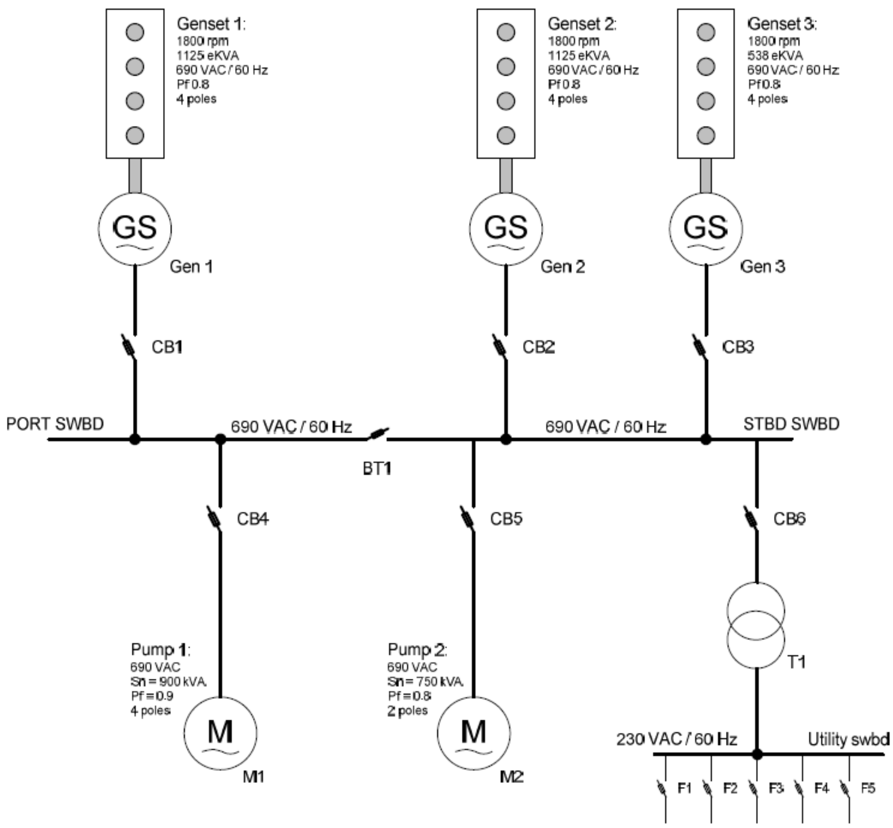
NTNU

TMR4290

Project 1 - Matlab calculations for a marine power plant

Authors:
Shiv Jeet Rai

March 13, 2016



Contents

1 Derivations

1.1 Loads

1.1.1 Equation for "Resulting per-phase impedance of induction motor"

The resulting per-phase impedance of the induction motor is given by equation(1)

$$Z_{\text{Motor}} = R_{\text{Sm}} + jX_{\text{Sm}} + \frac{1}{\frac{1}{R_C} + \frac{1}{jX_m} + \frac{1}{jX_2 + \frac{R_2}{s}}} \tag{1}$$

As stated by the project description, the utility load and the per-phase transformer can be represented as "the load in the secondary circuit seen from the source":

$$Z_{\text{utility load}} = \frac{Z_{L_a}}{N^2} = \frac{Z_{L_c}}{N^2} = \frac{Z_{L_c}}{N^2} \tag{2}$$

Based on the bus-tie breaker and the load feeders, the load impedance is calculated as the parallell load of the connected loads in port and starboard. If the bus-tie breaker is closed then the resulting load impedance was the total parallell impedance of the connected loads. The parallell loads consisted of: Z_{Motor1}, Z_{Motor2} and $Z_{\text{utility load}}$.

1.2 Generators

1.2.1 Equation for "Resulting Thevenin Voltage and Imepdance"

The circuit from the project description has three lines with a voltage source in series with an impedance each. By transforming each line to its Norten equivalent, and thereafter adding the lines together, we get:

$$I_{\text{Norton a-phase}} = \frac{\tilde{E}_1}{Z_{a1}} + \frac{\tilde{E}_2}{Z_{a2}} + \frac{\tilde{E}_3}{Z_{a3}} \tag{3}$$

$$Z_{\text{Thevenin}} = \frac{1}{\frac{1}{Z_{a1}} + \frac{1}{Z_{a2}} + \frac{1}{Z_{a3}}} \tag{4}$$

$$V_{\text{Thevenin a-phase}} = I_{\text{Norton a-phase}} Z_{\text{Thevenin}} \tag{5}$$

The b- and c-phase of the voltage was found by multiplying with a phase shift of -120 and 120 degrees respectively. Notice that the equations (3-5) are dependent on the bus-tie breaker and the load feeders. See the respective script in appendix for more detail.

1.3 MATLAB Programs

1.3.1 Node Analysis

By Node Analyses the following equations were used to find the bus phase voltages:

$$V_{\text{an}} = V_{\text{Thevenin a-phase}} \frac{Z_{\text{Load}}}{Z_{\text{Thevenin}} + Z_{\text{Load}}} \tag{6}$$

$$V_{\text{bn}} = V_{\text{Thevenin b-phase}} \frac{Z_{\text{Load}}}{Z_{\text{Thevenin}} + Z_{\text{Load}}} \tag{7}$$

$$V_{\text{cn}} = V_{\text{Thevenin c-phase}} \frac{Z_{\text{Load}}}{Z_{\text{Thevenin}} + Z_{\text{Load}}} \tag{8}$$

1.3.2 Electric parameters for loads

$$I_{\text{Motor1 1}} = V_{\text{an}}/Z_{\text{Motor1}} \tag{9}$$
$$I_{\text{Motor1 2}} = V_{\text{bn}}/Z_{\text{Motor1}} \tag{10}$$
$$I_{\text{Motor1 3}} = V_{\text{cn}}/Z_{\text{Motor1}} \tag{11}$$
$$I_{\text{Motor1 N}} = -I_{\text{Motor1 1}} - I_{\text{Motor1 2}} - I_{\text{Motor1 3}} \tag{12}$$

$$S_{\text{Motor1}} = \begin{bmatrix} V_{an} & V_{bn} & V_{cn} \end{bmatrix} \begin{bmatrix} I^*_{\text{Motor1 1}} & I^*_{\text{Motor1 2}} & I^*_{\text{Motor1 3}} \end{bmatrix}^T \tag{13}$$
$$P_{\text{Motor1}} = \text{real}(S_{\text{Motor1}}) \qquad Q_{\text{Motor1}} = \text{imag}(S_{\text{Motor1}}) \qquad \text{pf}_{\text{Motor1}} = \cos(\text{angle}(S_{\text{Motor1}})) \tag{14}$$

Similarly for Motor2 and Transformer1.

1.3.3 Electric parameters for generators

$$I_{\text{Generator1 1}} = \tilde{E}_1 e^{j\theta} - V_{an} \tag{15}$$
$$I_{\text{Generator1 2}} = \tilde{E}_1 e^{j\theta} - V_{bn} \tag{16}$$
$$I_{\text{Generator1 3}} = \tilde{E}_1 e^{j\theta} - V_{cn} \tag{17}$$

The nuetral current, the power (complex, active, reactive) and power factor are found in same manner as equations(12-14).Similarly for Generator 2 and 3. Look at the Matlab scripts for more detail.

2 Simulations cases

2.1 Case 1: All running - All breakers closed (Same Internal rms-voltage and phase)

$$\text{BT1} = \text{CB1} = \text{CB2} = \text{CB3} = \text{CB4} = \text{CB5} = \text{CB6} = 1$$

Given by task:

$$\tilde{E}_1 = \tilde{E}_2 = \tilde{E}_3 = 650V \qquad \theta_1 = \theta_2 = \theta_3 = 15\text{degrees}$$

2.1.1 a)

	Load
Load [Ohm]	0.19956+0.15426i
Thevenin impedance [Ohm]	4.7448e-05+0.20563i

Table 1: Load and Thevenin Impedance

	RMS	Angle
Thevenin [V][degrees]	650	15

Table 2: Load and Thevenin Impedance

2.1.2 b)

	rmsPort	anglePort	rmsStbd	angleStbd
$V_{an}[V]$	398.3787	-8.2821	398.3787	-8.2821
$V_{bn}[V]$	398.3787	-128.2821	398.3787	-128.2821
$V_{cn}[V]$	398.3787	111.7179	398.3787	111.7179
$V_{ab}[V]$	690.0122	21.7179	690.0122	21.7179
$V_{bc}[V]$	690.0122	-98.2821	690.0122	-98.2821
$V_{ca}[V]$	690.0122	141.7179	690.0122	141.7179

Table 3: The line-to-line and bus phase voltages in rms values

	Motor1	Motor2	Transformer1
$I_a[A]$	635.7399-517.907i	211.7769-365.425i	249.9379-252.5617i
$I_b[A]$	-766.3906-291.6134i	-422.3558-0.6916491i	-343.6938-90.17174i
$I_c[A]$	130.6507+809.5204i	210.5789+366.1167i	93.7559+342.7335i
$I_n[A]$	2.8422e-14+2.2737e-13i	-2.8422e-14+1.7053e-13i	0+5.6843e-14i
P [kW]	841.0323	313.3725	339.0745
Q [kVAR]	503.0679	395.7193	255.6694
pf [-]	0.85819	0.62082	0.79846

Table 4: Electric parameters for loads

	Generator1	Generator2	Generator3
$I_a[A]$	442.9115-458.4564i	442.9115-458.4564i	211.6317-218.9809i
$I_b[A]$	-618.4906-154.3444i	-618.4906-154.3444i	-295.4589-73.78793i
$I_c[A]$	175.5791+612.8008i	175.5791+612.8008i	83.82722+292.7689i
$I_n[A]$	-8.5265e-14+1.1369e-13i	-8.5265e-14+1.1369e-13i	-4.2633e-14+5.6843e-14i
P [kW]	602.7449	602.7449	287.9896
Q [kVAR]	465.9538	465.9538	222.549
$P/P_r[percentage]$	66.9717	66.9717	66.9121
$Q/Q_r[percentage]$	69.0302	69.0302	68.9433

Table 5: Electric parameters for generator

2.1.3 c)

	Supplied	Consumed
S [kVA]	1493.4793+1154.4566i	1493.4793+1154.4566i

Table 6: Total complex power supplied and consumed

2.1.4 d)

It can be verified from Table 5 that the active power is approx. equal in %-loading for the 3-gensets.

2.1.5 e)

It can be verified from Table 5 that the reactive power is approx. equal in %-loading for the 3-gensets.

2.1.6 f)

It can be verified from Table 5 that the current in the neutral wire for the 3 gensets is 0. I got a value with a factor 10^{-13}, which is approx 0.

2.2 Case 1: All running - All breakers closed (Same Internal rms-voltage, but different phase)

Given by task:
$\tilde{E}_1 = \tilde{E}_2 = \tilde{E}_3 = 650V$ $\theta_1 = 14$, $\theta_2 = 15$ and $\theta_3 = 16.5$ degrees

2.2.1 a)

	RMS	Angle
Thevenin [V][degrees]	649.9178	14.8856

Table 7: Thevenin voltage in rms with angle

2.2.2 b)

	Generator1	Generator2	Generator3
P/P$_r$[$percentage$]	64.5505	67.2737	71.2555
Q/Q$_r$[$percentage$]	70.3946	68.8602	66.3552

Table 8: Power delivered in %-loading by each generator

2.3 c)

The difference in phase angles mainly affected the sharing of active power. Genset1 went from 66.97 % to 64.55 % , Genset2 went from 66.97 % to 67.27 % and Genset3 went from 66.91 % to 71.25 % . While the biggest deviation in sharing of reactive power is a bit above 2 % .

2.4 Case 1: All running - All breakers closed (Different Internal rms-voltage, but same phase)

Given by task:
$\tilde{E}_1 = 640V$, $\tilde{E}_2 = 650V$ and $\tilde{E}_3 = 675V$ $\theta_1 = \theta_2 = \theta_3 = 15$ degrees

2.4.1 a)

	RMS	Angle
Thevenin [V][degrees]	650.7843	15.0001

Table 9: Thevenin voltage in rms with angle

2.4.2 b)

	Generator1	Generator2	Generator3
$P/P_r[percentage]$	66.0203	67.0522	69.5707
$Q/Q_r[percentage]$	65.7506	68.9464	76.84

Table 10: Power delivered in %-loading by each generator

2.4.3 c)

The difference in internal rms voltages mainly affected the sharing of reactive power between the generators. Generator3 had the biggest deviation and went from 69.94 % to 76.84 %, while the biggest deviation in sharing of active power was a bit below 3 %.

2.5 Case 2: 2-split system - Gen2 disconnected (Different Internal rms-voltage, but same phase)

$$CB1 = CB3 = CB4 = CB6 = 1$$
$$BT1 = CB2 = CB5 = 0$$

Given by task:
$\tilde{E}_1 = 710V$, $\tilde{E}_2 = 0$ and $\tilde{E}_3 = 696V$ $\quad \theta_1 = \theta_2 = \theta_3 = 20$ degrees

2.5.1 a)

	rmsPort	anglePort	rmsStbd	angleStbd
$V_a b[V]$	689.9549	19.6762	690.1776	24.2325
$V_b c[V]$	689.9549	-100.3238	690.1776	-95.7675
$V_c a[V]$	689.9549	139.6762	690.1776	144.2325

Table 11: Line-to-line voltage in rms with angle

2.5.2 b)

	Generator1	Generator2	Generator3
$P/P_r[percentage]$	93.4325	0	78.819
$Q/Q_r[percentage]$	74.5162	0	79.2416

Table 12: Power delivered in %-loading by each generator

2.6 Case 2: 2-split system - Gen2 disconnected (Same Internal rms-voltage and phase. (BT1 = closed))

$$\text{BT1} = \text{CB1} = \text{CB3} = \text{CB4} = \text{CB6} = 1$$
$$\text{CB2} = \text{CB5} = 0$$

Given by task:
$\tilde{E}_1 = \tilde{E}_3 = 705V$ and $\tilde{E}_2 = 0V$ $\theta_1 = \theta_2 = \theta_3 = 20$ degrees

2.6.1 a)

	rmsPort	anglePort	rmsStbd	angleStbd
$V_a b[V]$	690.034	21.1287	690.034	21.1287
$V_b c[V]$	690.034	-98.8713	690.034	-98.8713
$V_c a[V]$	690.034	141.1287	690.034	141.1287

Table 13: Line-to-line voltage in rms with angle

2.6.2 b)

	Generator1	Generator2	Generator3
$P/P_r[percentage]$	88.735	0	88.654
$Q/Q_r[percentage]$	76.0779	0	75.9786

Table 14: Power delivered in %-loading by each generator

2.6.3 c)

From Table 12 it can be seen that the load sharing between generator 1 and 3 is uneven in both active and reactive power. Generator 1 was working on 93.43 % while Generator 3 had 78.82 % on active power. However when the BT1 got closed and the rms internal voltages were set to be equal, the generators got a even loadsharing on both active and reactive power. This can be seen Table 14.

3 Appendix: Matlab-scripts

3.1 Resulting load impedances

```
function [Z_L_vec] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6)
global M1 M2 T1 Utility_load

%%%%%%%%%%%%%%%%%%%%%%%%%%
CLOSED = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%

% Resulting per-phase impedance of induction motor M1
Z_p_temp =  1/(1/(M1.R_c)+ 1/(j*M1.X_m)+ 1/(j*M1.X_2 + M1.R_2/M1.slip));
Z_m1 = M1.R_sm +  j*M1.X_sm + Z_p_temp;

% Resulting per-phase impedance of induction motor M2
Z_p_temp =  1/(1/(M2.R_c)+ 1/(j*M2.X_m)+ 1/(j*M2.X_2 + M2.R_2/M2.slip));
Z_m2 = M2.R_sm +  j*M2.X_sm + Z_p_temp;

% Utility load and per-phase transformer seen from the source
 Z_utility = Utility_load.Z_La/(T1.N^2)
 if (BT1 == CLOSED)
     if CB4 == 0 && CB5 == 0 && CB6 == 0
         Z = inf+j*inf;
     else
         Z = 1 /( (CB4/Z_m1) + (CB5/Z_m2) + (CB6/Z_utility));
     end
     Z_L_vec = [Z; Z];
 else %BT1 == OPEN
     if CB4 == 0
         Z_port = inf+j*inf;
     else
         Z_port = Z_m1;
     end
     if CB5 == 0 && CB6 == 0
         Z_starboard = inf + j*inf;
     else
         Z_starboard = 1/((CB5/Z_m2) + (CB6/Z_utility))
     end
     Z_L_vec = [Z_port;Z_starboard];
end
end
```

3.2 Thévenin equivalent supply circuit

```
%theta is taken in as degrees
function [V_T_vec, Z_T_vec] = getTheveninEquivalents(BT1, CB1, CB2,
    CB3,E_tilde_vec, theta_vec)
global G1 G2 G3

%%%%%%%%%%%%%%%%
CLOSED = 1;
%%%%%%%%%%%%%%%%
%Internal impedance of generators
Z_a1 = G1.R_a1 + j*G1.X_s1;
Z_a2 = G2.R_a1 + j*G2.X_s1;
Z_a3 = G3.R_a1 + j*G3.X_s1;

% Internal rms voltages and phase angles
E_a = [ E_tilde_vec(1)*(cosd(theta_vec(1)) + j*sind(theta_vec(1)));
        E_tilde_vec(2)*(cosd(theta_vec(2)) + j*sind(theta_vec(2)));
        E_tilde_vec(3)*(cosd(theta_vec(3)) + j*sind(theta_vec(3)));
      ];

% From voltage to current (Thev-Nort) transformation
I_a1 = CB1 * E_a(1)/Z_a1;
I_a2 = CB2 * E_a(2)/Z_a2;
I_a3 = CB3 * E_a(3)/Z_a3;

if (BT1 == CLOSED)
    if CB1 == 0 && CB2 == 0 && CB3 == 0
        Z_T     = inf + j*inf;
        I_a     = inf + j*inf;
        V_Ta    = 0 + j*0;
    else
        Z_T     = 1/((CB1/Z_a1)+(CB2/Z_a2)+(CB3/Z_a3));
        I_a     = I_a1+I_a2+I_a3;
        V_Ta    = Z_T*I_a;

    end
    V_Tb = V_Ta * (cosd(-120) + j*sind(-120));
    V_Tc = V_Ta * (cosd(120) + j*sind(120));

    Z_T_vec = [Z_T;Z_T];
    V_T_vec = [V_Ta,V_Tb,V_Tc; V_Ta,V_Tb,V_Tc];

else %BT1 is open
    if CB1 == 0
        Z_T_port      = inf + j*inf;
        V_Ta_port     = 0 + j*0;
    else
        Z_T_port      = Z_a1;
        V_Ta_port     = I_a1*Z_T_port;
    end
```

```
    V_Tb_port = V_Ta_port * (cosd(-120) + j*sind(-120));
    V_Tc_port = V_Ta_port * (cosd(120) + j*sind(120));


    if CB2 == 0 && CB3 == 0
        Z_T_starboard   = inf + j*inf;
        V_Ta_starboard  = 0 + j*0;
    else
        Z_T_starboard   = 1/((CB2/Z_a2)+(CB3/Z_a3));
        I_a_starboard   = CB2*I_a2+CB3*I_a3;
        V_Ta_starboard  = I_a_starboard*Z_T_starboard;
    end
    V_Tb_starboard = V_Ta_starboard * (cosd(-120) + j*sind(-120));
    V_Tc_starboard = V_Ta_starboard * (cosd(120) + j*sind(120));

    V_Ta = [V_Ta_port; V_Ta_starboard];
    V_Tb = [V_Tb_port; V_Tb_starboard];
    V_Tc = [V_Tc_port; V_Tc_starboard];

    Z_T_vec = [Z_T_port; Z_T_starboard];
    V_T_vec = [V_Ta,V_Tb,V_Tc];
end
end
```

3.3 Node voltage analysis

```
%uses getResultingPerPhaseLoadImpedance() and getTheveninEquivalents
    ()
function [V_bus_phase_vec] = nodeVoltage(BT1,CB1,CB2,CB3,CB4,CB5,CB6,
    E_vec,theta_vec)

[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);

V_an = V_T(:,1).*(Z_L ./(Z_T+Z_L));
V_bn = V_T(:,2).*(Z_L ./(Z_T+Z_L));
V_cn = V_T(:,3).*(Z_L ./(Z_T+Z_L));
V_bus_phase_vec = [V_an,V_bn,V_cn];
end
```

3.4 Electric parameters for loads

```
function [I_M, I_T1, P_M_T,Q_M_T, pf_M_T] = getElectricParamForLoads(CB4,CB5,CB6,V_bus_phase_vec)
global M1 M2 T1 Utility_load

%The bus phase voltages
V_an = V_bus_phase_vec(:,1);
V_bn = V_bus_phase_vec(:,2);
V_cn = V_bus_phase_vec(:,3);

% Resulting per-phase impedance of induction motor M1
Z_p_temp     = 1/(1/(M1.R_c)+ 1/(j*M1.X_m)+ 1/(j*M1.X_2 + M1.R_2/M1.slip));
Z_m1         = [
                 M1.R_sm +   j*M1.X_sm + Z_p_temp;
                 M1.R_sm +   j*M1.X_sm + Z_p_temp;
               ];

% Resulting per-phase impedance of induction motor M2
Z_p_temp     = 1/(1/(M2.R_c)+ 1/(j*M2.X_m)+ 1/(j*M2.X_2 + M2.R_2/M2.slip));
Z_m2         = [
                 M2.R_sm +   j*M2.X_sm + Z_p_temp;
                 M2.R_sm +   j*M2.X_sm + Z_p_temp;
               ];
% Utility load and per-phase transformer seen from the source
 Z_utility   = [
                 Utility_load.Z_La/(T1.N^2);
                 Utility_load.Z_La/(T1.N^2);
               ];

%%%%%%%%%%% Motor M1 %%%%%%%%%%%%%%
 if CB4 == 0
     I_M1 = [[0;0], [0;0], [0;0], [0;0]];
 else
     I_M1 = [CB4*V_an./Z_m1, CB4*V_bn./Z_m1, CB4*V_cn./Z_m1, [0;0]];
     I_M1(:,4) = -I_M1(:,1) - I_M1(:,2) - I_M1(:,3);
end
S_M1 = (V_bus_phase_vec*(I_M1(1,1:3))')/1000; % [kVA]
P_M1 = real(S_M1);
Q_M1 = imag(S_M1);
pf_M1 = cos(angle(S_M1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% Motor M2 %%%%%%%%%%%%%%
 if CB5 == 0
     I_M2 = [[0;0], [0;0], [0;0], [0;0]];
 else
     I_M2 = [CB5*V_an./Z_m2, CB5*V_bn./Z_m2, CB5*V_cn./Z_m2, [0;0]];
     I_M2(:,4) = -I_M2(:,1) - I_M2(:,2) - I_M2(:,3);
```

```
end
S_M2 = (V_bus_phase_vec*(I_M2(2,1:3))')/1000; % [kVA]
P_M2 = real(S_M2);
Q_M2 = imag(S_M2);
pf_M2 = cos(angle(S_M2));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Transformer T1 %%%%%%%%%%
if CB6 == 0
    I_T1 = [[0;0], [0;0], [0;0], [0;0]];
else
    I_T1= [CB6*V_an./Z_utility, CB6*V_bn./Z_utility, CB6*V_cn./
        Z_utility, [0;0]];
    I_T1(:,4) = -I_T1(:,1) - I_T1(:,2) - I_T1(:,3);
end
S_T1 = (V_bus_phase_vec*(I_T1(2,1:3))')/1000; % [kVA]
P_T1 = real(S_T1);
Q_T1 = imag(S_T1);
pf_T1 = cos(angle(S_T1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I_M = [I_M1,I_M2];
P_M_T = [P_M1,P_M2,P_T1];
Q_M_T = [Q_M1,Q_M2,Q_T1];
pf_M_T = [pf_M1,pf_M2,pf_T1];
end
```

3.5 Electric parameters for generators

```
function [I_G, P_G_and_per,Q_G_and_per] = getElectricParamForGen(CB1,
    CB2,CB3,V_bus_phase_vec,E_vec, theta_vec)
global G1 G2 G3

%The bus phase voltages
V_an = V_bus_phase_vec(:,1);

%Internal impedance of generators
Z_a1 = G1.R_a1 + j*G1.X_s1;
Z_a2 = G2.R_a1 + j*G1.X_s1;
Z_a3 = G3.R_a1 + j*G3.X_s1;

Sn = [G1.S_N; G2.S_N; G3.S_N];
pf = [G1.pf; G2.pf; G3.pf];

Pn = Sn.*pf;
Qn = Sn.*[sin(acos(G1.pf));sin(acos(G2.pf));sin(acos(G3.pf))];

%%%%%%%%%%%% Storage for the currents %%%%%%%%%%%%%%%%%%
I_G1 = zeros(2,4);
I_G2 = zeros(2,4);
I_G3 = zeros(2,4);

E_a = [ E_vec(1)*(cosd(theta_vec(1)) + j*sind(theta_vec(1)));
        E_vec(2)*(cosd(theta_vec(2)) + j*sind(theta_vec(2)));
        E_vec(3)*(cosd(theta_vec(3)) + j*sind(theta_vec(3)));
      ];

if CB1 == 0
    V_G1 = 0;
else
    V_G1 = E_a(1)-V_an;
end
if CB2 == 0
    V_G2 = 0;
else
    V_G2 = E_a(2)-V_an;
end
if CB3 == 0
    V_G3 = 0;
else
    V_G3 = E_a(3)-V_an;
end

I_G1(:,1) = CB1*V_G1/Z_a1;
I_G1(:,2) = (CB1*V_G1/Z_a1) *(cosd(-120)+j*sind(-120));
I_G1(:,3) = (CB1*V_G1/Z_a1) *(cosd(120)+j*sind(120));
I_G1(:,4) = - I_G1(:,1) - I_G1(:,2) - I_G1(:,3);

```

```
I_G2(:,1) =  CB2*V_G2/Z_a2;
I_G2(:,2) = (CB2*V_G2/Z_a2) *(cosd(-120)+j*sind(-120));
I_G2(:,3) = (CB2*V_G2/Z_a2) *(cosd(120)+j*sind(120));
I_G2(:,4) = - I_G2(:,1) - I_G2(:,2) - I_G2(:,3);

I_G3(:,1) =  CB3*V_G3/Z_a3;
I_G3(:,2) = (CB3*V_G3/Z_a3) *(cosd(-120)+j*sind(-120));
I_G3(:,3) = (CB3*V_G3/Z_a3) *(cosd(120)+j*sind(120));
I_G3(:,4) = - I_G3(:,1) - I_G3(:,2) - I_G3(:,3);


S_G1 = (V_bus_phase_vec*(I_G1(1,1:3))')/1000;
S_G2 = (V_bus_phase_vec*(I_G2(2,1:3))')/1000;
S_G3 = (V_bus_phase_vec*(I_G3(2,1:3))')/1000;

P_G1 = real(S_G1);
P_G2 = real(S_G2);
P_G3 = real(S_G3);

P_G1_per = 100* (P_G1/Pn(1,1));
P_G2_per = 100* (P_G2/Pn(2,1));
P_G3_per = 100* (P_G3/Pn(3,1));


Q_G1 = imag(S_G1);
Q_G2 = imag(S_G2);
Q_G3 = imag(S_G3);

Q_G1_per = 100* (Q_G1/Qn(1,1));
Q_G2_per = 100* (Q_G2/Qn(2,1));
Q_G3_per = 100* (Q_G3/Qn(3,1));


I_G = [I_G1,I_G2,I_G3];
P_G_and_per = [P_G1,P_G1_per,P_G2,P_G2_per,P_G3,P_G3_per];
Q_G_and_per = [Q_G1,Q_G1_per,Q_G2,Q_G2_per,Q_G3,Q_G3_per];
end
```

3.6 Overall program

```
function [V_bus_phase_vec,V_line_to_line_vec, ...
    I_M, I_T, P_M_T,Q_M_T, pf_M_T, ... %script 4
    I_G, P_G_and_per,Q_G_and_per, ... %script 5
    S_consumed, S_supplied] = overallProgram(BT1,CB1,CB2,CB3,CB4,CB5,
        CB6,E_vec,theta_vec)


[V_bus_phase_vec] = nodeVoltage(BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,
    theta_vec);

[I_M, I_T, P_M_T,Q_M_T, pf_M_T] = getElectricParamForLoads(CB4,CB5,
    CB6,V_bus_phase_vec);

[I_G, P_G_and_per,Q_G_and_per] = getElectricParamForGen(CB1,CB2,CB3,
    V_bus_phase_vec,E_vec,theta_vec);

P_consumed = P_M_T(1,1) + P_M_T(2,2) + P_M_T(2,3);
Q_consumed = Q_M_T(1,1) + Q_M_T(2,2) + Q_M_T(2,3);

P_supplied = P_G_and_per(1,1) + P_G_and_per(2,3) + P_G_and_per(2,5);
Q_supplied = Q_G_and_per(1,1) + Q_G_and_per(2,3) + Q_G_and_per(2,5);

S_consumed = [P_consumed,Q_consumed];
S_supplied = [P_supplied,Q_supplied];

V_an = V_bus_phase_vec(:,1);
V_bn = V_bus_phase_vec(:,2);
V_cn = V_bus_phase_vec(:,3);

%%% line-to-line-voltage %%%%
V_ab = V_an - V_bn;
V_bc = V_bn - V_cn;
V_ca = V_cn - V_an;
V_line_to_line_vec = [V_ab,V_bc,V_ca];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
```

3.7 Marine Power Plant Constants

```
global M1 M2 T1 Utility_load G1 G2 G3
G1 = struct('S_N', 1125, 'pf', 0.8, 'R_a1', 0.0001, 'X_s1', 0.5095, 'rpm', 1800, 'poles', 4, 'VAC_LtL', 690);
G2 = struct('S_N', 1125, 'pf', 0.8, 'R_a1', 0.0001, 'X_s1', 0.5095, 'rpm', 1800, 'poles', 4, 'VAC_LtL', 690);
G3  = struct('S_N', 538, 'pf', 0.8, 'R_a1', 0.0004, 'X_s1', 1.0665, 'rpm', 1800, 'poles', 4, 'VAC_LtL', 690);
M1  = struct('slip', 0.02, 'R_sm', 0.4169, 'X_sm', 0.2429,'X_m', 0.0065, 'R_c', 290, 'R_2', 0.023, 'X_2', 0.264);
M2  = struct('slip', 0.025, 'R_sm', 0.5854, 'X_sm', 0.7255,'X_m', 0.0140, 'R_c', 325, 'R_2', 0.026, 'X_2', 0.295);
T1  = struct('N',0.57735, 'Connection', 'Ynyn');
Utility_load = struct('Z_La',0.2984+j*0.2250, 'Z_Lb',0.2984+j*0.2250, 'Z_Lc',0.2984+j*0.2250);
```

3.8 Simulation of case 1.1

```
clc
clear all;
close all;

%% Values
%%%%% Power Plant Constants %%%%%%%%%
Param
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Power Plant Variables %%%%%%%%%
BT1 = 1;     CB1 = 1;     CB2 = 1;     CB3 = 1;
CB4 = 1;     CB5 = 1;     CB6 = 1;

E1 = 650;    E2 = 650;    E3 = 650;

theta_1 = 15;  theta_2 = 15; theta_3 = 15;

E_vec = [E1;E2;E3];
theta_vec = [theta_1;theta_2;theta_3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Functions
[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);
[V_bus_phase_vec,V_line_to_line_vec,I_M,I_T1, P_M_T,Q_M_T, pf_M_T,I_G
    ,P_G_and_per,Q_G_and_per,S_consumed, S_supplied] = overallProgram(
    BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,theta_vec);

%% Table
format long
%%%% Verification of program %%%%
Z_L = Z_L(1);
Z_TPhaseA = Z_T(1);
V_TPhaseA = V_T(1,1);

Thevenin = transpose([Z_TPhaseA, V_TPhaseA]);
Load = transpose([Z_L, inf]);
T_verification = table(Load, Thevenin,...
    'RowNames',{'Imedance [Ohm]' 'Voltage [V]'})
table2csv(T_verification,'case_5_1_1_verification.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format short
%%%% RMS of the bus voltages %%%%
V_bus_rms   = zeros(2,3);    V_bus_angle = zeros(2,3);
V_line_rms  = zeros(2,3);    V_line_angle = zeros(2,3);
for i_ = 1:1:2
    for j_ = 1:1:3
        V_bus_rms(i_,j_) = abs(V_bus_phase_vec(i_,j_));
```

```
            V_line_rms(i_,j_) = abs(V_line_to_line_vec(i_,j_));

            V_bus_angle(i_,j_) = angle(V_bus_phase_vec(i_,j_)) * 180/pi;
            V_line_angle(i_,j_) = angle(V_line_to_line_vec(i_,j_)) * 180/pi;
        end
end

rmsPort = transpose([V_bus_rms(1,:),V_line_rms(1,:)]);
rmsStbd = transpose([V_bus_rms(2,:),V_line_rms(2,:)]);
anglePort = transpose([V_bus_angle(1,:),V_line_angle(1,:)]);
angleStbd = transpose([V_bus_angle(2,:),V_line_angle(2,:)]);

T_voltage = table(rmsPort,anglePort,rmsStbd,angleStbd,...
    'RowNames',{'V_{an} [V]' 'V_{bn} [V]' 'V_{cn} [V]' 'V_{ab} [V]' 'V_{bc} [V]' 'V_{ca} [V]'})

table2csv(T_voltage,'case_5_1_1_voltage.csv')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Motor Table %%%%%%%%%%%%%%%%%%
Motor1 = transpose([I_M(1,1:4),P_M_T(1,1),Q_M_T(1,1),pf_M_T(1,1)]);
Motor2 = transpose([I_M(1,5:8),P_M_T(1,2),Q_M_T(1,2),pf_M_T(1,2)]);
Transformer1 = transpose([I_T1(1,:),P_M_T(1,3),Q_M_T(1,3),pf_M_T(1,3)]);

T_motor = table(Motor1,Motor2,Transformer1,...
    'RowNames',{'I_a [A]' 'I_b [A]' 'I_c [A]' 'I_n [A]' 'P [kW]' 'Q [kVAR]' 'pf [-]'})

table2csv(T_motor,'case_5_1_1_motor.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Generator Table %%%%%%%%%%%%%%%%%%
Generator1 = transpose([I_G(1,1:4),P_G_and_per(1,1),Q_G_and_per(1,1),P_G_and_per(1,2),Q_G_and_per(1,2)]);
Generator2 = transpose([I_G(1,5:8),P_G_and_per(1,3),Q_G_and_per(1,3),P_G_and_per(1,4),Q_G_and_per(1,4)]);
Generator3 = transpose([I_G(1,9:12),P_G_and_per(1,5),Q_G_and_per(1,5),P_G_and_per(1,6),Q_G_and_per(1,6)]);

T_generator = table(Generator1,Generator2,Generator3,...
    'RowNames',{'I_a [A]' 'I_b [A]' 'I_c [A]' 'I_n [A]' 'P [kW]' 'Q [kVAR]' 'P/P_r [percentage]' 'Q/Q_r [percentage]'})

table2csv(T_generator,'case_5_1_1_generator.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format long
%%%%%%%% Total effect %%%%%%%%%%%%%%%%%%%%%
```

```
Supplied = S_supplied(1) + j*S_supplied(2)
Consumed = S_consumed(1) + j*S_consumed(2)
T_power = table(Supplied, Consumed,...
    'RowNames',{'S [kVA]'})
table2csv(T_power,'case_5_1_1_power.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short
```

3.9 Simulation of case 1.2

```
clc
clear all;
close all;

%% Values
%%%%% Power Plant Constants %%%%%%%%%
Param
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Power Plant Variables %%%%%%%%%
BT1 = 1;    CB1 = 1;    CB2 = 1;    CB3 = 1;
CB4 = 1;    CB5 = 1;    CB6 = 1;

E1 = 650;   E2 = 650;   E3 = 650;

theta_1 = 14;  theta_2 = 15; theta_3 = 16.5;

E_vec = [E1;E2;E3];
theta_vec = [theta_1;theta_2;theta_3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Functions
[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);
[V_bus_phase_vec,V_line_to_line_vec,I_M,I_T1, P_M_T,Q_M_T, pf_M_T,I_G
    ,P_G_and_per,Q_G_and_per,S_consumed, S_supplied] = overallProgram(
    BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,theta_vec);

%% Table
format short
%%%% Verification of program %%%%
RMS = abs(V_T(1,1));
Angle = angle(V_T(1,1)) * 180/pi;
T_verification = table(RMS, Angle,...
    'RowNames',{'Thevenin [V][degrees]'})
table2csv(T_verification,'case_5_1_2_verification.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% Generator Power Table %%%%%%%%%%%%%%%%
Generator1 = transpose([P_G_and_per(1,2),Q_G_and_per(1,2)]);
Generator2 = transpose([P_G_and_per(1,4),Q_G_and_per(1,4)]);
Generator3 = transpose([P_G_and_per(1,6),Q_G_and_per(1,6)]);
```

```
T_generator = table(Generator1,Generator2,Generator3,...
    'RowNames',{'P/P_r [percentage]' 'Q/Q_r [percentage]'})

table2csv(T_generator,'case_5_1_2_generator_power.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.10 Simulation of case 1.3

```
clc
clear all;
close all;

%% Values
%%%%% Power Plant Constants %%%%%%%%%
Param
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Power Plant Variables %%%%%%%%%
BT1 = 1;    CB1 = 1;    CB2 = 1;    CB3 = 1;
CB4 = 1;    CB5 = 1;    CB6 = 1;

E1 = 640;   E2 = 650;   E3 = 675;

theta_1 = 15;  theta_2 = 15; theta_3 = 15;

E_vec = [E1;E2;E3];
theta_vec = [theta_1;theta_2;theta_3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Functions
[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);
[V_bus_phase_vec,V_line_to_line_vec,I_M,I_T1, P_M_T,Q_M_T, pf_M_T,I_G
    ,P_G_and_per,Q_G_and_per,S_consumed, S_supplied] = overallProgram(
    BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,theta_vec);

%% Table
format short
%%%% Verification of program %%%%
RMS = abs(V_T(1,1));
Angle = angle(V_T(1,1)) * 180/pi;
T_verification = table(RMS, Angle,...
    'RowNames',{'Thevenin [V][degrees]'})
table2csv(T_verification,'case_5_1_3_verification.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%% Generator Power Table %%%%%%%%%%%%%%%%%
Generator1 = transpose([P_G_and_per(1,2),Q_G_and_per(1,2)]);
Generator2 = transpose([P_G_and_per(1,4),Q_G_and_per(1,4)]);
Generator3 = transpose([P_G_and_per(1,6),Q_G_and_per(1,6)]);

T_generator = table(Generator1,Generator2,Generator3,...
    'RowNames',{'P/P_r [percentage]' 'Q/Q_r [percentage]'})

table2csv(T_generator,'case_5_1_3_generator_power.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.11 Simulation of case 2.1

```
clc
clear all;
close all;

%% Values
%%%%% Power Plant Constants %%%%%%%%%
Param
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Power Plant Variables %%%%%%%%%
BT1 = 0;     CB1 = 1;     CB2 = 0;     CB3 = 1;
CB4 = 1;     CB5 = 0;     CB6 = 1;

E1 = 710;    E2 = 0;    E3 = 696;

theta_1 = 20;  theta_2 = 20; theta_3 = 20;

E_vec = [E1;E2;E3];
theta_vec = [theta_1;theta_2;theta_3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Functions
[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);
[V_bus_phase_vec,V_line_to_line_vec,I_M,I_T1, P_M_T,Q_M_T, pf_M_T,I_G
    ,P_G_and_per,Q_G_and_per,S_consumed, S_supplied] = overallProgram(
    BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,theta_vec);

%% Table
%%%% RMS of the complex variables %%%%
V_line_rms  = zeros(2,3);    V_line_angle = zeros(2,3);
for i = 1:1:2
    for j = 1:1:3
        V_line_rms(i,j) = abs(V_line_to_line_vec(i,j));


        V_line_angle(i,j) = angle(V_line_to_line_vec(i,j)) * 180/pi;
    end
end

rmsPort = transpose(V_line_rms(1,:));
rmsStbd = transpose(V_line_rms(2,:));
anglePort = transpose(V_line_angle(1,:));
angleStbd = transpose(V_line_angle(2,:));

T_voltage = table(rmsPort,anglePort,rmsStbd,angleStbd,...
    'RowNames',{'V_ab [V]' 'V_bc [V]' 'V_ca [V]'})

table2csv(T_voltage,'case_5_2_1_voltage.csv')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Generator Power Table %%%%%%%%%%%%%%%%%%%
Generator1 = transpose([P_G_and_per(1,2),Q_G_and_per(1,2)]);
Generator2 = transpose([P_G_and_per(2,4),Q_G_and_per(2,4)]);
Generator3 = transpose([P_G_and_per(2,6),Q_G_and_per(2,6)]);

T_generator = table(Generator1,Generator2,Generator3,...
    'RowNames',{'P/P_r [percentage]' 'Q/Q_r [percentage]'})

table2csv(T_generator,'case_5_2_1_generator_power.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.12 Simulation of case 2.2

```
clc
clear all;
close all;

%% Values
%%%%% Power Plant Constants %%%%%%%%%
Param
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Power Plant Variables %%%%%%%%%
BT1 = 1;     CB1 = 1;     CB2 = 0;     CB3 = 1;
CB4 = 1;     CB5 = 0;     CB6 = 1;

E1 = 705;    E2 = 0;    E3 = 705;

theta_1 = 20;  theta_2 = 20; theta_3 = 20;

E_vec = [E1;E2;E3];
theta_vec = [theta_1;theta_2;theta_3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Functions
[Z_L] = getResultingPerPhaseLoadImpedance(BT1, CB4, CB5, CB6);
[V_T, Z_T] = getTheveninEquivalents(BT1, CB1, CB2, CB3,E_vec,
    theta_vec);
[V_bus_phase_vec,V_line_to_line_vec,I_M,I_T1, P_M_T,Q_M_T, pf_M_T,I_G
    ,P_G_and_per,Q_G_and_per,S_consumed, S_supplied] = overallProgram(
    BT1,CB1,CB2,CB3,CB4,CB5,CB6,E_vec,theta_vec);

%% Table
%%%% RMS of the complex variables %%%%
V_line_rms  = zeros(2,3);    V_line_angle = zeros(2,3);
for i = 1:1:2
    for j = 1:1:3
        V_line_rms(i,j) = abs(V_line_to_line_vec(i,j));


        V_line_angle(i,j) = angle(V_line_to_line_vec(i,j)) * 180/pi;
    end
end

rmsPort = transpose(V_line_rms(1,:));
rmsStbd = transpose(V_line_rms(2,:));
anglePort = transpose(V_line_angle(1,:));
angleStbd = transpose(V_line_angle(2,:));

T_voltage = table(rmsPort,anglePort,rmsStbd,angleStbd,...
    'RowNames',{'V_ab [V]' 'V_bc [V]' 'V_ca [V]'})

table2csv(T_voltage,'case_5_2_2_voltage.csv')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Generator Power Table %%%%%%%%%%%%%%%%%%%
Generator1 = transpose([P_G_and_per(1,2),Q_G_and_per(1,2)]);
Generator2 = transpose([P_G_and_per(2,4),Q_G_and_per(2,4)]);
Generator3 = transpose([P_G_and_per(2,6),Q_G_and_per(2,6)]);

T_generator = table(Generator1,Generator2,Generator3,...
    'RowNames',{'P/P_r [percentage]' 'Q/Q_r [percentage]'})

table2csv(T_generator,'case_5_2_2_generator_power.csv')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```