216	0.00075949	0.113	11.41	137.51	601.00	301.00	301.00	7,14,16,17,
217	0.00011992	0.038	186.61	128.29	503.00	251.00	252.00	7,14,16,17,
218	0.00007751	0.023	157.41	134.53	595.00	298.00	298.00	7,14,16,17,
219	0.00012658	0.045	230.41	128.39	524.00	262.00	262.00	7,14,16,17,
220	0.00021114	0.044	78.66	129.47	401.00	201.00	201.00	7,14,16,17,
221	0.00004603	0.027	449.41	133.09	577.00	288.00	288.00	7,14,16,17,
222	0.00006604	0.015	99.01	129.89	427.00	213.00	213.00	7,14,16,17,
223	0.00000631	0.003	361.81	119.07	405.00	202.00	202.00	7,14,16,17,
224	0.00000316	0.001	303.41	131.55	589.00	295.00	295.00	7,14,16,17,
225	0.00000116	0.001	887.41	128.66	551.00	276.00	276.00	7,14,16,17,
226	0.00000791	0.004	449.41	119.26	447.00	223.00	223.00	7,14,16,17,
227	0.00001224	0.006	332.61	125.31	497.00	248.00	249.00	7,14,16,17,
228	0.00001999	0.010	405.61	119.17	425.00	213.00	213.00	7,14,16,17,
229	0.00010793	0.039	241.28	120.42	324.00	162.00	162.00	7,14,16,17,
230	0.00000727	0.005	624.61	123.87	478.00	239.00	239.00	7,14,16,17,
231	0.00001043	0.004	274.21	120.67	328.00	164.00	164.00	7,14,16,17,
232	0.00001292	0.006	376.41	125.41	518.00	259.00	259.00	7,14,16,17,
233	0.00002114	0.007	215.81	126.57	395.00	198.00	198.00	7,14,16,17,
234	0.00000470	0.003	595.41	130.11	571.00	285.00	285.00	7,14,16,17,
235	0.00000674	0.003	245.01	126.91	421.00	210.00	210.00	7,14,16,17,
236	0.00000767	0.006	668.41	123.96	499.00	249.00	250.00	7,14,16,17,
237	0.00001101	0.005	318.01	120.77	349.00	174.00	175.00	7,14,16,17,
238	0.00001255	0.008	507.81	125.12	377.00	188.00	188.00	7,14,16,17,
239	0.00000102	0.001	551.61	116.19	419.00	210.00	210.00	7,14,16,17,
240	0.00000223	0.001	303.41	112.36	255.00	128.00	127.00	7,14,16,17,
241	0.00000284	0.001	332.61	112.71	280.00	140.00	140.00	7,14,16,17,
242	0.00000118	0.001	347.21	112.46	276.00	138.00	138.00	7,14,16,17,
243	0.00000123	0.000	215.81	113.97	185.00	92.00	93.00	7,14,16,17,
244	0.00075949	0.113	11.41	137.51	301.00	601.00	301.00	7,14,15,17,
245	0.00011992	0.038	186.61	128.29	251.00	503.00	252.00	7,14,15,17,
246	0.00007751	0.023	157.41	134.53	298.00	595.00	298.00	7,14,15,17,
247	0.00012658	0.045	230.41	128.39	262.00	524.00	262.00	7,14,15,17,
248	0.00021114	0.044	78.66	129.47	201.00	401.00	201.00	7,14,15,17,
249	0.00004603	0.027	449.41	133.09	288.00	577.00	288.00	7,14,15,17,
250	0.00006604	0.015	99.01	129.89	213.00	427.00	213.00	7,14,15,17,

251	0.00000631	0.003	361.81	119.07	202.00	405.00	202.00	7,14,15,17,
252	0.00000316	0.001	303.41	131.55	295.00	589.00	295.00	7,14,15,17,
253	0.00000116	0.001	887.41	128.66	276.00	551.00	276.00	7,14,15,17,
254	0.00000791	0.004	449.41	119.26	223.00	447.00	223.00	7,14,15,17,
255	0.00001224	0.006	332.61	125.31	248.00	497.00	249.00	7,14,15,17,
256	0.00001999	0.010	405.61	119.17	213.00	425.00	213.00	7,14,15,17,
257	0.00010793	0.039	241.28	120.42	162.00	324.00	162.00	7,14,15,17,
258	0.00000727	0.005	624.61	123.87	239.00	478.00	239.00	7,14,15,17,
259	0.00001043	0.004	274.21	120.67	164.00	328.00	164.00	7,14,15,17,
260	0.00001292	0.006	376.41	125.41	259.00	518.00	259.00	7,14,15,17,
261	0.00002114	0.007	215.81	126.57	198.00	395.00	198.00	7,14,15,17,
262	0.00000470	0.003	595.41	130.11	285.00	571.00	285.00	7,14,15,17,
263	0.00000674	0.003	245.01	126.91	210.00	421.00	210.00	7,14,15,17,
264	0.00000767	0.006	668.41	123.96	249.00	499.00	250.00	7,14,15,17,
265	0.00001101	0.005	318.01	120.77	174.00	349.00	175.00	7,14,15,17,
266	0.00001255	0.008	507.81	125.12	188.00	377.00	188.00	7,14,15,17,
267	0.00000102	0.001	551.61	116.19	210.00	419.00	210.00	7,14,15,17,
268	0.00000223	0.001	303.41	112.36	128.00	255.00	127.00	7,14,15,17,
269	0.00000284	0.001	332.61	112.71	140.00	280.00	140.00	7,14,15,17,
270	0.00000118	0.001	347.21	112.46	138.00	276.00	138.00	7,14,15,17,
271	0.00000123	0.000	215.81	113.97	92.00	185.00	93.00	7,14,15,17,
272	0.00075949	0.113	11.41	137.51	601.00	301.00	301.00	7,14,15,16,
273	0.00011992	0.038	186.61	128.29	503.00	251.00	252.00	7,14,15,16,
274	0.00007751	0.023	157.41	134.53	595.00	298.00	298.00	7,14,15,16,
275	0.00012658	0.045	230.41	128.39	524.00	262.00	262.00	7,14,15,16,
276	0.00021114	0.044	78.66	129.47	401.00	201.00	201.00	7,14,15,16,
277	0.00004603	0.027	449.41	133.09	577.00	288.00	288.00	7,14,15,16,
278	0.00006604	0.015	99.01	129.89	427.00	213.00	213.00	7,14,15,16,
279	0.00000631	0.003	361.81	119.07	405.00	202.00	202.00	7,14,15,16,
280	0.00000316	0.001	303.41	131.55	589.00	295.00	295.00	7,14,15,16,
281	0.00000116	0.001	887.41	128.66	551.00	276.00	276.00	7,14,15,16,
282	0.00000791	0.004	449.41	119.26	447.00	223.00	223.00	7,14,15,16,
283	0.00001224	0.006	332.61	125.31	497.00	248.00	249.00	7,14,15,16,
284	0.00001999	0.010	405.61	119.17	425.00	213.00	213.00	7,14,15,16,
285	0.00010793	0.039	241.28	120.42	324.00	162.00	162.00	7,14,15,16,
286	0.00000727	0.005	624.61	123.87	478.00	239.00	239.00	7,14,15,16,
287	0.00001043	0.004	274.21	120.67	328.00	164.00	164.00	7,14,15,16,
288	0.00001292	0.006	376.41	125.41	518.00	259.00	259.00	7,14,15,16,
289	0.00002114	0.007	215.81	126.57	395.00	198.00	198.00	7,14,15,16,
290	0.00000470	0.003	595.41	130.11	571.00	285.00	285.00	7,14,15,16,
291	0.00000674	0.003	245.01	126.91	421.00	210.00	210.00	7,14,15,16,
292	0.00000767	0.006	668.41	123.96	499.00	249.00	250.00	7,14,15,16,
293	0.00001101	0.005	318.01	120.77	349.00	174.00	175.00	7,14,15,16,
294	0.00001255	0.008	507.81	125.12	377.00	188.00	188.00	7,14,15,16,
295	0.00000102	0.001	551.61	116.19	419.00	210.00	210.00	7,14,15,16,
296	0.00000223	0.001	303.41	112.36	255.00	128.00	127.00	7,14,15,16,
297	0.00000284	0.001	332.61	112.71	280.00	140.00	140.00	7,14,15,16,
298	0.00000118	0.001	347.21	112.46	276.00	138.00	138.00	7,14,15,16,
299	0.00000123	0.000	215.81	113.97	185.00	92.00	93.00	7,14,15,16,
300	0.00000133	0.000	22.81	137.49	400.00	800.00	0.00	15,16,17,

301	0.00000133	0.000	22.81	137.49	800.00	400.00	0.00	14,16,17,
302	0.00000133	0.000	22.81	137.49	400.00	800.00	0.00	14,15,17,
303	0.00000133	0.000	22.81	137.49	800.00	400.00	0.00	14,15,16,
304	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,16,17,
305	0.00000133	0.000	22.81	137.49	0.00	800.00	400.00	7,15,17,
306	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,15,16,
307	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,14,17,
308	0.00000133	0.000	22.81	137.49	800.00	0.00	400.00	7,14,16,
309	0.00000133	0.000	22.81	137.49	400.00	400.00	400.00	7,14,15,

PROBABILITY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = 0.99829346
 FREQUENCY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = 261.3188
 TOTAL STATES OF MODEL ADEQUACY EQUIVALENTS = 309
 TIME TO DEVELOP MODEL ADEQUACY EQUIVALENTS = 252 Sec

ภาคผนวก ค.

๕

โปรแกรมสำหรับใช้ในการคำนวณ (Source programe)

[Last version was been updated 28-9-91]

UNIT ALLDATA;

INTERFACE

USES DOS,CRT,printer;

CONST
 MAXBUS = 35;
 MAXELEM = 50;
 ADQBUS = 3;
 MAXSTAT = 1050;
 PBCUTOFF = 1.0E-8;
 FQCUTOFF = 1.0E-6;

TYPE
 COMPONENT = (GN, LN, TR, LD);
 ELEMREC = RECORD
 ENO : BYTE;
 ECAP : REAL;
 CASE ETYPE : COMPONENT OF
 GN : (GNBUS : BYTE;
 GNFR,GNRR : REAL;
 GNID : BYTE);
 LN : (LSDB,LEDB : BYTE;
 LNFR,LNRR : REAL);
 LD : (LDBUS : BYTE);
 END;
 ELENTYPE = FILE OF ELEMREC;

SETNO = SET OF 1..MAXELEM;
 CTGLEVEL = 0..4;
 GROUP = (MS,SS);
 MSSS = ARRAY[1..MAXELEM] OF GROUP;
 BUS = ARRAY[1..MAXELEM] OF BYTE;
 NODE = ARRAY[1..MAXELEM] OF REAL;
 ARRY = ARRAY[1..MAXBUS,1..MAXBUS] OF REAL;
 AQE = ARRAY[1..ADQBUS] OF REAL;
 ARRZ = ARRAY[1..MAXSTAT] OF AQE;
 ARRX = ARRAY[1..MAXSTAT] OF REAL;
 ARRW = ARRAY[1..MAXSTAT] OF SETNO;

VAR
 TN_BUS,TN_GEN,TN_LINE,TN_LOAD : BYTE;
 TMS_BUS,TSS_BUS : BYTE; [amount of buses in MS,SS]
 MS_BUS,SS_BUS : SETNO; [bus number in MS,SS/INPUT MODE]
 MSB,SSB : BUS; [define new bus number in MS,SS]
 GENNO,LINENO,LOADNO,GIDEN : BUS;
 GENCAP,LINECAP,LOADCAP : NODE;
 GENB,SB,EB,LOADB : BUS;
 NGENB,NSB,NEB,NLOADB : BUS;
 G_GENB,G_SB,G_EB,G_LOADB : MSSS;
 MSMAT,SSMAT : ARRY;
 FRG,FRL : NODE; [failure rate of element]
 RRG,RRL : NODE; [repair rate of elements]
 PBG,PBL : NODE; [available probability]
 PBG1,PBL1,PBC1 : REAL; [full capacity prob of MS]
 DNG1,DNL1,DNC1 : REAL; [departure rate to down state of MS]
 SETG1,SETL1,SETC1 : SETNO; [set of element num in MS by type]
 NOG1,NOL1,NOC1 : BYTE; [number of element in MS by type]
 PBMS,UPMS,DNMS,FQMS : REAL;
 PBG2,PBL2 : REAL; [full capacity prob of SS]
 DNG2,DNL2 : REAL; [departure rate to down state of SS]
 SETG2,SETL2 : SETNO; [set of element num in SS by type]
 NOG2,NOL2 : BYTE; [number of element in SS by type]
 PBSS,UPSS,DNSS,FQSS : REAL;

```

SETCMS,SETCSS      : SETNO;[set of bus in MS connected with SS]
MSAB , SSAB       : ARRAY[1..10]OF BYTE;
TMSAB ,TSSAB      : BYTE;
GENMS ,GENSS      : BYTE;
INDEXT            : BOOLEAN; [conventional=FALSE,adq=TRUE]
INTSET            : SETNO; [ bus in area of interest]
CURT              : REAL; [% OF CURTAILABLE LOAD IN SYSTEM BUSES]
RD               : REAL; [APPROXIMATE STEP LEVEL IN MODEL ADQ ]
COMADQ           : REAL; [CUT-OFF VALUE TO SCREEN OUT PROB STATE IN MODEL ADQ]
LST              : TEXT;

```

```

FUNCTION PERMUTATION (C,R:LONGINT):LONGINT;
FUNCTION POWER (B:REAL;N:LONGINT):REAL;
PROCEDURE CONTING(VAR ELEM      : SETNO;
                  VAR NOELEM    : BYTE;
                  VAR LEVEL     : CTGLEVEL;
                  VAR STATENO   : LONGINT;
                  VAR ELEMOUT   : SETNO );
PROCEDURE MAXFLOW(VAR NETCHAR:ARRY;NB:BYTE;DUMMY:BOOLEAN);
PROCEDURE DATAENTRY;
PROCEDURE READDATA;
PROCEDURE MODFAREA ;
PROCEDURE ROUNDING (      ALD: REAL; [all connected lines capacities]
                       TMSAB1: BYTE; [total number of connected bus in MS]
                       RD: REAL; [increment in each step 5.0 or10.0 ]
                       VAR SLD: AQE; [marginal line capacity before rounding]
                       VAR RLD: AQE); [marginal line capacity after rounding]
  Procedure CTG(R:BYTE ; MS_SS:GROUP ; SMS_SS:SETNO ; TMS_SS:BYTE ;
               VAR CTL1,CTL2:SETNO );
  Procedure CTL(T:BYTE ; MS_SS:GROUP ; SMS_SS:SETNO ; TMS_SS:BYTE ;
               VAR CTL3,CTL4,CTL5,CTL6:SETNO);
  Procedure TIMECHECK(VAR TIMESEC:LONGINT);

```

IMPLEMENTATION

```

FUNCTION PERMUTATION (C,R:LONGINT):LONGINT;
VAR   X,Y : LONGINT;
BEGIN
  IF R=0 THEN PERMUTATION :=1
  ELSE
    BEGIN
      C:=C+1; R:=R+1; X:=1; Y:=1;
      REPEAT
        C:=C-1; R:=R-1;
        X:= X*C; Y:=Y*R; UNTIL R=1;
        PERMUTATION :=X DIV Y;
      END;
    END;
END;
FUNCTION POWER (B:REAL;N:LONGINT):REAL;
VAR   S: LONGINT; T :REAL;
BEGIN
  S:=1; T:=1;
  WHILE S<= N DO BEGIN T:=T*B; S:=S+1; END;
  POWER := T;
END;

```

```

PROCEDURE CONTING(VAR ELEM      : SETNO;
                  VAR NOELEM    : BYTE;
                  VAR LEVEL     : CTGLEVEL;
                  VAR STATENO   : LONGINT;
                  VAR ELEMOUT   : SETNO );

```

```

VAR      ELEMENT : ARRAY[1..MAXELEM] OF SETNO;
        X,Y,Z,XY      : BYTE ;
        DUMA,DUMB,DUMC      : LONGINT;
        DUM1,DUM2,DUM3,DUM4,DUM5,DUM6 : LONGINT;
        PROBE      : BOOLEAN;
        KEY       : CHAR;

BEGIN [main procedure CONTING]
  FOR X := 1 TO NOELEM DO ELEMENT[X]:=[];
  Y:=0;
  FOR X := 1 TO 100 DO
  BEGIN IF X IN ELEM THEN Y:=Y+1; END;
  IF Y <> NOELEM THEN
  BEGIN WRITELN ('Set of element not match with No.of element PRESS [E] TO EXIT');
    REPEAT KEY:=UPCASE(READKEY); UNTIL KEY ='E':halt;
  END;
  Y:=1;
  FOR X := 1 TO NOELEM DO
  BEGIN
    PROBE:= FALSE;
    REPEAT
    IF Y IN ELEM THEN
    BEGIN ELEMENT[X]:=ELEMENT[X]+[Y]; Y := Y+1; PROBE := TRUE; END
    ELSE BEGIN Y:= Y+1; PROBE := FALSE; END;
    UNTIL PROBE=TRUE;
  END;
  IF (LEVEL IN {0..4})AND(STATENO>0)AND(STATENO<=Round(PERMUTATION(NOELEM,LEVEL))) THEN
  BEGIN
  CASE LEVEL OF
    0 : ELEMOUT :=[];
    1 : ELEMOUT := ELEMENT[STATENO];
    2 : BEGIN
        DUM1 := 0; X := 0;
        REPEAT X := X+1;
        DUMA := NOELEM - X ;
        DUM1 := DUM1 + DUMA;
        UNTIL STATENO <= DUM1;
        DUM2 :=STATENO - (DUM1-DUMA) + X;
        ELEMOUT:=ELEMENT[X]+ELEMENT[DUM2];
      END;
    3 : BEGIN
        DUM1 := 0; X := 0;
        REPEAT X := X+1;
        DUMA := (NOELEM-1-X+1)*(NOELEM-1-X) DIV 2;
        DUM1 := DUM1 + DUMA;
        UNTIL STATENO <= DUM1;
        DUM2 :=STATENO - (DUM1-DUMA);
        DUM3 := 0; Y := X;
        REPEAT Y := Y+1;
        DUMB := NOELEM - Y;
        DUM3 := DUM3 + DUMB;
        UNTIL DUM2 <= DUM3;
        DUM4 := DUM2-(DUM3-DUMB)+Y;
        ELEMOUT:= ELEMENT[X]+ELEMENT[Y]+ELEMENT[DUM4];
      END;
    4 : BEGIN
        DUM1 := 0; X := 0;
        REPEAT X := X+1;
        DUMA := 0;
        FOR XY := X TO NOELEM-3 DO
        DUMA := DUMA + (NOELEM-2-XY+1)*(NOELEM-2-XY) DIV 2;

```

```

DUM1 := DUM1 + DUMA;
UNTIL STATENC <= DUM1;
DUM2 := STATENC - (DUM1-DUMA);
DUM3 := 0; Y := X;
REPEAT Y := Y+1;
DUMB := (NOELEM-1-Y+1)*(NOELEM-1-Y) DIV 2;
DUM3 := DUM3 + DUMB;
UNTIL DUM2 <= DUM3;
DUM4 := DUM2-(DUM3-DUMB);
DUM5 := 0; Z := Y;
REPEAT Z := Z+1;
DUMC := NOELEM-Z;
DUM5 := DUM5 + DUMC;
UNTIL DUM4 <= DUM5;
DUM6 := DUM4-(DUM5-DUMC)+Z;
ELEMOUT := ELEMENT[X]+ELEMENT[Y]+ELEMENT[Z]+ELEMENT[DUM6];
END;
END;
END
ELSE begin IF NOT (LEVEL IN [0..4]) THEN
  writeln('Selection over contingency level PRESS [E] TO EXIT ');
  else writeln('Contingency state is improper PRESS [E] TO EXIT ');
  REPEAT KEY:=UPCASE(READKEY); UNTIL KEY ='E':halt; end;
END;

PROCEDURE MAXFLOW(VAR NETCHAR:ARRAY;NB:BYTE;DUMMY:BOOLEAN);
VAR LABL,SCAN : ARRAY[1..MAXBUS] OF BOOLEAN;
    FROMNODE : BUS;
    ADDCAP : NODE;
    I,J,M,N,MN : BYTE;
    TERMINATE : BOOLEAN;

PROCEDURE INITIAL; {SCAN SOURCE}
VAR M :BYTE;
BEGIN
  FOR M:=2 TO NB+2 DO
    BEGIN
      IF NETCHAR[1,M]<>0 THEN
        BEGIN
          LABL[M] := TRUE;
          SCAN[M] := FALSE;
          FROMNODE[M] := 1;
          ADDCAP[M] := NETCHAR[1,M]-NETCHAR[M,1];
          IF ADDCAP[M]=0.0 THEN
            BEGIN LABL[M]:=FALSE; FROMNODE[M]:=0; END;
        END
      ELSE
        BEGIN
          LABL[M] := FALSE;
          SCAN[M] := FALSE;
          FROMNODE[M] := 0;
          ADDCAP[M] := 0;
        END;
    END;
  END;
  LABL[1] := TRUE; SCAN[1] := TRUE;
END;

PROCEDURE SCANNED(M1:BYTE);
VAR N,A,B : BYTE;
    DEL : REAL;
BEGIN
  FOR N := 2 TO NB+2 DO

```

```

BEGIN
  IF M1<>N THEN
    BEGIN
      IF LABL[N]=FALSE THEN
        BEGIN
          IF M1<N THEN BEGIN A:=M1; B:=N; END
          ELSE BEGIN A:=N; B:=M1; END;
          IF NETCHAR[A,B] <>0 THEN
            BEGIN
              IF M1<N THEN DEL := NETCHAR[M1,N]-NETCHAR[N,M1] [flow i to j]
              ELSE DEL := NETCHAR[N,M1]+NETCHAR[M1,N]; [flow j to i]
              IF ADDCAP[M1]>DEL THEN ADDCAP[N] := DEL
              ELSE ADDCAP[N] := ADDCAP[M1];
              IF ADDCAP[N]=0 THEN
                BEGIN LABL[N] :=FALSE;
                  SCAN[N] :=FALSE;
                  FROMNODE[N]:=0;
                END
                ELSE BEGIN
                  LABL[N] := TRUE;
                  SCAN[N] := FALSE;
                  FROMNODE[N]:= M1;
                END;
            END;
          END;
        END;
      END;
    END;
  END;
  END;
  END;
  END;
  BEGIN [MAIN PROCEDURE MAXFLOW]
    [CLEAR LEFT SIDE OF DIAGONAL MATRIX OR NETCHAR]
    IF DUMMY = TRUE THEN
      BEGIN
        FOR M :=2 TO NB+2 DO
          BEGIN
            FOR N :=1 TO M DO NETCHAR[M,N] :=0;
          END;
        NETCHAR[1,NB+2] :=0;
        END;
        INITIAL;
        M:=1; TERMINATE :=FALSE;
        REPEAT M:=M+1;
        IF (LABL[M] = TRUE) AND (SCAN[M] = FALSE) THEN
          BEGIN
            SCANNED(M);
            IF (LABL[NB+2]=FALSE)AND(M<NB+1) THEN BEGIN SCAN[M]:=TRUE; M:=1; END
            ELSE IF (LABL[NB+2]=FALSE)AND(M=NB+1) THEN
              BEGIN
                MN:=1;
                REPEAT MN:=MN+1;
                UNTIL (((SCAN[MN]=FALSE)AND(LABL[MN]=TRUE))-TRUE)OR(MN=NB);
                IF MN<NB THEN BEGIN SCAN[M]:=TRUE; M:=1; END
                ELSE IF (SCAN[NB]=FALSE)AND(LABL[NB]=TRUE) THEN
                  BEGIN SCAN[M]:=TRUE; M:=1; END
                ELSE TERMINATE:=TRUE;
              END
            END
          END
        ELSE
          BEGIN
            J :=NB+2;
            REPEAT
              I := FROMNODE[J];
              IF I<J THEN NETCHAR[J,I] := NETCHAR[J,I]+ADDCAP[NB+2]

```



```

ELSE NETCHAR[I,J] := NETCHAR[I,J]-ADDCAP[NB+2];
J := I;
UNTIL I=1;
INITIAL;
M:=1;
END;

END
ELSE IF (LABL[M]=FALSE)AND(SCAN[M]=FALSE)and(m=nb+1) THEN
  TERMINATE :=TRUE
ELSE IF (SCAN[M]=TRUE)AND(M=NB+1) THEN TERMINATE := TRUE;
UNTIL TERMINATE =TRUE;
END;

PROCEDURE DATAENTRY;
VAR
  ELEMTYPE : ELEMTYPE;
  SYSNAME : STRING(8);
  MODE : BOOLEAN;
  KEY1,KEY2 : CHAR ;
Procedure SHOWDATA(ONEREC:ELEMREC);
BEGIN
  WITH ONEREC DO
  BEGIN
    CASE ETYPE OF
      GN : BEGIN
        WRITELN('Generator No. = ',ENO);
        WRITELN('GEN capacity (MW) = ',ECAP);
        WRITELN('GEN bus No. = ',GNBUS);
        WRITELN('GEN failure rate = ',GNFR);
        WRITELN('GEN repair rate = ',GNRR);
        WRITELN('GEN identical = ',GNID);
      END;
      LN : BEGIN
        WRITELN('Line No. = ',ENO);
        WRITELN('LINE capacity (MW) = ',ECAP);
        WRITELN('LN sending bus No. = ',LSDB);
        WRITELN('LN ending bus No. = ',LEDB);
        WRITELN('LN failure rate = ',LNFR);
        WRITELN('LN repair rate = ',LNRR);
      END;
      LD : BEGIN
        WRITELN('Load No. = ',ENO);
        WRITELN('LOAD demand (MW) = ',ECAP);
        WRITELN('LOAD bus No. = ',LDBUS);
      END;
    END;|case|
  END;
END;

Procedure ENTERDATA(VAR FI : ELEMTYPE);
VAR
  ONEREC : ELEMREC;
  KEY : CHAR;
BEGIN
  SEEK(FI,FileSize(FI));
  REPEAT
  WITH ONEREC DO
  BEGIN
    WRITELN(' INPUT DATA OF EACH ELEMENT ');
    WRITELN('***** SELECT TYPE OF ELEMENT ***** ');
    WRITE ('G_enerator L_line loa_D_Quit-PRESS [G,L,D,Q] ONLY ONE ');

```

```

REPEAT KEY := UPCASE(Readkey);
UNTIL KEY IN ['G','L','D','Q']; write(key); writeln;
IF KEY <> 'Q' THEN
BEGIN
  CASE KEY OF
    'G': BEGIN ETYPE := GN;
          WRITE('GENERATOR NO      = '); READLN(ENO);
          WRITE('GEN CAPACITY (MW) = '); READLN(ECAP);
          WRITE('GEN BUS NO        = '); READLN(GNBUS);
          WRITE('GEN FAILURE RATE  = '); READLN(GNFR);
          WRITE('GEN REPAIR RATE   = '); READLN(GNRR);
          WRITE('GEN IDENTICAL     = '); READLN(GNID);
        END;
    'L': BEGIN ETYPE := LN;
          WRITE('LINE NO            = '); READLN(ENO);
          WRITE('LN CAPACITY (MW)   = '); READLN(ECAP);
          WRITE('LN SENDING BUS NO  = '); READLN(LSDB);
          WRITE('LN ENDING BUS NO   = '); READLN(LEDB);
          WRITE('LN FAILURE RATE    = '); READLN(LNFR);
          WRITE('LN REPAIR RATE     = '); READLN(LNRR);
        END;
    'D': BEGIN ETYPE := LD;
          WRITE('LOAD NO              = '); READLN(ENO);
          WRITE('LOAD DEMAND (MW)      = '); READLN(ECAP);
          WRITE('LOAD BUS NO          = '); READLN(LDBUS);
        END;
  END; [case]
  WRITE(FI,ONEREC);
END;
END;
UNTIL KEY = 'Q'; WRITELN ('FINISH ENTERDATA');
CLOSE(FI);
END;

Procedure CORRECTDATA (VAR FI:ELEMTYPE);
VAR  ONEREC  : ELEMREC;
      KEY    : CHAR;
      COMPTYPE : COMPONENT;
      COMPNO  : BYTE;
      FINISHED : BOOLEAN;
BEGIN
  REPEAT
  WRITE ('***** TYPE OF ELEMENT TO BE CORRECTED PRESS [G,L,D] ONLY ONE *****');
  REPEAT KEY := UPCASE(READKEY);
  UNTIL KEY IN ['G','L','D']; write(key); writeln;
  CASE KEY OF
    'G': COMPTYPE := GN;
    'L': COMPTYPE := LN;
    'D': COMPTYPE := LD;
  END;
  WRITE(' CORRECTED ELEMENT NO = '); READLN(COMPNO);
  FINISHED := FALSE;  SEEK(FI,0);
  WHILE NOT EOF(FI) AND NOT FINISHED DO
  BEGIN
    READ(FI,ONEREC);
    WITH ONEREC DO
    BEGIN
      IF (ETYPE=COMPTYPE) AND (ENO=COMPNO) THEN
      BEGIN
        FINISHED := TRUE;
        WRITELN('EXISTING DATA OF ELEMENT');
      END;
    END;
  END;
END;

```

```

        SHOWDATA(ONEREC);
        SEEK(FI,FilePos(FI)-1);
        WRITELN('CORRECTED DATA OF ELEMENT NO = ',COMPNO:3);
        ETYPE := COMPTYPE; ENO := COMPNO;
        CASE ETYPE OF
GN : BEGIN
        WRITE('GENERATOR NO      = '); READLN(ENO);
        WRITE('GEN CAPACITY (MW) = '); READLN(ECAP);
        WRITE('GEN BUS NO       = '); READLN(GNBUS);
        WRITE('GEN FAILURE RATE = '); READLN(GNFR);
        WRITE('GEN REPAIR RATE  = '); READLN(GNRR);
        WRITE('GEN IDENTICAL   = '); READLN(GNID);
        END;
LN : BEGIN
        WRITE('LINE NO          = '); READLN(ENO);
        WRITE('LN CAPACITY (MW)  = '); READLN(ECAP);
        WRITE('LN SENDING BUS NO = '); READLN(LSDB);
        WRITE('LN ENDING BUS NO  = '); READLN(LEDB);
        WRITE('LN FAILURE RATE   = '); READLN(LNFR);
        WRITE('LN REPAIR RATE    = '); READLN(LNRR);
        END;
LD : BEGIN
        WRITE('LOAD NO          = '); READLN(ENO);
        WRITE('LOAD DEMAND (MW)   = '); READLN(ECAP);
        WRITE('LOAD BUS NO       = '); READLN(LDBUS);
        END;
        END;
        WRITE(FI,ONEREC);
    END;
END;
WRITELN('##### FINISHED CORRECTDATA Yes OR No PRESS [Y,N] ##### ');
REPEAT KEY := UPCASE(READKEY);
UNTIL KEY IN ['Y','N'];
UNTIL KEY = 'Y';
CLOSE(FI);
END;

Procedure DELETEDATA(VAR FI : ELEMTYPE);
VAR
    ONEREC, LASTREC : ELEMREC;
    KEY              : CHAR;
    COMPTYPE        : COMPONENT;
    COMPNO          : BYTE;
    DELETED         : BOOLEAN;
BEGIN
    REPEAT
        WRITE ('***** TYPE OF ELEMENT TO BE DELETED PRESS [G,L,D] ONLY ONE *****');
        REPEAT KEY := UPCASE(READKEY);
        UNTIL KEY IN ['G','L','D']; write(key); writeln;
        CASE KEY OF
            'G': COMPTYPE := GN;
            'L': COMPTYPE := LN;
            'D': COMPTYPE := LD;
        END;
        SEEK(FI,FileSize(FI)-1);
        READ(FI,LASTREC);
        WRITE(' DELETED ELEMENT NO = '); READLN(COMPNO);
        DELETED := FALSE; SEEK (FI,0);
        WHILE NOT EOF(FI) AND NOT DELETED DO
            BEGIN
                READ(FI,ONEREC);

```

```

        WITH ONEREC DO
        BEGIN
            IF (ETYPE=COMPTYPE) AND (ENO=COMPNO) THEN
            BEGIN
                DELETED := TRUE;
                SEEK(FI,FilePos(FI)-1);
                WRITE(FI,LASTREC);
                SEEK(FI,FileSize(FI)-1);
                TRUNCATE(FI);
                WRITELN('FINISH DELETS');
            END;
        END;
    END;
    WRITELN('##### FINISHED DELETEDATA Yes OR No PRESS [Y,N] ##### ');
    REPEAT KEY := UPCASE(READKEY);
    UNTIL KEY IN {'Y','N'};
    UNTIL KEY = 'Y';
    CLOSE(FI);
END;
BEGIN [ main procedure DATANETRY]
MODE := TRUE;
REPEAT
WRITE('ENTER TEST SYSTEM NAME STRING[8] = ');READLN (SYSNAME);
ASSIGN(ELEMFI,SYSNAME);
[SI-] RESET(ELEMFI); [SI+]
IF IOresult <> 0 THEN
BEGIN
    WRITELN('### THIS "',SYSNAME,'" IS NEW FILE OF TEST SYSTEM PRESS [Y,N] ###');
    REPEAT KEY1 := UPCASE(READKEY);
    UNTIL KEY1 IN {'Y','N'};
    IF KEY1 <> 'Y' THEN MODE := FALSE
    ELSE BEGIN
        MODE := TRUE;
        REWRITE(ELEMFI); CLOSE(ELEMFI); RESET(ELEMFI);
    END;
END
ELSE MODE := TRUE;
UNTIL MODE = TRUE;

REPEAT
WRITELN('***** SELECT MODE OF DATAENTRY E_nter C_orrect D_lete AND Q_uit *****');
WRITELN('PRESS [E,C,D,Q] ONLY ONE'); KEY2 := UPCASE(READKEY);
    IF KEY2 = 'E' THEN BEGIN ENTERDATA (ELEMFI); WRITELN; END
    ELSE IF KEY2 = 'C' THEN BEGIN CORRECTDATA(ELEMFI); WRITELN; END
    ELSE IF KEY2 = 'D' THEN BEGIN DELETEDATA (ELEMFI); WRITELN; END;
WRITELN('##### ARE YOU SURE TO FINISH DATAENTRY Yes OR No PRESS [Y,N] ##### ');
REPEAT KEY1 := UPCASE(READKEY);
UNTIL KEY1 IN {'Y','N'};
IF KEY1 = 'Y' THEN MODE:= FALSE
ELSE BEGIN RESET(ELEMFI); MODE:= TRUE; END;
UNTIL MODE = FALSE;
WRITELN('*****FINISH DATAENTRY *****');
END;

PROCEDURE READDATA;
VAR
    TESTFI      : ELEMTYPE;
    TESTREC     : ELEMREC;
    TESTNAME    : STRING[8];
    MODE        : BOOLEAN;
    M           : BYTE;      TGEN : BYTE;
    KRV        : CHAR;

```

```

BEGIN
  MODE := TRUE;
  REPEAT
  WRITE('READ TEST SYSTEM NAME STRING[8] = '); READLN(TESTNAME);
  ASSIGN(TESTFI,TESTNAME);
  [SI-] RESET(TESTFI); [SI+]
  IF IOresult <> 0 THEN
  BEGIN
    WRITELN('RETRY THE RIGHT NAME OF TEST SYSTEM AGAIN');
    MODE := FALSE;
  END
  ELSE MODE := TRUE;
  UNTIL MODE = TRUE;
  TN_GEN:=0; TN_LINE:=0; TN_LOAD:=0; TGEN := 0;
  WHILE NOT EOF(TESTFI) DO
  BEGIN
    READ(TESTFI,TESTREC);
    WITH TESTREC DO
    BEGIN
      CASE ETYPE OF
        GN: BEGIN Inc(TN_GEN);
              GENNO [TN_GEN] := ENO;
              GENCAP[TN_GEN] := ECAP;
              GENB [TN_GEN] := GNBUS;
              FRG [TN_GEN] := GNFR;
              RRG [TN_GEN] := GNRR;
              GIDEN [TN_GEN] := GNID; TGEN:=TGEN+GNID;
            END;
        LN: BEGIN Inc(TN_LINE);
              LINENO [TN_LINE] := ENO;
              LINECAP[TN_LINE] := ECAP;
              SB [TN_LINE] := LSDB;
              EB [TN_LINE] := LEDB;
              FRL [TN_LINE] := LNFR;
              RRL [TN_LINE] := LNRR;
            END;
        LD: BEGIN Inc(TN_LOAD);
              LOADNO [TN_LOAD] := ENO;
              LOADCAP[TN_LOAD] := ECAP;
              LOADB [TN_LOAD] := LDBUS;
            END;
      END;
    END;
  END;
  CLOSE(TESTFI);
  TN_BUS:=0;
  FOR M:= 1 TO TN_GEN DO
  BEGIN IF GENB[M] > TN_BUS THEN TN_BUS := GENB[M]; END;
  FOR M:= 1 TO TN_LOAD DO
  BEGIN IF LOADB[M]> TN_BUS THEN TN_BUS := LOADB[M]; END;
  FOR M:= 1 TO TN_LINE DO
  BEGIN
    IF SB[M]> TN_BUS THEN TN_BUS := SB[M];
    IF EB[M]> TN_BUS THEN TN_BUS := EB[M];
  END;
  WRITELN(' TOTAL BUSSES IN TEST SYSTEM = ',TN_BUS);
  WRITELN('If total busses in test system is INCORRECT. " quit AND go to DATAENTRY "');
  WRITELN;
  WRITELN('##### Would you like to see TEST SYSTEM DATAFILE PRESS (Y,N) #####');
  REPEAT KEY := UPCASE(READKEY); UNTIL KEY IN ['Y','N'];
  IF KEY = 'Y' THEN

```

```

BEGIN
  REPEAT
    WRITELN('***** TYPE OF ELEMENT TO BE SHOWN PRESS [G,L,D,A_1] *****');
    REPEAT KEY := UPCASE(READKEY); UNTIL KEY IN ['G','L','D','A'];
    IF (KEY = 'G') OR (KEY = 'A') THEN
      BEGIN
        WRITELN ('                      GENERATOR DATA ',TN_GEN:3,' ITEMS ',TGEN:3,' UNITS ');
        WRITELN (' |-----|');
        WRITELN (' |          GEN NO.GEN IDN  GEN CAP (MW)  GEN BUS  FAILURE RATE  REPAIR RATE |');
        WRITELN (' |-----|');
        FOR M:= 1 TO TN_GEN DO
          WRITELN (' |',M:4,GENNO[M]:6,GIDEN[M]:6,GENCAP[M]:12:3,GENB[M]:10,FRG[M]:15:3,RRG[M]:15:3, ' |');
          WRITELN (' |-----|');
        END;
        IF (KEY = 'L') OR (KEY = 'A') THEN
          BEGIN
            WRITELN;
            WRITELN ('                      LINE DATA ',TN_LINE:3,' UNITS ');
            WRITELN (' |-----|');
            WRITELN (' |          LINE NO. LINE CAP(MW) SD_BUS  ED_BUS  FAILURE_RATE  REPAIR_RATE |');
            WRITELN (' |-----|');
            FOR M:= 1 TO TN_LINE DO
              WRITELN (' |',M:4,LINENO[M]:6,LINECAP[M]:12:3,SB[M]:10,EB[M]:8,FRL[M]:12:3,RRL[M]:12:3, ' |');
              WRITELN (' |-----|');
            END;
            IF (KEY = 'D') OR (KEY = 'A') THEN
              BEGIN
                WRITELN;
                WRITELN ('                      LOAD DATA ',TN_LOAD:3,' BUSSES ');
                WRITELN (' |-----|');
                WRITELN (' |          LOAD NO.  LOAD DEMAND(MW)  LOAD BUS |');
                WRITELN (' |-----|');
                FOR M:= 1 TO TN_LOAD DO
                  WRITELN (' |',M:4,LOADNO[M]:6,LOADCAP[M]:15:3,LOADB[M]:15, ' |');
                  WRITELN (' |-----|');
                END;
                END;
                WRITELN('##### Would you like to see more PRESS [Y N] #####');
                REPEAT KEY := UPCASE(READKEY); UNTIL KEY IN ['Y','N'];
                IF KEY = 'Y' THEN MODE := TRUE
                ELSE MODE :=FALSE;
                UNTIL MODE = FALSE;
            END;
            WRITELN (' FINISH DISPLAY TEST SYSTEM DATA ');
          END;
        END;
      END;
    END;
  END;

PROCEDURE MODFAREA ; {ARRANGE NEW BUS NUMBER IN MS & SS AREA}
VAR
  I,J,L,M,N : BYTE;
  KEYM,KEY : CHAR;
  Procedure TRANSFORM(AMOUNT : BYTE;
                     VAR OLDB : BUS ;
                     VAR G_BUS: MSSS;
                     VAR NEWB : BUS );

  VAR II,JJ,KK : BYTE;
  BEGIN {MAIN sub_procedure transform}
    FOR KK := 1 TO AMOUNT DO
      BEGIN
        IF OLDB[KK] IN MS_BUS THEN
          BEGIN
            II := 0; G_BUS[KK] := MS;
            REPEAT II := II+1;

```

```

        UNTIL MSB[II] = OLDB[KK];
        NEWB[KK] := II;
        END
        ELSE
        BEGIN
        JJ := 0; G_BUS[KK] := SS;
        REPEAT JJ := JJ+1;
        UNTIL SSB[JJ] = OLDB[KK];
        NEWB[KK] := JJ;
        END;
    END;
END;
BEGIN {MAIN procedure MODFAREA}
    WRITELN('          TOTAL BUSSES IN TEST SYSTEM = ',TN_BUS);
    MS_BUS := []; SS_BUS := [];
    WRITE('***** Find adequacy indices by ADEQUACY EQUIVALENTS PRESS [Y N] *****');
    REPEAT KEYM := UPCASE(READKEY); UNTIL KEYM IN ['Y','N']; WRITELN(KEYM);
    IF KEYM = 'N' THEN
    BEGIN
        MS_BUS :=[];          TMS_BUS := 0;
        SS_BUS :=[1..TN_BUS]; TSS_BUS := TN_BUS;
        INDEXT := FALSE; INTSET:=[];
        WRITELN('    HLII RELIABILITY EVALUATION by CONVENTIONAL METHOD');
        WRITE('*****INPUT NUMBER of BUS IN AREA of INTEREST = '); READLN(J);
        IF J<>0 THEN
        BEGIN
            FOR I := 1 TO J DO
            BEGIN
                REPEAT
                WRITE('INPUT',I:3,'Bus No.in AREA of INTEREST = '); READLN(M);
                UNTIL(M>0) AND (M<=TSS_BUS) AND (NOT(M IN INTSET));
                INTSET:=INTSET+[M];
            END;
        END;
    END
    ELSE
    BEGIN
        WRITELN('    HLII RELIABILITY EVALUATION by ADEQUACY EQUIVALENT METHOD');
        INDEXT:=TRUE;
        REPEAT
        WRITE('          Input amount of busses in SUBSYSTEM SS = '); READLN(TSS_BUS);
        UNTIL (TSS_BUS>0) AND (TSS_BUS<TN_BUS);
        FOR I := 1 TO TSS_BUS DO
        BEGIN
            REPEAT
            WRITE('Input ',I:3,' bus no. in SS = '); READLN(J);
            UNTIL (J>0) AND (J<=TN_BUS) AND (NOT (J IN SS_BUS));
            SS_BUS := SS_BUS +[J];
        END;
        MS_BUS := [1..TN_BUS] - SS_BUS;
        TMS_BUS := TN_BUS - TSS_BUS;
    END;
END;

L := 0; M := 0;
FOR N := 1 TO TN_BUS DO
BEGIN
    IF N IN MS_BUS THEN
    BEGIN L:=L+1; MSB[L]:=N; END
    ELSE BEGIN M:=M+1; SSB[M]:=N; END;
END;
TRANSFORM(TN_LINE, SB, G_SB, NSB);

```

```

TRANSFORM(TN_LINE, EB, G_EB, NEB);
TRANSFORM(TN_GEN, GENB, G_GENB, NGENB );
TRANSFORM(TN_LOAD, LOADB, G_LOADB, NLOADB);

[*****constructed full capacity model of MS*****]
PBG1:=1; PBL1:=1; PBC1:=1;
DNG1:=0; DNL1:=0; DNC1:=0;
NOG1:=0; NOL1:=0; NOC1:=0;
SETG1:={}; SETL1:={}; SETC1:={}; SETCMS:={};
PBG2:=1; PBL2:=1;
DNG2:=0; DNL2:=0;
NOG2:=0; NOL2:=0;
SETG2:={}; SETL2:={};          SETCSS:={};
GENMS:=0; GENSS:=0;

FOR I := 1 TO MAXBUS DO [clear matrix MSMATRIX first]
BEGIN
FOR J := 1 TO MAXBUS DO BEGIN MSMAT[I,J]:=0.0; SSMAT[I,J]:=0.0; END;
END;

IF TN_GEN <> 0 THEN
BEGIN
FOR I := 1 TO TN_GEN DO [caculate all generation capacity at any bus in MS]
BEGIN
PBG[I]:=RRG[I]/(RRG[I]+FRG[I]);
IF G_GENB[I] = MS THEN
BEGIN
NOG1 :=NOG1+1;    GENMS:=GENMS+GIDEN[I];
SETG1 :=SETG1+[I];  PBG1 :=PBG1*Power(PBG[I],GIDEN[I]);  DNG1 :=DNG1+FRG[I]*GIDEN[I];
MSMAT[1,NGENB[I]+1] := MSMAT[1,NGENB[I]+1]+GENCAP[I]*GIDEN[I];
END
ELSE BEGIN
NOG2 :=NOG2+1;    GENSS:=GENSS+GIDEN[I];
SETG2 :=SETG2+[I];  PBG2 :=PBG2*Power(PBG[I],GIDEN[I]);  DNG2 :=DNG2+FRG[I]*GIDEN[I];
SSMAT[1,NGENB[I]+1] := SSMAT[1,NGENB[I]+1]+GENCAP[I]*GIDEN[I];
END;
END;
END;

FOR I:= 1 TO TN_LOAD DO [calculate all load capacity at any bus in MS]
BEGIN
IF G_LOADB[I] = MS THEN
MSMAT[1+NLOADB[I],TMS_BUS+2]:=MSMAT[1+NLOADB[I],TMS_BUS+2]+LOADCAP[I]
ELSE SSMAT[1+NLOADB[I],TSS_BUS+2]:=SSMAT[1+NLOADB[I],TSS_BUS+2]+LOADCAP[I];
END;

IF TN_LINE <> 0 THEN
BEGIN
FOR I:= 1 TO TN_LINE DO [calculate all line capacity]
BEGIN
PBL[I]:=RRL[I]/(RRL[I]+FRL[I]);
IF (G_SB[I]=MS) AND (G_EB[I]=MS) THEN
BEGIN
NOL1 :=NOL1+1;
SETL1 :=SETL1+[I];  PBL1 :=PBL1*PBL[I];  DNL1 :=DNL1+FRL[I];
IF NSB[I] < NEB[I] THEN
MSMAT[NSB[I]+1,NEB[I]+1]:=MSMAT[NSB[I]+1,NEB[I]+1]+LINECAP[I]
ELSE
MSMAT[NEB[I]+1,NSB[I]+1]:=MSMAT[NEB[I]+1,NSB[I]+1]+LINECAP[I];
END
ELSE IF (G_SB[I]= MS) AND (G_EB[I]= SS) THEN

```



```

BEGIN NOC1 :=NOC1+1;
MSMAT[NSB[I]+1,TMS_BUS+3]:=MSMAT[NSB[I]+1,TMS_BUS+3]+LINECAP[I];
SETC1 :=SETC1+[I]; PBC1 :=PBC1+PBL[I]; DNC1 :=DNC1+FRL[I];
SETCMS:=SETCMS + [NSB[I]]; SETCSS:=SETCSS+[NEB[I]];
END
ELSE IF (G_SB[I]= SS) AND (G_EB[I]= MS) THEN
BEGIN NOC1 :=NOC1+1;
MSMAT[NEB[I]+1,TMS_BUS+3]:=MSMAT[NEB[I]+1,TMS_BUS+3]+LINECAP[I];
SETC1 :=SETC1+[I]; PBC1 :=PBC1+PBL[I]; DNC1 :=DNC1+FRL[I];
SETCMS:=SETCMS + [NEB[I]]; SETCSS:=SETCSS+[NSB[I]];
END
ELSE BEGIN
NOL2 :=NOL2+1;
SETL2 :=SETL2+[I]; PBL2 :=PBL2+PBL[I]; DNL2 :=DNL2+FRL[I];
IF NSB[I] < NEB[I] THEN
SSMAT[NSB[I]+1,NEB[I]+1]:=SSMAT[NSB[I]+1,NEB[I]+1]+LINECAP[I]
ELSE
SSMAT[NEB[I]+1,NSB[I]+1]:=SSMAT[NEB[I]+1,NSB[I]+1]+LINECAP[I];
END;
END;
END;
writeln;
write('NOG1 = ',nog1:2,' ');WRITE('GEN IN MS = ',GENMS:2,' ');
write('NOL1 = ',nol1:2,' ');writeln;
write('NOG2 = ',nog2:2,' ');WRITE('GEN IN SS = ',GENSS:2,' ');
write('NOL2 = ',nol2:2,' ');writeln;

PBMS :=PBG1+PBL1*PBC1;      writeln('pbms = ',pbms);
UPMS :=0;                  writeln('upms = ',upms);
DNMS :=DNG1+DNL1+DNC1;     writeln('dnms = ',dnms);
FQMS :=PBMS*DNMS;          writeln('fqms = ',fqms);
PBSS :=PBG2+PBL2;          writeln('pbss = ',pbss);
UPSS :=0;                  writeln('upss = ',upss);
DNSS :=DNG2+DNL2;          writeln('dnss = ',dnss);
FQSS :=PBSS*DNSS;          writeln('fqss = ',fqss);
WRITELN(' PRESS [C] to continue ');
REPEAT KEY := UPCASE(READKEY); UNTIL KEY ='C';

J := 0;
FOR I:=1 TO TMS_BUS DO
BEGIN IF I IN SETCMS THEN BEGIN J:=J+1; MSAB[J]:=I;writeln('MSAB['',J:2,''] =',msb[I]:2); END; END;
TMSAB := J;
J := 0;
FOR I:=1 TO TSS_BUS DO
BEGIN IF I IN SETCSS THEN BEGIN J:=J+1; SSAB[J]:=I;writeln('SSAB['',J:2,''] =',ssb[I]:2); END; END;
TSSAB := J;
WRITELN('##### Would you like to see MSMAT & SSMAT or not? PRESS [Y N] #####');
REPEAT KEY := UPCASE(READKEY); UNTIL KEY IN ['Y','N'];
IF KEY = 'Y' THEN
BEGIN
IF KEYM = 'Y' THEN
BEGIN
WRITELN('***** MSMAT DATA ***** ');
FOR I := 1 TO TMS_BUS+2 DO
BEGIN
FOR J :=1 TO TMS_BUS+3 DO
WRITE(MSMAT[I,J]:7:2,' ');WRITELN;
END;
END;
WRITELN(' PRESS [C] to continue ');
REPEAT KEY := UPCASE(READKEY); UNTIL KEY ='C';

```

```

WRITELN('***** SSMAT DATA ***** ');
FOR I := 1 TO TSS_BUS+2 DO
  BEGIN
    FOR J :=1 TO TSS_BUS+3 DO
      WRITE(SSMAT[I,J]:7:2, ' ');WRITELN;
    END;
  END;
WRITELN('Would you like to finish? PRESS [Y] to EXIT');
REPEAT KEY := UPCASE(READKEY); UNTIL KEY = 'Y';
END;

PROCEDURE ROUNDING (   ALD: REAL; [all connected lines capacities]
                     TMSAB1: BYTE; [total number of connected bus in MS]
                     RD: REAL; [increment in each step 5.0 or10.0 ]
                     VAR SLD: AQE; [marginal line capacity before rounding]
                     VAR RLD: AQE); [marginal line capacity after rounding]

VAR
  CKSET   : SET OF 1..10;
  CKMAX   : REAL;
  A,B,E   : REAL;
  C,D     : ARRAY[1..10] OF REAL;
  M,N     : BYTE;
  RANK    : ARRAY[1..10] OF BYTE;

BEGIN
  A:=0.0; B:=0.0;
  FOR M := 1 TO TMSAB1 DO
    BEGIN
      A:= A+Abs(Round(SLD[M]/RD));
      B:= B+Abs(SLD[M]);
      C[M]:=Abs(Trunc(SLD[M]/RD));
      D[M]:=Abs(Frac(SLD[M]/RD));
    END;
    IF (A = Round(B/RD)) AND (A<=Round(ALD/RD)) THEN
      BEGIN
        FOR M:= 1 TO TMSAB1 DO RLD[M] := RD*(Round(SLD[M]/RD));
      END
    ELSE
      BEGIN
        CKSET :=[1..TMSAB1]; E:=0.0;
        FOR M:= 1 TO TMSAB1 DO
          BEGIN
            CKMAX:= 0.0;
            FOR N:= 1 TO TMSAB1 DO
              BEGIN
                IF (N IN CKSET) AND (D[N]>=CKMAX) THEN
                  BEGIN RANK[M]:= N; CKMAX:=D[N] END;
              END;
            CKSET := CKSET-[RANK[M]];
          END;
        FOR M:= 1 TO TMSAB1 DO E:= E+C[M];
        M:=0;
        REPEAT M:=M+1;
          IF SLD[RANK[M]]<0.0 THEN RLD[RANK[M]]:=- (C[RANK[M]]+1)*RD
          ELSE
            RLD[RANK[M]]:= (C[RANK[M]]+1)*RD;
          UNTIL M=(Round(B/RD)-E);
          REPEAT M:=M+1;
            IF SLD[RANK[M]]<0.0 THEN RLD[RANK[M]]:= -C[RANK[M]]*RD
            ELSE
              RLD[RANK[M]]:= C[RANK[M]]*RD;
            UNTIL M = TMSAB1;
          END;
        END;
      END;
END;

```

```

Procedure CTG(R:BYTE ; MS_SS:GROUP ; SMS_SS:SETNO ; TMS_SS:BYTE ;
              VAR CTL1,CTL2:SETNO );
VAR      K,L,M      : BYTE;
          CTLOAD,CTLD : SETNO;
          BSET ,LSET  : SETNO;

BEGIN
  L:=0;
  REPEAT L:=L+1;
  UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NGENB[R]));
  IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NGENB[R])) THEN
    BEGIN CTLOAD := [L]; BSET :=[NGENB[R]]; END
  ELSE BEGIN CTLOAD := []; BSET :=[NGENB[R]]; END;
  LSET := [];
  FOR K:= 1 TO TN_LINE DO
  BEGIN
    IF (K IN SMS_SS)AND((NSB[K]=NGENB[R])OR(NEB[K]=NGENB[R])) THEN
    BEGIN
      IF NSB[K]=NGENB[R] THEN
        BEGIN BSET:=BSET+[NEB[K]]; LSET:=LSET+[K]; END
      ELSE BEGIN BSET:=BSET+[NSB[K]]; LSET:=LSET+[K]; END;
      L:=0;
      REPEAT L:=L+1;
      UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
      IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])) THEN
      BEGIN
        IF NOT(L IN CTLOAD) THEN CTLOAD := CTLOAD + [L];
      END;
    END;
  END;
  CTLD:=[];
  FOR M := 1 TO TMS_SS DO
  BEGIN
    IF M IN BSET THEN
    BEGIN
      FOR K:= 1 TO TN_LINE DO
      BEGIN
        IF (K IN SMS_SS)AND(((NSB[K]=M)AND NOT(NEB[K] IN BSET) AND NOT(K IN LSET))OR
                              ((NEB[K]=M)AND NOT(NSB[K] IN BSET) AND NOT(K IN LSET))) THEN
          BEGIN
            L:=0;
            REPEAT L:=L+1;
            UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
            IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])) THEN
            BEGIN
              IF NOT(L IN (CTLOAD+CTLD)) THEN CTLD := CTLD + [L];
            END;
          END;
        END;
      END;
    END;
  END;
  CTL1:=CTL1+CTLOAD;
  CTL2:=CTL2+CTLD;
  [writeln('exitCTG');]
END;

Procedure CTL(T:BYTE ;MS_SS:GROUP ;SMS_SS:SETNO ;TMS_SS:BYTE ;
              VAR CTL3,CTL4,CTL5,CTL6:SETNO);
VAR      K,L,M      : BYTE;
          CTLOAD1,CTLOAD2: SETNO;

```

```

CTLD1,CTLD2   : SETNO;
BESET,BSSET   : SETNO;
LESET,LSSET   : SETNO;

```

```

BEGIN
  L:=0;
  REPEAT L:=L+1;
  UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NEB[T]));
  IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NEB[T])) THEN
    BEGIN CTLOAD1 := [L]; BESET:=[NEB[T]]; END
  ELSE BEGIN CTLOAD1 := []; BESET:=[NEB[T]]; END;
  L:=0;
  REPEAT L:=L+1;
  UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NSB[T]));
  IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND(NLOADB[L]=NSB[T])) THEN
    BEGIN CTLOAD2 := [L]; BSSET:=[NSB[T]]; END
  ELSE BEGIN CTLOAD2 := []; BSSET:=[NSB[T]]; END;

  LESET:=[]; LSSET:=[];
  FOR K:= 1 TO TN_LINE DO
  BEGIN
    IF (K IN SMS_SS)AND(K<>T)AND((NSB[K]=NEB[T])OR(NEB[K]=NEB[T])) THEN
    BEGIN
      IF NSB[K]=NEB[T] THEN
      BEGIN
        IF NOT(NEB[K] IN BSSET) AND NOT(K IN LSSET) THEN
        BEGIN BESET:=BESET+[NEB[K]]; LESET:=LESET+[K]; END;
      END
      ELSE
      BEGIN
        IF NOT(NSB[K] IN BSSET) AND NOT(K IN LSSET) THEN
        BEGIN BESET:=BESET+[NSB[K]]; LESET:=LESET+[K]; END;
      END;
      L:=0;
      REPEAT L:=L+1;
      UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
      IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K]))) THEN
      BEGIN
        IF NOT(L IN CTLOAD1) THEN CTLOAD1 := CTLOAD1 + [L];
      END;
    END;
    IF (K IN SMS_SS)AND(K<>T)AND((NSB[K]=NSB[T])OR(NEB[K]=NSB[T])) THEN
    BEGIN
      IF NSB[K]=NSB[T] THEN
      BEGIN
        IF NOT(NEB[K] IN BESET) AND NOT(K IN LESET) THEN
        BEGIN BSSET:=BSSET+[NEB[K]]; LSSET:=LSSET+[K]; END;
      END
      ELSE
      BEGIN
        IF NOT(NSB[K] IN BESET) AND NOT(K IN LESET) THEN
        BEGIN BSSET:=BSSET+[NSB[K]]; LSSET:=LSSET+[K]; END;
      END;
      L:=0;
      REPEAT L:=L+1;
      UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
      IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K]))) THEN
      BEGIN
        IF NOT(L IN CTLOAD2) THEN CTLOAD2:= CTLOAD2+ [L];
      END;
    END;
  END;

```

```

END;
CTLD1:=[]; CTLD2:=[];
[ FOR M:= 1 TO TMS_SS DO
BEGIN
  IF M IN BESET THEN
  BEGIN
    FOR K := 1 TO TN_LINE DO
    BEGIN
      IF (K IN SMS_SS)AND(K<>T)AND(((NSB[K]=M)AND NOT(NEB[K] IN (BESET+BSSET))AND
      NOT(K IN (LESET+LSSET))) OR ((NEB[K]=M)AND NOT(NSB[K] IN (BESET+BSSET))AND
      NOT(K IN (LESET+LSSET)))) THEN
      BEGIN
        L:=0;
        REPEAT L:=L+1;
        UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
        IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])) THEN
        BEGIN
          IF NOT(L IN (CTLOAD1+CTLD1)) THEN CTLD1 := CTLD1 + [L];
        END;
        END;
        END;
      END;
    IF M IN BSSET THEN
    BEGIN
      FOR K := 1 TO TN_LINE DO
      BEGIN
        IF (K IN SMS_SS)AND(K<>T)AND(((NSB[K]=M)AND NOT(NEB[K] IN (BESET+BSSET))AND
        NOT(K IN (LESET+LSSET))) OR ((NEB[K]=M)AND NOT(NSB[K] IN (BESET+BSSET))AND
        NOT(K IN (LESET+LSSET)))) THEN
        BEGIN
          L:=0;
          REPEAT L:=L+1;
          UNTIL (L=TN_LOAD)OR((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])));
          IF (L<=TN_LOAD)AND((G_LOADB[L]=MS_SS)AND((NLOADB[L]=NEB[K])OR(NLOADB[L]=NSB[K])) THEN
          BEGIN
            IF NOT(L IN (CTLOAD2+CTLD2)) THEN CTLD2 := CTLD2 + [L];
          END;
          END;
          END;
        END;
      END;
    END;
    [ write('exitCTL ');]
    CTL3:=CTL3+CTLOAD1; CTL5:=CTL5+CTLD1;
    CTL4:=CTL4+CTLOAD2; CTL6:=CTL6+CTLD2;
  END;

  Procedure TIMECHECK(VAR TIMESEC:LONGINT);
  var y,m,dt,d,h,mn,s,f : word;
  begin
    getdate(y,m,dt,d);
    gettime(h,mn,s,f);
    timesec:=dt*24*60*60 + h*60*60 + mn*60 + s;
  end;
END.

```

```

[This program will test MS ADEQUACY MODEL update 28_9_91 ]
UNIT MSCAL9;
INTERFACE
USES      DOS,CRT,PRINTER,ALLDATA;

PROCEDURE MSMODEL(VAR AQEQ1:ARRZ; {aquequacy equiv in each state of MS conting|
                  VAR AQPBI:ARRX; {probability   in each state of MS conting|
                  VAR AQUP1:ARRX; {departure to up in each state of MS conting|
                  VAR AQDN1:ARRX; {departure to dn in each state of MS conting|
                  VAR TAQST1:LONGINT; {total state of ADQ model before rearrange|
                  VAR AQLINK1:ARRW);

IMPLEMENTATION

PROCEDURE MSMODEL(VAR AQEQ1:ARRZ; {aquequacy equiv in each state of MS conting|
                  VAR AQPBI:ARRX; {probability   in each state of MS conting|
                  VAR AQUP1:ARRX; {departure to up in each state of MS conting|
                  VAR AQDN1:ARRX; {departure to dn in each state of MS conting|
                  VAR TAQST1:LONGINT; {total state of ADQ model before rearrange|
                  VAR AQLINK1:ARRW);

VAR      CLEV ,LLEV ,GLEV      ,TGLEV      : CTGLEVEL;
         CST ,LNST ,GST        : LONGINT;
         COUT1,LOUT1,GOUT1     ,TSETG1     : SETNO;
         R,S,T,U,K,L,N        ,TNOG1     : BYTE;
         PB,UP,DN             ,TPB,TUP,TDN : REAL;
         TOTALPB,TOTALFQ      : REAL;
         NET                  ,TNET       : ARRY;
         AC                   : NODE;
         ST                   : LONGINT;
         MTP                  : WORD;
         CTL1,CTL2,CTL3,CTL4,CTL5,CTL6    : SETNO;

PROCEDURE MSSTATE(VAR MSNET : ARRY; VAR AVAILCAP : NODE; TMS_BUS : BYTE);
VAR DUMMY                : BOOLEAN;
    LOAD_MSB,GEN_MSB     : NODE;
    FGEN,SLOAD          : NODE;
    I                   : BYTE;

    MAXLOAD,MAXGEN,AVALOAD,EXPLOAD: REAL;
    SBYG,SBYL,SBYGL     : SETNO;
    BYG, BYL, BYGL      : REAL;
    MAXLOAD1            : REAL;
    TM1,TM2             : REAL;
    SETLDCT ,TSET       : SETNO;
    FLOWIN,FLOWOUT      : REAL;
    msgen,ssgen,msload,ssload : real;
    noctb,mscurt,gbml,gbal,gbcurt : real;
    ck1,ck2             : boolean;

BEGIN {main procedure MSSTATE}
  DUMMY := TRUE;
  MAXLOAD:=0; BYG:=0; BYL:=0; BYGL:=0;
  SBYG:=[]; SBYL:=[]; SBYGL:=[];
  FOR I:= 1-TO TMS_BUS DO
  BEGIN
    MAXLOAD :=MAXLOAD+MSNET[I+1,TMS_BUS+2];
    LOAD_MSB[I] :=MSNET[I+1,TMS_BUS+2];
    AVAILCAP[I] :=MSNET[I+1,TMS_BUS+3];
  END;
  FOR I:= 1 TO TN_LOAD DO
  BEGIN

```

```

IF I IN (CTL1) THEN
BEGIN BYG := BYG+LOADCAP[I]; SBYG := SBYG+[NLOADB[I]]; END;
IF I IN (CTL3+CTL4) THEN
BEGIN BYL := BYL+LOADCAP[I]; SBYL := SBYL+[NLOADB[I]]; END;
IF I IN (CTL1+CTL3+CTL4) THEN
BEGIN BYGL:=BYGL+LOADCAP[I]; SBYGL:=SBYGL+[NLOADB[I]]; END;
END;
msgen:=0; msload:=0; ssgen:=0; ssload:=0; maxgen:=0;
for i:= 1 to tms_bus do
begin
msgen := msgen + msmat[1,i+1];
msload:=msload + msnet[i+1,tms_bus+2];
maxgen:=maxgen + msnet[1,i+1];
end;
for i:= 1 to tss_bus do
begin
ssgen := ssgen + ssmat[1,i+1];
ssload:= ssload+ ssmat[i+1,tss_bus+2];
end;
if (maxgen+ssgen)<(msload+ssload) then
begin
ck1:=false; ck2:=false; noctb:=0;
mscurt:=(msload+ssload)-(maxgen+ssgen);
if (mscurt>curt/100*msload) and (mscurt>curt/100*(msload+ssload)) then
begin
ck1 := true;
mscurt:= mscurt*msload/(msload+ssload);
for i := 1 to tms_bus do
begin
if not(i in sbygl) then
begin
msnet[i+1,tms_bus+2]:=load_msb[i]-mscurt*load_msb[i]/msload;
noctb:=noctb+msnet[i+1,tms_bus+2];
end;
end;
end;
end
else if (mscurt)=curt/100*msload) and (mscurt<=curt/100*(msload+ssload)) then
begin
ck2 := true;
mscurt:= curt/100*msload;
for i := 1 to tms_bus do
begin
if not(i in sbygl) then
begin
msnet[i+1,tms_bus+2]:=(1-curt/100)*load_msb[i];
noctb:=noctb+msnet[i+1,tms_bus+2];
end;
end;
end;
end;
end;
gbml:=0; gbal:=0; maxload1:=0; setldct:=[];
for i := 1 to tms_bus do
begin
if(i in sbygl)and(msnet[1,i+1]>0)and(msnet[i+1,tms_bus+2]>0) then
begin
tm1:= msnet[i+1,tms_bus+2]-mscurt*load_msb[i]/bygl;
tm2:= (1-curt/100)*load_msb[i];
if msnet[1,i+1]>msnet[i+1,tms_bus+2] then
begin
if tm1<=tm2 then msnet[i+1,tms_bus+2]:=tm2
else
msnet[i+1,tms_bus+2]:=tm1;
end
end
end

```

```

else begin
    if tml<=msnet[1,i+1] then msnet[i+1,tms_bus+2]:=msnet[1,i+1]
    else
        msnet[i+1,tms_bus+2]:=tml;
    end;
    setldct:=setldct+[i];
    gbml :=gbml+load_msb[i];
    gbal :=gbal+msnet[i+1,tms_bus+2];
end;
if (i in sbygl) and not(i in setldct) then maxload1:=maxload1+load_msb[i];
end;
gbcurt:=gbml-gbal;
if maxload1<>0 then
begin
    if (mscurt-gbcurt)>=curt/100*maxload1 then
    begin
        if (ck1=true) or (ck2=true) then
        begin
            for i := 1 to tms_bus do
            begin
                if (i in sbygl)and not(i in setldct) then
                    msnet[i+1,tms_bus+2]:=((msload-mscurt)-(noctb+gbal))*load_msb[i]/maxload1;
                end;
            end
        else begin
            for i := 1 to tms_bus do
            begin
                if (i in sbygl)and not(i in setldct) then
                    msnet[i+1,tms_bus+2]:=(1-curt/100)*load_msb[i]
                else if not(i in sbygl) then
                    msnet[i+1,tms_bus+2]:=((msload-mscurt)-(1-curt/100)*maxload1-gbal)*load_msb[i]
                    . (msload-maxload1-gbml);
                end;
            end;
        end
    else begin
        if (ck1=true) or (ck2=true) then
        begin
            for i := 1 to tms_bus do
            begin
                if (i in sbygl)and not(i in setldct) then
                    msnet[i+1,tms_bus+2]:=((msload-mscurt)-(noctb+gbal))*load_msb[i]/maxload1;
                end;
            end
        else begin
            for i:= 1 to tms_bus do
            begin
                if (i in sbygl)and not(i in setldct) then
                    msnet[i+1,tms_bus+2]:=load_msb[i]-(mscurt-gbcurt)*load_msb[i]/maxload1;
                end;
            end;
        end;
    end
end
else begin [maxload1=0]
    for i:= 1 to tms_bus do
    begin
        if not(i in sbygl) then
            msnet[i+1,tms_bus+2]:=load_msb[i]-(mscurt-gbcurt)*load_msb[i]/(msload-gbml);
        end;
    end;
end;
MAXLOAD:=0;
FOR I:= 1 TO TMS_BUS DO

```



```

        BEGIN
            MAXLOAD      :=MAXLOAD+MSNET[I+1,TMS_BUS+2];
            LOAD_MSB[I] :=MSNET[I+1,TMS_BUS+2];
        END;
    end;
MAXFLOW(MSNET,TMS_BUS,DUMMY);
EXPLOAD:=0; AVALOAD:=0;
FOR I := 1 TO TMS_BUS DO
    BEGIN
        EXPLOAD:=EXPLOAD+MSNET[I+1,TMS_BUS+2];
        AVALOAD:=AVALOAD+MSNET[TMS_BUS+2,I+1];
    END;
( IF Abs(EXPLOAD)-Abs(AVALOAD) > 1.0E-2 THEN
  WRITELN('IN FIRST STAGE " EXPLOAD <> AVALOAD " '); )
TM1:=0;
FOR I := 1 TO TMS_BUS DO
    BEGIN
        GEN_MSB[I]      := MSNET[1,I+1];
        MSNET[I+1,TMS_BUS+2]:= LOAD_MSB[I];
        FGEN[I]         := MSNET[I+1,1];
        TM1             := TM1+AVAILCAP[I];
    END;
IF TM1 <> 0 THEN
    BEGIN
        IF AVALOAD < MAXLOAD THEN
            BEGIN
                FLOWIN := MAXLOAD-AVALOAD;
                IF TM1 > FLOWIN THEN
                    BEGIN
                        FOR I:= 1 TO TMS_BUS DO
                            BEGIN
                                IF AVAILCAP[I] <> 0 THEN
                                    MSNET[1,I+1] := MSNET[1,I+1] + AVAILCAP[I]*FLOWIN/TM1;
                                END;
                                DUMMY := FALSE;
                                MAXFLOW(MSNET,TMS_BUS,DUMMY);
                                TSET:=[]; TM2:=0;
                                FOR I := 1 TO TMS_BUS DO
                                    BEGIN
                                        IF (AVAILCAP[I] <> 0)AND(MSNET[I+1,1]>FGEN[I]) THEN
                                            BEGIN
                                                TSET := TSET+[I];
                                                TM2 := TM2 +(MSNET[I+1,1]-FGEN[I]);
                                            END;
                                        END;
                                    END;
                                IF (TM2 <> FLOWIN) AND (TM2 <> 0) THEN
                                    BEGIN
                                        TM1:=0;
                                        FOR I := 1 TO TMS_BUS DO
                                            BEGIN
                                                IF (AVAILCAP[I]<>0) AND (I IN TSET) THEN
                                                    TM1 := TM1+AVAILCAP[I];
                                                END;
                                            END;
                                        IF TM1 > FLOWIN THEN
                                            BEGIN
                                                FOR I:= 1 TO TMS_BUS DO
                                                    BEGIN
                                                        MSNET[1,I+1]:=GEN_MSB[I];
                                                        IF I IN TSET THEN
                                                            MSNET[1,I+1] := MSNET[1,I+1] + AVAILCAP[I]*FLOWIN/TM1;
                                                        END;
                                                    END;
                                                END;
                                            END;
                                        END;
                                    END;
                                END;
                            END;
                        END;
                    END;
                END;
            END;
        END;
    END;

```

```

        DUMMY := FALSE;
        MAXFLOW(MSNET,TMS_BUS,DUMMY);
        TM2:=0;
        FOR I := 1 TO TMS_BUS DO
        BEGIN
            IF (AVAILCAP[I] <> 0)AND(MSNET[I+1,1]>FGEN[I]) THEN
                TM2 := TM2 +(MSNET[I+1,1]-FGEN[I]);
            END;
        END;
        IF TM2 <> FLOWIN THEN
        BEGIN
            FOR I := 1 TO TMS_BUS DO
            BEGIN
                MSNET[1,I+1]:=GEN_MSB[I];
                IF AVAILCAP[I] <> 0 THEN MSNET[1,I+1]:=MSNET[1,I+1]+AVAILCAP[I];
            END;
            DUMMY := FALSE;
            MAXFLOW(MSNET,TMS_BUS,DUMMY);
        END;
    END;
END
ELSE
BEGIN
    FOR I := 1 TO TMS_BUS DO
    BEGIN
        IF AVAILCAP[I] <> 0 THEN MSNET[1,I+1]:=MSNET[1,I+1]+AVAILCAP[I];
    END;
    DUMMY := FALSE;
    MAXFLOW(MSNET,TMS_BUS,DUMMY);
END;
END;
AVALOAD:=0; MAXGEN:=0;
FOR I := 1 TO TMS_BUS DO
BEGIN
    IF AVAILCAP[I] <> 0.0 THEN
    BEGIN
        IF (FGEN[I]-MSNET[I+1,1])<>0.0 THEN
        BEGIN
            AVAILCAP[I] := FGEN[I]-MSNET[I+1,1];
            MSNET[1,I+1]:= GEN_MSB[I]-AVAILCAP[I];
        END
        ELSE MSNET[1,I+1]:= GEN_MSB[I];
    END;
    SLOAD[I] := MSNET[TMS_BUS+2,I+1];
    AVALOAD := AVALOAD + MSNET[TMS_BUS+2,I+1];
    MAXGEN := MAXGEN + MSNET[1,I+1];
END;
TM1:=0;
FOR I := 1 TO TMS_BUS DO
BEGIN
    IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2) THEN TM1 := TM1+AVAILCAP[I];
END;
IF (AVALOAD < MAXGEN) AND (TM1 <> 0) THEN
BEGIN
    FLOWOUT := MAXGEN-AVALOAD;
    IF TM1 > FLOWOUT THEN
    BEGIN
        FOR I:= 1 TO TMS_BUS DO
        BEGIN
            IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2) THEN
                MSNET[I+1,TMS_BUS+2]:=MSNET[I+1,TMS_BUS+2]+AVAILCAP[I]*FLOWOUT/TM1;
        END;
    END;
END;

```

```

END;
DUMMY := FALSE;
MAXFLOW(MSNET,TMS_BUS,DUMMY);
TSET:=[]; TM2:=0;
FOR I := 1 TO TMS_BUS DO
BEGIN
  IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2)AND(MSNET[TMS_BUS+2,I+1]>SLOAD[I]) THEN
  BEGIN
    TSET := TSET+[I];
    TM2 := TM2 + (MSNET[TMS_BUS+2,I+1]-SLOAD[I]);
  END;
END;
IF (TM2 <> FLOWOUT) AND (TM2 <> 0) THEN
BEGIN
  TM1:=0;
  FOR I := 1 TO TMS_BUS DO
  BEGIN
    IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2)AND(I IN TSET) THEN
    TM1 := TM1+AVAILCAP[I];
  END;
  IF TM1 > FLOWOUT THEN
  BEGIN
    FOR I:= 1 TO TMS_BUS DO
    BEGIN
      MSNET[I+1,TMS_BUS+2]:=LOAD_MSB[I];
      IF I IN TSET THEN
      MSNET[I+1,TMS_BUS+2]:=MSNET[I+1,TMS_BUS+2]+AVAILCAP[I]*FLOWOUT/TM1;
    END;
    DUMMY := FALSE;
    MAXFLOW(MSNET,TMS_BUS,DUMMY);
    TM2:=0;
    FOR I := 1 TO TMS_BUS DO
    BEGIN
      IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2)
      AND(MSNET[TMS_BUS+2,I+1]>SLOAD[I]) THEN
      TM2 := TM2 +(MSNET[TMS_BUS+2,I+1]-SLOAD[I]);
    END;
  END;
  IF TM2 <> FLOWOUT THEN
  BEGIN
    FOR I := 1 TO TMS_BUS DO
    BEGIN
      MSNET[I+1,TMS_BUS+2]:=LOAD_MSB[I];
      IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2) THEN
      MSNET[I+1,TMS_BUS+2]:=MSNET[I+1,TMS_BUS+2]+AVAILCAP[I];
    END;
    DUMMY := FALSE;
    MAXFLOW(MSNET,TMS_BUS,DUMMY);
  END;
END;
END;
ELSE
BEGIN
  FOR I := 1 TO TMS_BUS DO
  BEGIN
    IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2) THEN
    MSNET[I+1,TMS_BUS+2]:=MSNET[I+1,TMS_BUS+2]+AVAILCAP[I];
  END;
  DUMMY := FALSE;
  MAXFLOW(MSNET,TMS_BUS,DUMMY);
END;

```

```

END;

FOR I := 1 TO TMS_BUS DO
BEGIN
  IF (AVAILCAP[I]>0)AND(Abs(LOAD_MSB[I])-Abs(SLOAD[I])<1.0E-2) THEN
    AVAILCAP[I]:=MSNET[TMS_BUS+2,I+1]-SLOAD[I]
  ELSE IF (AVAILCAP[I]>0)AND(SLOAD[I]<LOAD_MSB[I]) THEN AVAILCAP[I]:=0;
END;
END;
END;

```

Procedure MSCAL;

```

VAR X,Y           : LONGINT;
    V             : BYTE;
    CK1,CK2       : BOOLEAN;
    TEMPB,TEMUP,TEMDN : REAL;
    KEY           : CHAR;
    RAQEQ         : AQE;
    TIE_CAP,ADD_CAP (RD) : REAL;

```

BEGIN

```

TOTALPB:= PBMS*PB;
TOTALFQ:= TOTALPB*(UP+(DNMS-DN));
IF (TOTALPB>1.0e-6)AND(TOTALFQ>1.0e-5) THEN
BEGIN
  ST:=ST+1; TIE_CAP :=0; ADD_CAP :=0;
  MSSTATE(NET,AC,TMS_BUS);
  FOR V:= 1 TO TMSAB DO
  BEGIN
    AQEQ1[ST,V]:= AC[MSAB[V]];
    ADD_CAP := ADD_CAP +AQEQ1[ST,V];
  END;
  FOR V:= 1 TO TMS_BUS DO TIE_CAP := TIE_CAP + NET[V+1,TMS_BUS+3];
  | WRITELN(' ACTUAL MODIFIED CAPACITY MARGIN AT ANY CONNECTED BUSES');
  WRITELN('-----');
  WRITE('STATE PROB FREQ TIE ADD');
  FOR V:= 1 TO TMSAB DO WRITE(MSB[MSAB[V]]:7);WRITELN;
  WRITELN('-----');
  WRITE(ST:5,' ',TOTALPB:8:6,TOTALFQ:8:4,TIE_CAP:7:2,ADD_CAP:7:2,' ');
  FOR V:= 1 TO TMSAB DO WRITE(AQEQ1[ST,V]:7:2);WRITELN;
  RD :=5.0;|
  ROUNDING(TIE_CAP,TMSAB,RD,AQEQ1[ST],RAQEQ);
  | ADD_CAP:=0;
  FOR V:= 1 TO TMSAB DO ADD_CAP := ADD_CAP + RAQEQ[V];
  WRITE(ST:5,' ',TOTALPB:8:6,TOTALFQ:8:4,TIE_CAP:7:2,ADD_CAP:7:2,' ');
  FOR V:= 1 TO TMSAB DO WRITE(RAQEQ[V]:7:2);WRITELN;
  WRITELN('-----');
  WRITELN(' PRESS [C] TO CONTINUE '); REPEAT KEY:=UPCASE(READKEY); UNTIL KEY='C';|
  AQEQ1[ST]:=RAQEQ;
|simultaneously combined equal state with last contingency|
  IF ST <> 1 THEN
  BEGIN
    X:= ST; CK1:= FALSE;
    REPEAT X:=X-1; Y:=0; CK2:= FALSE;
    REPEAT Y:=Y+1;
    IF (AQLINK1[X]=SETC1-COUT1)AND(AQEQ1[ST,Y] = AQEQ1[X,Y]) THEN
      CK2 := TRUE ELSE CK2 := FALSE;
    UNTIL (CK2=FALSE) OR (Y=TMSAB);
    IF (CK2=TRUE)AND(Y=TMSAB) THEN CK1 := TRUE ELSE CK1 := FALSE;
    UNTIL (CK1=TRUE) OR (X=1);
  END ELSE CK1:= FALSE;

```

```

IF CK1=TRUE THEN
BEGIN
    TEMPB:=AQPBI[X]; TEMUP:=AQUPI[X]; TEMDN:=AQDNI[X];
    AQPBI[X]:= TEMPB+TOTALPB;
    AQUPI[X]:= ((TEMPB*TEMUP)+(TOTALPB*UP))/AQPBI[X];
    AQDNI[X]:= ((TEMPB*TEMDN)+(TOTALPB*(DNMS-DN))/AQPBI[X];
    FOR V:= 1 TO TMSAB DO AQEQ1[ST,V]:=0;
    ST := ST-1;
END
ELSE BEGIN
    AQPBI[ST]:=TOTALPB;
    AQUPI[ST]:=UP;
    AQDNI[ST]:=DNMS-DN;
    AQLINK1[ST]:=SETC1-COUT1;
END;
END;
CTL1:=[]; CTL2:=[];
END;

BEGIN [start main procedure MSMODEL]
ST := 0; [CURT:=20;]
IF NOC1 < 2 THEN K:= NOC1 ELSE K :=2;
IF NOL1 < 2 THEN L:= NOL1 ELSE L :=2;
IF GENMS< 4 THEN N:=GENMS ELSE N :=4;
FOR CLEV := 0 TO K DO
BEGIN CST := 1;
    WHILE CST <= PERMUTATION (NOC1,CLEV) DO
    BEGIN
        FOR LLEV := 0 TO L DO
        BEGIN LNST := 1;
            WHILE (LNST <= PERMUTATION (NOL1,LLEV))AND(CLEV+LLEV<=2) DO
            BEGIN
                FOR GLEV := 0 TO N DO
                BEGIN
                    IF GLEV+LLEV+CLEV <= 4 THEN
BEGIN
                    TPB:= 1; TUP:= 0; TDN:=0; TNET:=MSMAT;
                    IF NOL1 <> 0 THEN
                    BEGIN
                        CTL3:=[]; CTL4:=[]; CTL5:=[]; CTL6:=[];
                        CONTING(SETL1,NOL1,LLEV,LNST,LOUT1);
                        FOR T:= 1 TO TN_LINE DO
                        BEGIN
                            IF T IN LOUT1 THEN
                            BEGIN
                                TPB := TPB*(1.0-PBL[T])/PBL[T];
                                TUP := TUP+RRL[T]; TDN := TDN+FRL[T];
                                IF NSB[T]<NEB[T] THEN
                                    TNET[NSB[T]+1,NEB[T]+1]:=TNET[NSB[T]+1,NEB[T]+1]-LINECAP[T]
                                ELSE TNET[NEB[T]+1,NSB[T]+1]:=TNET[NEB[T]+1,NSB[T]+1]-LINECAP[T];
                                CTL(T,MS,SETL1,TMS_BUS,CTL3,CTL4,CTL5,CTL6);
                                [ writeln('Line = ',lineno[t]:2,',');]
                            END;
                        END;
                    END;
                END;
            END;
        END;
    END;
END;

IF NOC1 <> 0 THEN
BEGIN
    CONTING(SETC1,NOC1,CLEV,CST,COUT1);
    FOR U:= 1 TO TN_LINE DO
    BEGIN

```

```

IF U IN COUT1 THEN
BEGIN
  TPB := TPB*(1.0-PBL[U])/PBL[U];
  TUP := TUP+RRL[U]; TDN := TDN+FRL[U];
  IF G_SB[U]= MS THEN
  TNET[NSB[U]+1,TMS_BUS+3]:=TNET[NSB[U]+1,TMS_BUS+3]-LINECAP[U]
  ELSE TNET[NEB[U]+1,TMS_BUS+3]:=TNET[NEB[U]+1,TMS_BUS+3]-LINECAP[U];
  [ writeln('Tie = ',lineno[u]:2,',');]
END;
END;
END;
CTL1:={}; CTL2:={};
IF (GLEV=0) OR (GENMS=0) THEN
BEGIN
  NET:=TNET; PB :=TPB; UP:=TUP; DN:=TDN; MSCAL;
  [ writeln('process glev = ',glev);]
END
ELSE
BEGIN
  CASE GLEV OF
    1 :BEGIN
      GST:=1;
      WHILE GST<=PERMUTATION(NOG1,GLEV) DO
      BEGIN
        MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; NET:=TNET;
        CONTING(SETG1,NOG1,GLEV,GST,GOUT1);
        FOR R:=1 TO TN_GEN DO
        BEGIN
          IF R IN GOUT1 THEN
          BEGIN
            MTP := MTP+GIDEN[R];
            PB := PB*(1-PBG[R])/PBG[R];
            UP := UP +RRG[R];
            DN := DN +FRG[R];
            NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-GENCAP[R];
            CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
          END;
        END;
        PB:= PB*MTP; MSCAL; GST:=GST+1;
        END; [writeln('process glev = ',glev);]
      END;
    2 :BEGIN
      FOR R:= 1 TO TN_GEN DO
      BEGIN
        NET:=TNET;
        IF (R IN SETG1)AND(GIDEN[R]>=2) THEN
        BEGIN
          MTP :=PERMUTATION(GIDEN[R],GLEV);
          PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
          UP :=TUP +2*RRG[R];
          DN :=TDN +2*FRG[R];
          NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-2*GENCAP[R];
          CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
          PB:= PB*MTP; MSCAL;
        END;
      END;
      GST:=1;
      WHILE GST<=PERMUTATION(NOG1,GLEV) DO
      BEGIN

```

```

MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; NET:=TNET;
CONTING(SETG1,NOG1,GLEV,GST,GOUT1);
FOR R:=1 TO TN_GEN DO
BEGIN
  IF R IN GOUT1 THEN
  BEGIN
    MTP := MTP*GIDEN[R];
    PB := PB*(1-PBG[R])/PBG[R];
    UP := UP +RRG[R];
    DN := DN +FRG[R];
    NET[1,NGENB[R]+1]:=-NET[1,NGENB[R]+1]-GENCAP[R];
    CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
  END;
END;
PB:= PB*MTP; MSCAL; GST:=GST+1;
END; |writeln('process glev = ',glev);|
END;

3 :BEGIN
FOR R:= 1 TO TN_GEN DO
BEGIN
  NET:=TNET;
  IF (R IN SETG1)AND(GIDEN[R]>=3) THEN
  BEGIN
    MTP :=PERMUTATION(GIDEN[R],GLEV);
    PB :=TPB*(Sqr((1-PBG[R])/PBG[R]))*(1-PBG[R])/PBG[R];
    UP :=TUP +3*RRG[R];
    DN :=TDN +3*FRG[R];
    NET[1,NGENB[R]+1]:=-NET[1,NGENB[R]+1]-3*GENCAP[R];
    CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
    PB:= PB*MTP; MSCAL;
  END;
END;

FOR R:= 1 TO TN_GEN DO
BEGIN
  IF (R IN SETG1)AND(GIDEN[R]>=2) THEN
  BEGIN
    TSETG1 :=SETG1 - [R];
    TNOG1 :=NOG1 -1;
    TGLEV :=GLEV -2;
    GST :=1 ;
    WHILE GST<=PERMUTATION(TNOG1,TGLEV) DO
    BEGIN
      MTP :=PERMUTATION(GIDEN[R],2);
      PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
      UP :=TUP +2*RRG[R];
      DN :=TDN +2*FRG[R];
      CONTING(TSETG1,TNOG1,TGLEV,GST,GOUT1);
      NET:=TNET;
      FOR S:=1 TO TN_GEN DO
      BEGIN
        IF S IN GOUT1 THEN
        BEGIN
          MTP := MTP*GIDEN[S];
          PB := PB*(1-PBG[S])/PBG[S];
          UP := UP +RRG[S];
          DN := DN +FRG[S];
          NET[1,NGENB[S]+1]:=-NET[1,NGENB[S]+1]-GENCAP[S];
          CTG(S,MS,SETL1,TMS_BUS,CTL1,CTL2);
        END;
      END;
    END;
  END;
END;

```

```

                                END;
                                NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-2*GENCAP[R];
                                CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
                                PB:= PB*MTP; MSCAL; GST:=GST+1;
                                END;
                                END;
                                END;

                                GST:=1;
                                WHILE GST<=PERMUTATION(NOG1,GLEV) DO
                                BEGIN
                                MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; NET:=TNET;
                                CONTING(SETG1,NOG1,GLEV,GST,GOUT1);
                                FOR R:=1 TO TN_GEN DO
                                BEGIN
                                IF R IN GOUT1 THEN
                                BEGIN
                                MTP := MTP*GIDEN[R];
                                PB := PB*(1-PBG[R])/PBG[R];
                                UP := UP +RRG[R];
                                DN := DN +FRG[R];
                                NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-GENCAP[R];
                                CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
                                END;
                                END;
                                PB:= PB*MTP; MSCAL; GST:=GST+1;
                                END; [writeln('process glev = ',glev);]
                                END;

                                4 :BEGIN
                                FOR R:= 1 TO TN_GEN DO
                                BEGIN
                                NET:=TNET;
                                IF (R IN SETG1)AND(GIDEN[R]>=4) THEN
                                BEGIN
                                MTP :=PERMUTATION(GIDEN[R],GLEV);
                                PB :=TPB*Sqr(Sqr((1-PBG[R])/PBG[R]));
                                UP :=TUP +4*RRG[R];
                                DN :=TDN +4*FRG[R];
                                NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-4*GENCAP[R];
                                CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
                                PB:= PB*MTP; MSCAL;
                                END;
                                END;

                                FOR R:= 1 TO TN_GEN DO
                                BEGIN
                                IF (R IN SETG1)AND(GIDEN[R]>=3) THEN
                                BEGIN
                                TSETG1 :=SETG1 - [R];
                                TNOG1 :=NOG1 -1;
                                TGLEV :=GLEV -3;
                                GST :=1 ;
                                WHILE GST<=PERMUTATION(TNOG1,TGLEV) DO
                                BEGIN
                                MTP :=PERMUTATION(GIDEN[R],3);
                                PB :=TPB*(Sqr((1-PBG[R])/PBG[R]))*(1-PBG[R])/PBG[R];
                                UP :=TUP +3*RRG[R];
                                DN :=TDN +3*FRG[R];
                                CONTING(TSETG1,TNOG1,TGLEV,GST,GOUT1);
                                NET:=TNET;

```



```

FOR S:=1 TO TN_GEN DO
BEGIN
  IF S IN GOUT1 THEN
  BEGIN
    MTP := MTP*GIDEN[S];
    PB := PB*(1-PBG[S])/PBG[S];
    UP := UP +RRG[S];
    DN := DN +FRG[S];
    NET[1,NGENB[S]+1]:=NET[1,NGENB[S]+1]-GENCAP[S];
    CTG(S,MS,SETL1,TMS_BUS,CTL1,CTL2);
  END;
  END;
  NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-3*GENCAP[R];
  CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
  PB:= PB*MTP; MSCAL; GST:=GST+1;
END;
END;
FOR R:= 1 TO TN_GEN DO
BEGIN
  IF (R IN SETG1)AND(GIDEN[R]>=2) THEN
  BEGIN
    TSETG1 :=SETG1 - [R];
    TNOG1 :=NOG1 -1;
    TGLEV :=GLEV -2;
    GST :=1 ;
    WHILE GST<=PERMUTATION(TNOG1,TGLEV) DO
    BEGIN
      MTP :=PERMUTATION(GIDEN[R],2);
      PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
      UP :=TUP +2*RRG[R];
      DN :=TDN +2*FRG[R];
      CONTING(TSETG1,TNOG1,TGLEV,GST,GOUT1);
      NET:=TNET;
      FOR S:=1 TO TN_GEN DO
      BEGIN
        IF S IN GOUT1 THEN
        BEGIN
          MTP := MTP*GIDEN[S];
          PB := PB*(1-PBG[S])/PBG[S];
          UP := UP +RRG[S];
          DN := DN +FRG[S];
          NET[1,NGENB[S]+1]:=NET[1,NGENB[S]+1]-GENCAP[S];
          CTG(S,MS,SETL1,TMS_BUS,CTL1,CTL2);
        END;
      END;
      NET[1,NGENB[R]+1]:=NET[1,NGENB[R]+1]-2*GENCAP[R];
      CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
      PB:= PB*MTP; MSCAL; GST:=GST+1;
    END;
  END;
END;
END;
GST:=1;
WHILE GST<=PERMUTATION(NOG1,GLEV) DO
BEGIN
  MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; NET:=TNET;
  CONTING(SETG1,NOG1,GLEV,GST,GOUT1);
  FOR R:=1 TO TN_GEN DO
  BEGIN

```

```

IF R IN GOUT1 THEN
BEGIN
MTP := MTP+GIDEN[R];
PB := PB*(1-PBG[R])/PBG[R];
UP := UP +RRG[R];
DN := DN +FRG[R];
NET[1,NGENB[R]+1]:=-NET[1,NGENB[R]+1]-GENCAP[R];
CTG(R,MS,SETL1,TMS_BUS,CTL1,CTL2);
END;
END;
PB:= PB*MTP; MSCAL; GST:=-GST+1;
END; [writeln('process glev = ',glev);]
END;
END;
LNST:=LNST+1;
END;
END;
CST:=CST+1;
END;
END;
TAQST1:=ST; [total adequacy equivalent states]
END;
END.

```

```

[Last version was been updated 28-9-91]
[Normal conditions 1 line-away bus transmission outage, 1 line-away bus generation outage]
UNIT      SSCAL9;
INTERFACE
USES      DOS,CRT,PRINTER,ALLDATA,MSCAL9;

```

```

PROCEDURE SSMODEL (INDEX2 :BOOLEAN; [adqev=true, entire cal=false]
  VAR AQEQ2:ARRZ; [agequacy equiv in each state of MS conting]
  VAR AQP2:ARRX; [probability in each state of MS conting]
  VAR AQUP2:ARRX; [departure to up in each state of MS conting]
  VAR AQDN2:ARRX; [departure to dn in each state of MS conting]
  VAR TAQST2:LONGINT; [total state of ADQ model before rearrange]
  VAR AQLINK2:ARRW);

```

IMPLEMENTATION

```

PROCEDURE SSMODEL (INDEX2 :BOOLEAN; [adqev=true, entire cal=false]
  VAR AQEQ2:ARRZ; [agequacy equiv in each state of MS conting]
  VAR AQP2:ARRX; [probability in each state of MS conting]
  VAR AQUP2:ARRX; [departure to up in each state of MS conting]
  VAR AQDN2:ARRX; [departure to dn in each state of MS conting]
  VAR TAQST2:LONGINT; [total state of ADQ model before rearrange]
  VAR AQLINK2:ARRW);

```

```

TYPE      SSBXX                = ARRAY[1..MAXBUS,1..5] OF REAL;

VAR
  A                : LONGINT;
  B,C,E            : BYTE;
  TEMSET           : SETNO;
  FNET             : ARRAY;
  FST,TT           : LONGINT;
  ADDLN            : REAL;
  LLEV ,TLEV ,GLEV : CTGLEVEL;
  LNST , TST , GST : LONGINT;
  LOUT2,TOUT2,GOUT2 : SETNO;
  R,S,T,M          : BYTE;
  PB,UP,DN,TOTALPB,TOTALPQ : REAL;
  MODE             : BOOLEAN;
  MGLD             : ARRAY[1..30] OF REAL;
  BINDEX           : SSBXX;
  KEY              : CHAR;
  X,Y,Z            : BYTE;
  TFNET           : ARRAY;
  TPB,TUP,TDN,prob,freq : REAL;
  TSETG2          : SETNO;
  TNOG2           : BYTE;
  MTP              : WORD;
  TGLEV           : CTGLEVEL;
  CTL1,CTL2,CTL3,CTL4,CTL5,CTL6 : SETNO;
  PLC,FLC,PLC1,FLC1 : REAL;
  SSDEMAND,BPI,BPA,BPE,MBP,SVI : REAL;
  TIMES,TIMEF,TIMESS : LONGINT;

```

```

Procedure INDEXCAL; [revise 23-7-91]

```

```

VAR
  I,J,U           : BYTE;
  [ CURT,|AVALOAD,EXPLOAD,TM1,TM2 : REAL;
  MAXLOAD ,MAXGEN : REAL;
  MAXLOAD1       : REAL;
  BYG,BYL,BYGL  : REAL;
  SBYG,SBYL,SBYGL : SETNO;
  SETLDCT1      : SETNO;
  CTLINDEX,ARINDEX : BOOLEAN;

```

```
sscurt,gbml,gbal,gbcurt,modfload : real;
```

```
BEGIN
[ CURT := 20.0; |
  IF INDEX2 = TRUE THEN
    BEGIN
      [ WRITELN('                USE NSMODEL STATE = ',A:4);|
      TOTALPB:= PBSS*PB*AQPB2[A];
      TOTALFQ:= TOTALPB*(UP*AQUP2[A]+(DNSS-DN*AQDN2[A]));
    END
  ELSE
    BEGIN
      TOTALPB:= PBSS*PB;          [ writeln('totalpb = ',totalpb);|
      TOTALFQ:= TOTALPB*(UP+(DNSS-DN)); [ writeln('totalfq = ',totalfq);|
    END;
  IF (TOTALPB>PBCUTOFF) AND (TOTALFQ>FQCUTOFF) THEN
    BEGIN
      FST := FST+1; MODE :=TRUE;
      MAXLOAD:=0; MAXGEN:=0; BYG:=0;  BYL:=0;  BYGL:=0;
      SBYG:=[]; SBYL:=[]; SBYGL:=[];
      FOR I:= 1 TO TSS_BUS DO
        BEGIN
          IF FNET[I+1,TSS_BUS+3] > 0 THEN FNET[1,I+1]:=FNET[1,I+1]+FNET[I+1,TSS_BUS+3];
          MAXLOAD :=MAXLOAD+FNET[I+1,TSS_BUS+2];
          MAXGEN  :=MAXGEN +FNET[1,I+1];
        END;
      IF MAXGEN > (1-CURT/100)*MAXLOAD THEN
        BEGIN
          MAXLOAD:=0;
          FOR I := 1 TO TSS_BUS DO
            BEGIN
              IF FNET[I+1,TSS_BUS+3] < 0 THEN
                FNET[I+1,TSS_BUS+2]:=FNET[I+1,TSS_BUS+2] - FNET[I+1,TSS_BUS+3];
                MAXLOAD := MAXLOAD+FNET[I+1,TSS_BUS+2];
            END;
          END;
        FOR I:= 1 TO TSS_BUS DO FNET[I+1,TSS_BUS+3]:= FNET[I+1,TSS_BUS+2];
        FOR I:= 1 TO TN_LOAD DO
          BEGIN
            IF I IN (CTL1) THEN
              BEGIN BYG := BYG+LOADCAP[I]; SBYG := SBYG+[NLOADB[I]]; END;
            IF I IN (CTL3+CTL4) THEN
              BEGIN BYL := BYL+LOADCAP[I]; SBYL := SBYL+[NLOADB[I]]; END;
            IF I IN (CTL1+CTL3+CTL4) THEN
              BEGIN BYGL:=BYGL+LOADCAP[I]; SBYGL:=SBYGL+[NLOADB[I]]; END;
          END;
        IF MAXGEN < MAXLOAD THEN
          BEGIN
            sscurt:=maxload-maxgen;
            gbml:=0; gbal:=0;
            MAXLOAD1:=0; SETLDCT1:=[];
            FOR I:= 1 TO TSS_BUS DO
              BEGIN
                IF (I IN SBYGL) AND (FNET[1,I+1]>0) AND(FNET[I+1,TSS_BUS+2]>0) THEN
                  BEGIN
                    TM1:=FNET[I+1,TSS_BUS+2] - (MAXLOAD-MAXGEN)*FNET[I+1,TSS_BUS+3]/BYGL;
                    TM2:=(1-CURT/100)*FNET[I+1,TSS_BUS+2];
                    IF (FNET[1,I+1]>FNET[I+1,TSS_BUS+2]) THEN
                      BEGIN
                        IF (TM1 <= TM2) THEN FNET[I+1,TSS_BUS+2] := TM2

```

```

ELSE
    FNET[I+1,TSS_BUS+2] := TM1;
END;
IF (FNET[1,I+1] < FNET[I+1,TSS_BUS+2]) THEN
BEGIN
    IF (TM1 <= FNET[1,I+1]) THEN FNET[I+1,TSS_BUS+2] := FNET[1,I+1]
    ELSE
        FNET[I+1,TSS_BUS+2] := TM1;
    END;
    gbml:=gbml+fnet[i+1,tss_bus+3];
    gbal:=gbal+fnet[i+1,tss_bus+2];
    SETLDCT1 :=SETLDCT1+[I];
END
else IF (I IN SBYGL) AND NOT(I IN SETLDCT1) THEN MAXLOAD1 :=MAXLOAD1+FNET[I+1,TSS_BUS+3];
END;
gbcurt:=gbml-gbal;
if maxload1 <> 0 then
begin
    if maxgen>(1-curt/100)*maxload then
    begin
        if sscurt-gbcurt>=curt/100*maxload1 then
        begin
            if(sscurt-gbcurt-curt/100*maxload1)<=curt/100*(maxload-gbml-maxload1) then
            begin
                for i := 1 to tss_bus do
                begin
                    if (i in sbygl)and not(i in setldct1) then
                        fnet[i+1,tss_bus+2]:= (1-curt/100)*fnet[i+1,tss_bus+3]
                    else if not(i in sbygl) then
                        fnet[i+1,tss_bus+2]:=fnet[i+1,tss_bus+3]-fnet[i+1,tss_bus+3]
                            *(sscurt-gbcurt-curt/100*maxload1)/(maxload-gbml-maxload1);
                    end;
                end
            else begin
                for i := 1 to tss_bus do
                begin
                    if not(i in sbygl) then
                        fnet[i+1,tss_bus+2]:= (1-curt/100)*fnet[i+1,tss_bus+3]
                    else if (i in sbygl)and not(i in setldct1) then
                        fnet[i+1,tss_bus+2]:=fnet[i+1,tss_bus+3]
                            -fnet[i+1,tss_bus+3]*(sscurt-gbcurt-curt/100*(maxload-gbml-maxload1))/maxload1;
                    end;
                end
            end
        else begin
            for i:= 1 to tss_bus do
            begin
                if (i in sbygl)and not(i in setldct1) then
                    fnet[i+1,tss_bus+2]:= fnet[i+1,tss_bus+3]
                        -fnet[i+1,tss_bus+3]*(sscurt-gbcurt)/maxload1;
                end;
            end
        end
    else begin
        for i := 1 to tss_bus do
        begin
            if not(i in setldct1) then
                fnet[i+1,tss_bus+2]:= fnet[i+1,tss_bus+3]
                    -fnet[i+1,tss_bus+3]*(sscurt-gbcurt)/(maxload-gbml);
            end;
        end
    end
end
else begin (maxload1=0)

```

```

      for i:= 1 to tss_bus do
      begin
        if not(i in sbygl) then
          fnet[i+1,tss_bus+2]:= fnet[i+1,tss_bus+3]
                                -fnet[i+1,tss_bus+3]*(sscurt-gbcurt)/(maxload-gbml);
        end;
      end;
      modfload:=0;
      for i := 1 to tss_bus do modfload:=modfload+fnet[i+1,tss_bus+2];
      if (Abs(maxgen)-Abs(modfload)>1.0E-2) or (Abs(maxgen)-Abs(modfload)<-1.0E-2) then
        writeln('INAPPROPRIATE LOAD BUS CURTAILMENT IN " SSCAL " ');
      END;
      MAXFLOW(FNET,TSS_BUS,MODE);
      EXPLOAD :=0; AVALOAD :=0;
      FOR I:= 1 TO TSS_BUS DO
      BEGIN
        EXPLOAD:=EXPLOAD+FNET[I+1,TSS_BUS+2];
        AVALOAD:=AVALOAD+FNET[TSS_BUS+2,I+1];
      END;
      IF AVALOAD <> MAXLOAD THEN
      BEGIN
        IF (EXPLOAD-AVALOAD) > 1.0E-5 THEN
        BEGIN
          IF MAXGEN >= MAXLOAD THEN
          BEGIN
            FOR I:= 1 TO TSS_BUS DO
            BEGIN
              IF I IN SBYL THEN
                FNET[I+1,TSS_BUS+2] := FNET[I+1,TSS_BUS+2]
                                      -(EXPLOAD-AVALOAD)*FNET[I+1,TSS_BUS+3]/BYL;
            END;
            MAXFLOW(FNET,TSS_BUS,MODE);
          END;
          MODE := FALSE;
          FOR I:= 1 TO TSS_BUS DO FNET[I+1,TSS_BUS+2]:=FNET[I+1,TSS_BUS+3];
          MAXFLOW(FNET,TSS_BUS,MODE);
        END
      ELSE
      BEGIN
        MODE := FALSE;
        FOR I:= 1 TO TSS_BUS DO FNET[I+1,TSS_BUS+2]:=FNET[I+1,TSS_BUS+3];
        MAXFLOW(FNET,TSS_BUS,MODE);
      END;
    ]
    WRITELN('*****FNET DATA AFTER MAXFLOW ***** ');
    FOR I := 1 TO TSS_BUS+2 DO
    BEGIN
      FOR J :=1 TO TSS_BUS+3 DO
        WRITE(FNET[I,J]:7:2, ' ');WRITELN;
      END;
      WRITELN(' PRESS [C] to continue ');
      REPEAT KEY := UPCASE(READKEY); UNTIL KEY ='C';!
      AVALOAD:=0;
      FOR I:=1 TO TSS_BUS DO AVALOAD:=AVALOAD+FNET[TSS_BUS+2,I+1];
      IF AVALOAD < MAXLOAD THEN
      BEGIN
        FOR U:=1 TO TSS_BUS DO
        BEGIN
          IF abs(FNET[U+1,TSS_BUS+2])-abs(FNET[TSS_BUS+2,U+1]) <1.0e-5 THEN MGLD[U]:=0
          ELSE IF FNET[TSS_BUS+2,U+1] < FNET[U+1,TSS_BUS+2] THEN
          BEGIN

```

```

IF INDEX2 = TRUE THEN
BEGIN
    IF FNET[U+1,TSS_BUS+2]>SSMAT[U+1,TSS_BUS+2] THEN
    BEGIN
        IF FNET[TSS_BUS+2,U+1]*SSMAT[U+1,TSS_BUS+2]/FNET[U+1,TSS_BUS+2]
        <= (1-CURT/100)*SSMAT[U+1,TSS_BUS+2] THEN
        BEGIN
            IF FNET[TSS_BUS+2,U+1] < SSMAT[U+1,TSS_BUS+2] THEN
                MGLD[U]:=FNET[TSS_BUS+2,U+1]-SSMAT[U+1,TSS_BUS+2]
            ELSE MGLD[U]:= 0;
        END
    END
    ELSE
MGLD[U]:=SSMAT[U+1,TSS_BUS+2]*FNET[TSS_BUS+2,U+1]/FNET[U+1,TSS_BUS+2]-ssmat[u+1,tss_bus+2];
    END
    ELSE MGLD[U]:=FNET[TSS_BUS+2,U+1]-FNET[U+1,TSS_BUS+2];
    END
    ELSE
        MGLD[U]:=FNET[TSS_BUS+2,U+1]-FNET[U+1,TSS_BUS+2];
    END;
END;
[*****CALCULATE BUS INDICES HERE*****]
CTLINDEX := FALSE; ARINDEX := FALSE;
FOR U:=1 TO TSS_BUS DO
BEGIN
    IF MGLD[U]<0.0 THEN
    BEGIN
        CTLINDEX := TRUE;
[prob of failure] BINDEX[U,1]:=BINDEX[U,1]+TOTALPB;
[freq of failure] BINDEX[U,2]:=BINDEX[U,2]+TOTALFQ;
[ELC] BINDEX[U,3]:=BINDEX[U,3]+(abs(MGLD[U]))*TOTALFQ;
[EENS] BINDEX[U,4]:=BINDEX[U,4]+(abs(MGLD[U]))*TOTALPB*8760;
[EDLC] BINDEX[U,5]:=BINDEX[U,5]+TOTALPB*8760;
        IF (INDEX2=FALSE)AND(U IN INTSET) THEN ARINDEX:=TRUE;
    END;
END;
[*****CALCULATE BUS INDICES HERE*****]
IF CTLINDEX = TRUE THEN BEGIN PLC :=PLC +TOTALPB; FLC :=FLC +TOTALFQ; END;
IF ARINDEX = TRUE THEN BEGIN PLC1:=PLC1+TOTALPB; FLC1:=FLC1+TOTALFQ; END;
END
ELSE IF AVALOAD > MAXLOAD THEN
BEGIN
    WRITELN(' ERROR IN CASE OF AVALOAD > MAXLOAD: PRESS [C] to continue ');
    REPEAT KEY := UPCASE(READKEY); UNTIL KEY ='C';
END;
END;
prob:=prob+totalpb;freq:=freq+totalfq;
END; CTL1:=[]; CTL2:=[];
END;

BEGIN [start main procedure SSMODEL]
FST:=0; prob:=0;freq:=0; PLC:=0; FLC:=0; PLC1:=0; FLC1:=0;
FOR R:=1 TO MAXBUS DO FOR S:= 1 TO 5 DO BEGIN BINDEX[R,S]:=0.0; MGLD[S]:=0; END;
IF INDEX2=FALSE THEN TT:=1
ELSE TT:= TAQST2;
IF NOL2 < 2 THEN X:= NOL2 ELSE X:= 2;
IF GENSS< 4 THEN Z:= GENSS ELSE Z:= 4;
TIMES:=0;
TIMECHECK(TIMES);
FOR A:=1 TO TT DO
BEGIN
    if (index2=false)or((index2=true)and(aqpb2[a]>COMADQ)) then
    begin

```

```

FOR LLEV := 0 TO X DO
BEGIN LNST := 1;
  WHILE (LNST <= PERMUTATION (NOL2,LLEV)) AND (LLEV <= 2) DO
  BEGIN
    FOR GLEV := 0 TO Z DO
    BEGIN
      IF LLEV+GLEV <= 4 THEN
      BEGIN
        IF INDEX2 = TRUE THEN
        BEGIN
          TFNET:=SSMAT;
          FOR B:= 1 TO TMSAB DO
          BEGIN
            TEMSET:={}; ADDLN:=0.0;
            FOR C:=1 TO TN_LINE DC
            BEGIN
              IF (C IN AQLINK2[A])AND((NEB[C]=MSAB[B])OR(NSB[C]=MSAB[B])) THEN
              BEGIN TEMSET:=TEMSET+[C]; ADDLN:=ADDLN+LINECAP[C]; END;
            END;
            FOR E:=1 TO TN_LINE DC
            BEGIN
              IF E IN TEMSET THEN
              BEGIN
                IF (G_SB[E]=MS)AND(AQEQ2[A,B]>0.0) THEN
                TFNET[NEB[E]+1,TSS_BUS+3] := TFNET[NEB[E]+1,TSS_BUS+3]+LINECAP[E]*AQEQ2[A,B]/ADDLN
                ELSE IF (G_SB[E]=MS)AND(AQEQ2[A,B]<0.0) THEN
                TFNET[NEB[E]+1,TSS_BUS+3] := TFNET[NEB[E]+1,TSS_BUS+3]+LINECAP[E]*AQEQ2[A,B]/ADDLN
                ELSE IF (G_EB[E]=MS)AND(AQEQ2[A,B]>0.0) THEN
                TFNET[NSB[E]+1,TSS_BUS+3] := TFNET[NSB[E]+1,TSS_BUS+3]+LINECAP[E]*AQEQ2[A,B]/ADDLN
                ELSE IF (G_EB[E]=MS)AND(AQEQ2[A,B]<0.0) THEN
                TFNET[NSB[E]+1,TSS_BUS+3] := TFNET[NSB[E]+1,TSS_BUS+3]+LINECAP[E]*AQEQ2[A,B]/ADDLN;
              END;
            END;
          END;
        ELSE TFNET:=SSMAT;
          TPB:=1.0; TUP:=0.0; TDN:=0.0;

        IF NOL2 <> 0 THEN
        BEGIN
          CTL3:={}; CTL4:={}; CTL5:={}; CTL6:={};
          CONTING(SETL2,NOL2,LLEV,LNST,LOUT2);
          FOR T:= 1 TO TN_LINE DO
          BEGIN
            IF T IN LOUT2 THEN
            BEGIN
              TPB := TPB*(1.0-PBL[T])/PBL[T];
              TUP := TUP+RRL[T]; TDN := TDN+FRL[T];
              IF NSB[T]<NEB[T] THEN
              TFNET[NSB[T]+1,NEB[T]+1]:=TFNET[NSB[T]+1,NEB[T]+1]-LINECAP[T]
              ELSE TFNET[NEB[T]+1,NSB[T]+1]:=TFNET[NEB[T]+1,NSB[T]+1]-LINECAP[T];
              CTL(T,SS,SETL2,TSS_BUS,CTL3,CTL4,CTL5,CTL6);
              | write('Line = ',LINENO[t]:2,',');|
            END;
          END; |writeln;|
        END;
      END;
    END;
  END;
  CTL1:={}; CTL2:={};
  IF (GLEV=0) OR (GENSS=0) THEN
  BEGIN
    FNET:=TFNET; PB :=TPB; UP:=TUP; DN:=TDN; INDEXCAL;
    | writeln('process glev = ',glev); |
  END;
END;

```



```

END
ELSE
  BEGIN
    CASE GLEV OF
      1 :BEGIN
        GST:=1;
        WHILE GST<=PERMUTATION(NOG2,GLEV) DO
          BEGIN
            MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; FNET:=TFNET;
            CONTING(SETG2,NOG2,GLEV,GST,GOUT2);
            FOR R:=1 TO TN_GEN DO
              BEGIN
                IF R IN GOUT2 THEN
                  BEGIN
                    MTP := MTP*GIDEN[R];
                    PB := PB*(1-PBG[R])/PBG[R];
                    UP := UP +RRG[R];
                    DN := DN +FRG[R];
                    FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-GENCAP[R];
                    CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
                  END;
                END;
                PB:= PB*MTP; INDEXCAL; GST:=GST+1;
            END; [writeln('process glev = ',glev);]
          END;
        2 :BEGIN
          FOR R:= 1 TO TN_GEN DO
            BEGIN
              FNET:=TFNET;
              IF (R IN SETG2)AND(GIDEN[R]>=2) THEN
                BEGIN
                  MTP :=PERMUTATION(GIDEN[R],GLEV);
                  PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
                  UP :=TUP +2*RRG[R];
                  DN :=TDN +2*FRG[R];
                  FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-2*GENCAP[R];
                  CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
                  PB:= PB*MTP; INDEXCAL;
                END;
              END;
            GST:=1;
            WHILE GST<=PERMUTATION(NOG2,GLEV) DO
              BEGIN
                MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; FNET:=TFNET;
                CONTING(SETG2,NOG2,GLEV,GST,GOUT2);
                FOR R:=1 TO TN_GEN DO
                  BEGIN
                    IF R IN GOUT2 THEN
                      BEGIN
                        MTP := MTP*GIDEN[R];
                        PB := PB*(1-PBG[R])/PBG[R];
                        UP := UP +RRG[R];
                        DN := DN +FRG[R];
                        FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-GENCAP[R];
                        CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
                      END;
                    END;
                    PB:= PB*MTP; INDEXCAL; GST:=GST+1;
                END; [writeln('process glev = ',glev);]
              END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

END;

3 :BEGIN
  FOR R:= 1 TO TN_GEN DO
    BEGIN
      FNET:=TFNET;
      IF (R IN SETG2)AND(GIDEN[R]>=3) THEN
        BEGIN
          MTP :=PERMUTATION(GIDEN[R],GLEV);
          PB :=TPB*(Sqr((1-PBG[R])/PBG[R]))*(1-PBG[R])/PBG[R];
          UP :=TUP +3*RRG[R];
          DN :=TDN +3*FRG[R];
          FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-3*GENCAP[R];
          CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
          PB:= PB*MTP; INDEXCAL;
        END;
      END;

      FOR R:= 1 TO TN_GEN DO
        BEGIN
          IF (R IN SETG2)AND(GIDEN[R]>=2) THEN
            BEGIN
              TSETG2 :=SETG2 - [R];
              TNOG2 :=NOG2 -1;
              TGLEV :=GLEV -2;
              GST :=1 ;
              WHILE GST<=PERMUTATION(TNOG2,TGLEV) DO
                BEGIN
                  MTP :=PERMUTATION(GIDEN[R],2);
                  PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
                  UP :=TUP +2*RRG[R];
                  DN :=TDN +2*FRG[R];
                  CONTING(TSETG2,TNOG2,TGLEV,GST,GOUT2);
                  FNET:=TFNET;
                  FOR S:=1 TO TN_GEN DO
                    BEGIN
                      IF S IN GOUT2 THEN
                        BEGIN
                          MTP := MTP*GIDEN[S];
                          PB := PB*(1-PBG[S])/PBG[S];
                          UP := UP +RRG[S];
                          DN := DN +FRG[S];
                          FNET[1,NGENB[S]+1]:=FNET[1,NGENB[S]+1]-GENCAP[S];
                          CTG(S,SS,SETL2,TSS_BUS,CTL1,CTL2);
                        END;
                      END;
                    FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-2*GENCAP[R];
                    CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
                    PB:= PB*MTP; INDEXCAL; GST:=GST+1;
                  END;
                END;
              END;
            END;
          END;

          GST:=1;
          WHILE GST<=PERMUTATION(NOG2,GLEV) DO
            BEGIN
              MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; FNET:=TFNET;
              CONTING(SETG2,NOG2,GLEV,GST,GOUT2);
              FOR R:=1 TO TN_GEN DO
                BEGIN
                  IF R IN GOUT2 THEN

```

```

BEGIN
    MTP := MTP*GIDEN[R];
    PB := PB*(1-PBG[R])/PBG[R];
    UP := UP +RRG[R];
    DN := DN +FRG[R];
    FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-GENCAP[R];
    CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
END;
END;
PB:= PB*MTP; INDEXCAL; GST:=GST+1;
END; {writeln('process glev = ',glev);}
END;

4 :BEGIN
FOR R:= 1 TO TN_GEN DO
BEGIN
    FNET:=TFNET;
    IF (R IN SETG2)AND(GIDEN[R]>=4) THEN
    BEGIN
        MTP :=PERMUTATION(GIDEN[R],GLEV);
        PB :=TPB*Sqr(Sqr((1-PBG[R])/PBG[R]));
        UP :=TUP +4*RRG[R];
        DN :=TDN +4*FRG[R];
        FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-4*GENCAP[R];
        CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
        PB:= PB*MTP; INDEXCAL;
    END;
END;

FOR R:= 1 TO TN_GEN DO
BEGIN
    IF (R IN SETG2)AND(GIDEN[R]>=3) THEN
    BEGIN
        TSETG2 :=SETG2 - [R];
        TNOG2 :=NOG2 -1;
        TGLEV :=GLEV -3;
        GST :=1 ;
        WHILE GST<=PERMUTATION(TNOG2,TGLEV) DO
        BEGIN
            MTP :=PERMUTATION(GIDEN[R],3);
            PB :=TPB*(Sqr((1-PBG[R])/PBG[R]))*(1-PBG[R])/PBG[R];
            UP :=TUP +3*RRG[R];
            DN :=TDN +3*FRG[R];
            CONTING(TSETG2,TNOG2,TGLEV,GST,GOUT2);
            FNET:=TFNET;
            FOR S:=1 TO TN_GEN DO
            BEGIN
                IF S IN GOUT2 THEN
                BEGIN
                    MTP := MTP*GIDEN[S];
                    PB := PB*(1-PBG[S])/PBG[S];
                    UP := UP +RRG[S];
                    DN := DN +FRG[S];
                    FNET[1,NGENB[S]+1]:=FNET[1,NGENB[S]+1]-GENCAP[S];
                    CTG(S,SS,SETL2,TSS_BUS,CTL1,CTL2);
                END;
            END;
            FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-3*GENCAP[R];
            CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
            PB:= PB*MTP; INDEXCAL; GST:=GST+1;
        END;
    END;
END;

```

```

        END;
    END;

    FOR R:= 1 TO TN_GEN DO
    BEGIN
        IF (R IN SETG2)AND(GIDEN[R]>=2) THEN
        BEGIN
            TSETG2 :=SETG2 - [R];
            TNOG2 :=NOG2 -1;
            TGLEV :=GLEV -2;
            GST :=1 ;
            WHILE GST<=PERMUTATION(TNOG2,TGLEV) DO
            BEGIN
                MTP :=PERMUTATION(GIDEN[R],2);
                PB :=TPB*Sqr((1-PBG[R])/PBG[R]);
                UP :=TUP +2*RRG[R];
                DN :=TDN +2*FRG[R];
                CONTING(TSETG2,TNOG2,TGLEV,GST,GOUT2);
                FNET:=TFNET;
                FOR S:=1 TO TN_GEN DO
                BEGIN
                    IF S IN GOUT2 THEN
                    BEGIN
                        MTP := MTP*GIDEN[S];
                        PB := PB*(1-PBG[S])/PBG[S];
                        UP := UP +RRG[S];
                        DN := DN +FRG[S];
                        FNET[1,NGENB[S]+1]:=FNET[1,NGENB[S]+1]-GENCAP[S];
                        CTG(S,SS,SETL2,TSS_BUS,CTL1,CTL2);
                    END;
                END;
                FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-2*GENCAP[R];
                CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
                PB:= PB*MTP; INDEXCAL; GST:=GST+1;
            END;
        END;
    END;

    GST:=1;
    WHILE GST<=PERMUTATION(NOG2,GLEV) DO
    BEGIN
        MTP :=1; PB:=TPB; UP:=TUP; DN:=TDN; FNET:=TFNET;
        CONTING(SETG2,NOG2,GLEV,GST,GOUT2);
        FOR R:=1 TO TN_GEN DO
        BEGIN
            IF R IN GOUT2 THEN
            BEGIN
                MTP := MTP*GIDEN[R];
                PB := PB*(1-PBG[R])/PBG[R];
                UP := UP +RRG[R];
                DN := DN +FRG[R];
                FNET[1,NGENB[R]+1]:=FNET[1,NGENB[R]+1]-GENCAP[R];
                CTG(R,SS,SETL2,TSS_BUS,CTL1,CTL2);
            END;
        END;
        PB:= PB*MTP; INDEXCAL; GST:=GST+1;
    END; [writeln('process glev = ',glev);]
END; [case4]
END; [case]
END; [else]
END;

```

```

                                END;
                                LNST:=LNST+1;
                                END;
                                END;
                                end;
END;
TIMECHECK(TIMEF);
TIMSS:=TIMEF-TIMES;
WRITELN('          TIME TO CALCULATE ADEQUACY INDICES      = ',TIMSS:5,'      Sec');
WRITELN('          PROBABILITY OF ALL STATES IN CALCULATION = ',PROB:10:8);
WRITELN('          FREQUENCY OF ALL STATES IN CALCULATION   = ',FREQ:10:4,' occ/year');
WRITELN('          TOTAL STATES IN INDEX CALCULATION       = ',FST:5,'      states');
IF INDEX2=TRUE THEN
BEGIN
  WRITELN('          PROBABILITY OF FAILURE IN SUB-SYSTEM     = ',PLC:10:8);
  WRITELN('          FREQUENCY OF FAILURE IN SUB-SYSTEM      = ',FLC:10:4,' occ/year');
END
ELSE
BEGIN
  WRITELN('          PROBABILITY OF FAILURE IN OVERALL SYSTEM = ',PLC:10:8);
  WRITELN('          FREQUENCY OF FAILURE IN OVERALL SYSTEM = ',FLC:10:4,' occ/year');
  IF INTSET<>[] THEN
  BEGIN
    WRITELN('          PROBABILITY OF FAILURE IN SUB-SYSTEM     = ',PLC1:10:8);
    WRITELN('          FREQUENCY OF FAILURE IN SUB-SYSTEM      = ',FLC1:10:4,' occ/year');
  END;
END;
WRITELN('          |*****|');
WRITELN('          |          BASIC BUS INDICES IN SUBSYSTEM          |');
WRITELN('          |*****|');
WRITELN('          |-----|');
WRITELN('          | BUS PROBABILITY FREQUENCY ELC EENS EDLC |');
WRITELN('          | NO. OF FAILURE OF FAILURE (MW) (MWh) (hours) |');
WRITELN('          |-----|');
FOR S:= 1 TO TSS_BUS DO
WRITELN('          |',SSB[S]:4,' ',BINDEX[S,1]:12:9,' ',BINDEX[S,2]:12:8,BINDEX[S,3]:11:4,BINDEX[S,4]:11:4,
          BINDEX[S,5]:11:4,' |');
WRITELN('          |-----|');
SSDEMAND:=0; BPI:=0; BPA:=0; BPE:=0;
FOR S:=1 TO TSS_BUS DO
BEGIN
  SSDEMAND:=SSDEMAND+SSMAT[S+1,TSS_BUS+2];
  BPI :=BPI+BINDEX[S,3];
  BPA :=BPA+BINDEX[S,3];
  BPE :=BPE+BINDEX[S,4];
END;
BPI:=BPI/SSDEMAND; BPA:=BPA/FLC; BPE:=BPE/SSDEMAND; MBP:=BPE/8760; SVI:=BPE*60;
IF INDEX2=FALSE THEN
  WRITELN('          ANNUALIZED OVERALL SYSTEM INDICES  ')
ELSE WRITELN('          ANNUALIZED SUB-SYSTEM INDICES  ');
WRITELN('Bulk Power Interruption Index      = ',BPI:11:4,' MW/MW-year');
WRITELN('Bulk Power-supply Average MW Curtailment Index = ',BPA:11:4,' MW/disturbance');
WRITELN('Bulk Power/Energy Curtailment Index   = ',BPE:11:4,' MWh/MW-year');
WRITELN('Modified Bulk Power Energy Curtailment Index = ',MBP:15:8);
WRITELN('Severity Index                      = ',SVI:11:4,' System-Min');
SSDEMAND:=0; BPI:=0; BPA:=0; BPE:=0;
IF (INDEX2=FALSE)AND(INTSET<>[]) THEN
BEGIN
  FOR S:= 1 TO TN_LOAD DO
  BEGIN
    IF LOADB[S] IN INTSET THEN SSDEMAND:=SSDEMAND+LOADCAP[S];
  END;
END;

```

```

END;
FOR S:= 1 TO TSS_BUS DO
BEGIN
  IF S IN INTSET THEN
  BEGIN
    BPI      :=BPI+BINDEX[S,3];
    BPA      :=BPA+BINDEX[S,3];
    BPE      :=BPE+BINDEX[S,4];
  END;
  END;
  BPI:=BPI/SSDEMAND; BPA:=BPA/FLC1; BPE:=BPE/SSDEMAND; MBP:=BPE/8760; SVI:=BPE*60;
  Writeln('          ANNUALIZED SUB-SYSTEM INDICES          ');
  Writeln('Bulk Power Interruption Index          = ',BPI:11:4,' MW/MW-year');
  Writeln('Bulk Power-supply Average MW Curtailment Index = ',BPA:11:4,' MW/disturbance');
  Writeln('Bulk Power/Energy Curtailment Index          = ',BPE:11:4,' MWh/MW-year');
  Writeln('Modified Bulk Power Energy Curtailment Index = ',MBP:15:8);
  Writeln('Severity Index                              = ',SVI:11:4,' System-Min');
END;
[ PRINT OUT ALL AVAILABLE INDICES]
Writeln('          CHECK PRINTER IS READY          ');
Writeln('          PRESS [P] TO PRINT OUTPUT      ');
REPEAT KEY:= UPCASE(READKEY); UNTIL KEY='P';
Writeln(LST,'          TIME TO CALCULATE ADEQUACY INDICES = ',TIMESS:5,' Sec');
Writeln(LST,'          PROBABILITY OF ALL STATES IN CALCULATION = ',PROB:10:8);
Writeln(LST,'          FREQUENCY OF ALL STATES IN CALCULATION = ',FREQ:10:4,' occ/year');
Writeln(LST,'          TOTAL STATES IN INDEX CALCULATION = ',FST:5,' states');
IF INDEX2=TRUE THEN
BEGIN
  Writeln(LST,'          PROBABILITY OF FAILURE IN SUB-SYSTEM = ',PLC:10:8);
  Writeln(LST,'          FREQUENCY OF FAILURE IN SUB-SYSTEM = ',FLC:10:4,' occ/year');
END
ELSE
BEGIN
  Writeln(LST,'          PROBABILITY OF FAILURE IN OVERALL SYSTEM = ',PLC:10:8);
  Writeln(LST,'          FREQUENCY OF FAILURE IN OVERALL SYSTEM = ',FLC:10:4,' occ/year');
  IF INTSET<>[] THEN
  BEGIN
    Writeln(LST,'          PROBABILITY OF FAILURE IN SUB-SYSTEM = ',PLC1:10:8);
    Writeln(LST,'          FREQUENCY OF FAILURE IN SUB-SYSTEM = ',FLC1:10:4,' occ/year');
  END;
END;
Writeln(LST,'          *****');
Writeln(LST,'          BASIC BUS INDICES IN SUBSYSTEM          ');
Writeln(LST,'          *****');
Writeln(LST,'          -----');
Writeln(LST,'          | BUS PROBABILITY FREQUENCY ELC EENS EDLC |');
Writeln(LST,'          | NO. OF FAILURE OF FAILURE (MW) (MWh) (hours) |');
Writeln(LST,'          |-----|');
FOR S:= 1 TO TSS_BUS DO
Writeln(LST,'          |',SSB[S]:4,' |',BINDEX[S,1]:12:9,' |',BINDEX[S,2]:12:8,BINDEX[S,3]:11:4,BINDEX[S,4]:11:4,
          BINDEX[S,5]:11:4,' |');
Writeln(LST,'          |-----|');
SSDEMAND:=0; BPI:=0; BPA:=0; BPE:=0;
FOR S:=1 TO TSS_BUS DO
BEGIN
  SSDEMAND:=SSDEMAND+SSMAT[S+1,TSS_BUS+2];
  BPI      :=BPI+BINDEX[S,3];
  BPA      :=BPA+BINDEX[S,3];
  BPE      :=BPE+BINDEX[S,4];
END;
  BPI:=BPI/SSDEMAND; BPA:=BPA/FLC; BPE:=BPE/SSDEMAND; MBP:=BPE/8760; SVI:=BPE*60;

```

```

IF INDEX2=FALSE THEN
  WRITELN(LST,'          ANNUALIZED OVERALL SYSTEM INDICES  ')
ELSE WRITELN(LST,'          ANNUALIZED SUB-SYSTEM INDICES  ');
WRITELN(LST,'Bulk Power Interruption Index           = ',BPI:11:4,' MW/MW-year');
WRITELN(LST,'Bulk Power-supply Average MW Curtailment Index = ',BPA:11:4,' MW/disturbance');
WRITELN(LST,'Bulk Power/Energy Curtailment Index         = ',BPE:11:4,' MWh/MW-year');
WRITELN(LST,'Modified Bulk Power Energy Curtailment Index   = ',MBP:15:8);
WRITELN(LST,'Severity Index                                   = ',SVI:11:4,' System-Min');
SSDEMAND:=0; BPI:=0; BPA:=0; BPE:=0;
IF (INDEX2=FALSE)AND(INTSET<>[]) THEN
BEGIN
  FOR S:= 1 TO TN_LOAD DO
  BEGIN
    IF LOADB[S] IN INTSET THEN SSDEMAND:=SSDEMAND+LOADCAP[S];
  END;
  FOR S:= 1 TO TSS_BUS DO
  BEGIN
    IF S IN INTSET THEN
    BEGIN
      BPI      :=BPI+BINDEX[S,3];
      BPA      :=BPA+BINDEX[S,3];
      BPE      :=BPE+BINDEX[S,4];
    END;
  END;
  BPI:=BPI/SSDEMAND; BPA:=BPA/FLC1; BPE:=BPE/SSDEMAND; MBP:=BPE/8760; SVI:=BPE*60;
  WRITELN(LST,'          ANNUALIZED SUB-SYSTEM INDICES  ');
  WRITELN(LST,'Bulk Power Interruption Index           = ',BPI:11:4,' MW/MW-year');
  WRITELN(LST,'Bulk Power-supply Average MW Curtailment Index = ',BPA:11:4,' MW/disturbance');
  WRITELN(LST,'Bulk Power/Energy Curtailment Index         = ',BPE:11:4,' MWh/MW-year');
  WRITELN(LST,'Modified Bulk Power Energy Curtailment Index   = ',MBP:15:8);
  WRITELN(LST,'Severity Index                                   = ',SVI:11:4,' System-Min');
END;
| WRITELN('          PRESS [E] TO EXIT MAIN PROGRAM ');
REPEAT KEY:= UPCASE(READKEY); UNTIL KEY='E';|
END;

END.

```

```

[This main program to calculate aequacy indices updated 28_9_91 ]
PROGRAM HL2INDEX;
[SN+]
[SR+]
[$M 40000,0,655360]
USES      DOS,CRT,PRINTER,ALLDATA,MSCAL9,SSCAL9;
VAR       AQEQ           : ARRZ;
          AQPBAQUPAQDN   : ARRX;
          AQLINK         : ARRW;
          P,Q,TAQST      : LONGINT;
          A,B,FREQ       : REAL;
          TIMES,TIMEF,TIMEMS: LONGINT;
          KEYP           : CHAR;

BEGIN {main program HL2INDEX}
  DATAENTRY;
  READDATA;
  MODFAREA;
  FOR P:= 1 TO MAXSTAT DO
  BEGIN
    FOR Q:= 1 TO ADQBUS DO
    BEGIN
      AQEQ[P,Q] := 0;
      AQPBA[P]  := 0;
      AQUP[P]   := 0;
      AQDN[P]   := 0;
    END; AQLINK[P] := [];
  END;
  WRITELN;
  WRITE('INPUT % of CURTAILABLE LOAD in SYSTEM BUSES (10,15,20..) = '); READLN(CURT);
  TIMEMS:=0;
  assign(lst,'outp.dat');rewrite(lst);
  IF INDEXT = TRUE THEN
  BEGIN
    WRITE('INPUT APPROXIMATE STEP LEVEL in MODEL ADEQUACY EQUIVALENTS (1,5,10..) = '); READLN(RD);
    WRITELN('INPUT CUT-OFF VALUE of SELECTED STATE PROBABILITY in MADQ');
    WRITE('TO COMBINE WITH SUB-SYSTEM INDEX CALCULATION (REAL 1.0E-5..1.0E-8) = '); READLN(COMADQ);
    TIMECHECK(TIMES);
    MSMODEL(AQEQ,AQPBA,AQUP,AQDN,TAQST,AQLINK);
    TIMECHECK(TIMEF);
    TIMEMS:=TIMEF-TIMES;
    | WRITELN(LST,'-----');
    WRITE (LST,'STATE PROBABIL FREQ UP_RATE DN_RATE ');
    FOR P:= 1 TO TMSAB DO WRITE(LST,MSB[MSAB[P]]:8);WRITELN(LST);
    WRITELN(LST,'-----');
    A:=0;B:=0;
    FOR Q := 1 TO TAQST DO
    BEGIN
      FREQ := (AQUP[Q]+AQDN[Q])*AQPBA[Q];
      A:= A+AQPBA[Q]; B:= B+FREQ;
      | WRITE(LST,Q:3,' ',AQPBA[Q]:10:8,FREQ:8:3,AQUP[Q]:8:2,AQDN[Q]:8:2,' ');
      FOR P:= 1 TO TMSAB DO WRITE(LST,' ',AQEQ[Q,P]:7:2);WRITE(LST,' ');
      FOR P:= 1 TO MAXELEM DO
      BEGIN
        IF P IN AQLINK[Q] THEN WRITE(LST,P:2,',');
      END; WRITELN(LST);|
    END;
  | WRITELN(LST,'-----');
  WRITELN(LST);
  WRITELN('          TOTAL STATES OF MODEL ADEQUACY EQUIVALENTS = ',TAQST:5);
  WRITELN('          TIME TO DEVELOP MODEL ADEQUACY EQUIVALENTS = ',TIMEMS:6,'          Sec');

```



```
WRITELN(LST,'PROBABILITY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = ',A:10:8);
WRITELN(LST,'FREQUENCY OF ALL STATES IN MODEL ADEQUACY EQUIVALENTS = ',B:10:4);
WRITELN(LST,'          TOTAL STATES OF MODEL ADEQUACY EQUIVALENTS = ',TAQST:5);
WRITELN(LST,'          TIME TO DEVELOP MODEL ADEQUACY EQUIVALENTS = ',TIMEMS:6,'      Sec');
END;
SSMODEL(INDEXT,AQEQ,AQPB,AQUP,AQDN,TAQST,AQLINK);
close(lst);
WRITELN(' RUNNING PROGRAM IS READY!!  PRESS [E] TO EXIT PROGRAM ');
REPEAT KEYP:=UPCASE(READKEY); UNTIL KEYP='E';
END.
```

ประวัติผู้เขียน

นาย สันติ ชุ่มศรี เกิดวันที่ 28 กุมภาพันธ์ พ.ศ.2507 ที่ อ.เมือง จังหวัดชลบุรี สำเร็จปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี 2529 แล้วเข้าร่วมงานในฐานะวิศวกรควบคุม บำรุงรักษาระบบไฟฟ้า เหมืองใต้ดินและโรงงานลอยแร่ บริษัท KEMCO จ.กาญจนบุรี เป็นเวลาประมาณ 1 ปี จากนั้นได้เข้าทำงานเป็นวิศวกรออกแบบระบบข่าสสายโทรศัพท์ตอนนอก เป็นเวลาอีก 2 ปี ก่อนเข้าศึกษาต่อปริญญาโทในภาควิชาวิศวกรรมไฟฟ้า สาขาระบบพลังงาน ภาควิชาการศึกษา 2532 ระหว่างศึกษาต่อได้ทำหน้าที่เป็นผู้ช่วยวิจัยประจำศูนย์วิจัยและอบรมพลังงาน จุฬาลงกรณ์มหาวิทยาลัย และมูลนิธิสถาบันวิจัยเพื่อการพัฒนาประเทศไทย