

```
1 clc;clear;
2 %% Standardizing generator sub-transient reactances
3 s_new = 100; % MVA
4 s_old = [250; 100; 80; 50; 50;]; % MVA
5
6 x1 = 15/100;
7 x2 = 12/100;
8 x3 = 10/100;
9 x4 = 9/100;
10 x5 = 8/100;
11 x = [1 x1; 2 x2; 3 x3; 4 x4; 5 x5];
12
13 % converting xolds to xnews
14 x(:,2) = (x(:,2)./s_old)*s_new;
15 [r_x,~] = size(x);
16 %% Y bus calculation
17
18 % import data as table
19 linedata = readmatrix("line_data.xlsx");
20 % for k=1:r_x
21 %     linedata(end+1,1) = x(k,1);
22 %     linedata(end,2) = x(k,1);
23 %     linedata(end,4) = x(k,2);
24 % end
25 [r_data,~] = size(linedata);
26
27 % Preprocessing
28 number_of_buses = max(max(linedata(:,1:2))); % picking the highest numbered bus
29 y_bus = zeros(number_of_buses);
30 [length_y_bus,~] = size(y_bus);
31
32 % formatting the impedances
33 R = linedata(:,3);
34 X = linedata(:,4);
35 B = linedata(:,5);
36 impedances = [R X.*1i B*1i./2];
37
38 % calculating the diagonal elements
39 for m=1:length_y_bus % m is the index of the y bus (row)
40     z = zeros(1,3); % initialize 3 columns (R,X,B)
41     for n=1:r_data % n is the index of the linedata (row)
42         if linedata(n,1)==m||linedata(n,2)==m
43             z = [z; linedata(n,3:5)]; % collecting all the impedances
44         end
45     end
46     [r_z,~] = size(z);
47     for impedance=1:r_z
48         R = z(impedance,1)
49         X = z(impedance,2)
```

```

50     B = z(impedance,3)
51     if R~=0 || X~=0
52         y_bus(m,m) = y_bus(m,m) + 1./(R+X.*1i) + B.*1i/2;
53     end
54 end
55 end
56
57 % calculating off diagonal elements
58 nodes = linedata(:,1:2);
59 y_off = zeros(14,14);
60 impedances = impedances(:,1)+impedances(:,2);
61 [r,~] = size(linedata);
62
63
64 for m=1:r
65     nodes(m,1)
66     nodes(m,2)
67     y_off(nodes(m,1),nodes(m,2)) = -1/impedances(m);
68     y_off(nodes(m,2),nodes(m,1)) = y_off(nodes(m,1),nodes(m,2));
69 end
70
71 y_bus = y_bus + y_off;
72 %% z bus calculation
73 z_bus = y_bus^-1;
74
75 % adding the generator sub-transient reactances to the z-bus matrix
76 for k=1:length(x)
77     z_bus(k,k) = x(k,2)+z_bus(k,k);
78 end
79
80 %% prefault voltages
81
82 v = readmatrix('voltages.xlsx'); % prefault voltages
83 v_magnitudes = v(:,1);
84 v_angles = deg2rad(v(:,2));
85
86 % converting to rectangular
87 vx = v_magnitudes.*cos(v_angles);
88 vy = v_magnitudes.*sin(v_angles);
89 v = vx+vy*1i;
90
91 %% q1: sags
92
93 %Calculate the voltage sag at bus 4 and bus 13 when a three-phase-fault occurs at
each bus
94 %in the system
95 sag_4_q1 = zeros(14,1);
96 sag_13_q1 = zeros(14,1);
97 for bus=1:14

```

```
98     %if bus~=4|bus~=13
99         sag_4_q1(bus) = (1-z_bus(4,bus)/z_bus(bus,bus))*v(bus);
100         sag_13_q1(bus) = (1-z_bus(13,bus)/z_bus(bus,bus))*v(bus);
101     %end
102 end
103
104 % sag magnitudes
105 sag_4_q1 = abs(sag_4_q1);
106 sag_13_q1 = abs(sag_13_q1);
107 %% q2: sags (lines 4-5 and 6-13 are open)
108
109 % forming the y,z buses
110 y_bus_q2 = find_y_bus('line_data_q2.xlsx');
111 z_bus_q2 = y_bus_q2^-1;
112 % adding the generator sub-transient reactances to the z-bus matrix
113 for k=1:length(x)
114     z_bus_q2(k,k) = x(k,2)+z_bus_q2(k,k);
115 end
116
117 %faults
118 sag_4_q2 = zeros(14,1);
119 sag_13_q2 = zeros(14,1);
120 for bus=1:14
121     %if bus~=4|bus~=13
122         sag_4_q2(bus) = (1-z_bus_q2(4,bus)/z_bus_q2(bus,bus))*v(bus);
123         sag_13_q2(bus) = (1-z_bus_q2(13,bus)/z_bus_q2(bus,bus))*v(bus);
124     %end
125 end
126 % sag magnitudes
127 sag_4_q2 = abs(sag_4_q2);
128 sag_13_q2 = abs(sag_13_q2);
129 %% q3: sags (no x3 and x5)
130
131 s_old = [250 100 50]; % MVA
132 x_q3 = [x1 x2 x4]; % removing the sub transient reactances
133 x_q3 = x_q3*1i.*s_new./s_old; % converting to new MVA base
134
135 z_bus_q3 = y_bus^-1;
136 % adding the generator sub-transient reactances to the z-bus matrix
137 for k=1:length(x_q3)
138     z_bus_q3(k,k) = x_q3(k)+z_bus_q3(k,k);
139 end
140
141 % faults
142 sag_4_q3 = zeros(14,1);
143 sag_13_q3 = zeros(14,1);
144 for bus=1:14
145     %if bus~=4|bus~=13
146         sag_4_q3(bus) = (1-z_bus_q3(4,bus)/z_bus_q3(bus,bus))*v(bus);
```

```
147         sag_13_q3(bus) = (1-z_bus_q3(13,bus)/z_bus_q3(bus,bus))*v(bus);
148     %end
149 end
150 % sag magnitudes
151 sag_4_q3 = abs(sag_4_q3);
152 sag_13_q3 = abs(sag_13_q3);
153 %% q4: no of sags
154
155 d = linedata(:,6); %line distance
156 frequency_100 = linedata(:,7); % faults/100km/yr
157
158 frequency_d = frequency_100.*d./100;
159 [r,c] = size(nodes);
160 % calculating average sags
161 average_sags_4 = zeros(r,3); % will be of the same size as the nodes matrix
162 average_sags_4(:,1:2) = nodes;
163 average_sags_13 = zeros(r,3); % will be of the same size as the nodes matrix
164 average_sags_13(:,1:2) = nodes;
165
166 % sags at lines for bus 4
167 for k=1:r
168     sag1 = sag_4_q1(nodes(k,1));
169     sag2 = sag_4_q1(nodes(k,2));
170     average_sags_4(k,3) = 0.5*(sag1+sag2);
171 end
172
173 % sags at lines for bus 13
174 for k=1:r
175     sag1 = sag_13_q1(nodes(k,1));
176     sag2 = sag_13_q1(nodes(k,2));
177     average_sags_13(k,3) = 0.5*(sag1+sag2);
178 end
179
180 frequency_table = [nodes average_sags_4(:,3) average_sags_13(:,3) frequency_d];
181 f_sag_4 = 0;
182 f_sag_13 = 0;
183 % frequency of sag under 40%=0.4 pu
184 [r,~] = size(frequency_table);
185 for k=1:r
186     if frequency_table(k,3)<0.4
187         f_sag_4 = f_sag_4 + frequency_table(k,5);
188     end
189     if frequency_table(k,4)<0.4
190         f_sag_13 = f_sag_13 + frequency_table(k,5);
191     end
192 end
193
194 %% q4: Bar chart
195
```

```
196 % creating chart
197 chart_intervals = 0:0.1:1; % each number represents an upper bound for an interval
198 chart_intervals = chart_intervals';
199 chart = [chart_intervals zeros(length(chart_intervals),2)];
200 [r,~] = size(chart);
201
202 % filling the chart
203 % sags at bus 4
204 for k=1:length(sag_4_q3)
205     for index=2:r
206         if chart(index-1,1)<sag_4_q3(k)&&chart(index,1)<sag_4_q3(k)
207             chart(index,2) = chart(index,2)+1;
208         end
209     end
210 end
211 % sags at bus 13
212 for k=1:length(sag_13_q3)
213     for index=2:r
214         if chart(index-1,1)<sag_13_q3(k)&&chart(index,1)<sag_13_q3(k)
215             chart(index,3) = chart(index,3)+1;
216         end
217     end
218 end
219
220 %%
```