Power Systems Laboratory

User Manual

Department of Electrical and Computer Engineering

University of Minnesota

Revised : September 13, 2010

Textbook: First Course in Power Systems by Ned Mohan, www.mnpere.com.

Simulation Files: The simulation files mentioned in this lab manual are taken from the CD that accompanies the above Textbook.

Video Clips: The video clips mentioned in this lab manual are on the following website: www.ece.umn.edu/groups/power. These are designed to show the usage of the following software programs: PSCAD-EMTDC, PowerWorld, and MATLAB/Simulink.

Table of Contents

Laboratory Experiment 1: Visit local substation

Visit to a Local Substation or a Generating Plant

Objective: To see firsthand apparatus that we will be studying in this course and learn about their role in operation and protection of power systems.

Laboratory Task: Visit a local substation.

Report:
Write a few sentences about each apparatus you saw, include its photograph if you were allowed to take it (of course with permission and always reference the source), and state its role, as you understand it at this stage in your study, in operation and protection of power systems. State the approximate physical size and the electric ratings in terms of voltage, current, power, kVA etc. These apparatus may include transmission line towers and their structure, transmission line conductors, their size and bundling, transformers, circuit breakers, surge arresters, relays, line traps for line-carrier communication, microwave towers, bus bars and their arrangement, substation grounding, battery backup as uninterruptible power supplies and so on.

Note:
It is not always possible to arrange such a visit due to a host of reasons. In the following pages, photographs taken during a visit on May 16, 2006 to an HVDC substation are attached. This visit was organized by Mr. Jack Christofersen, and hosted and conducted by Mr. David Eisenschenk of the Great River Energy Cooperative. These photographs were taken by Prof. Bruce Wollenberg, who has kindly given his permission for them to be included here.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Laboratory Experiment 2: Intro to PSCAD/EMTDC

Familiarization with PSCAD/EMTDC and Understanding of Reactive Power and Power Factor Correction in AC Circuits

Objectives:

1. Learn the usage of PSCAD/EMTDC in modeling of ac circuits and plotting of results.
2. Understanding reactive power and power factor in single-phase and three-phase circuits.

Install PSCAD-EMTDC and PowerWorld: see video clip# 1.

Laboratory Tasks and Report:

1. Familiarization with PSCAD/EMTDC:
   Read ahead the "Simple Guide to using PSCAD/EMTDC" below to model ac circuits for this experiment and plot their results.
2. Single-Phase AC Circuit in Steady State: **(RLC.psc**; see video clip# 2**)**
   a. Model a single-phase ac circuit where a voltage source $v_s(t)$ at the fundamental frequency of 60 Hz is supplying an inductive series *R*-*L* load, with an impedance of $10\angle 30^0\,\Omega$. $\bar{V}_s = 132.7\angle 0^0\,kV(rms)$. Plot the waveforms for the voltage $v_s(t)$, the current $i_s(t)$, the instantaneous power $p(t) = v_s(t)i_s(t)$ and the average power P delivered to the load; all in the same plot.
   b. Calculate a capacitive reactance X_C, to be connected in parallel with the load, to bring the overall power factor seen from the source to unity. Plot $v_s(t)$ and $i_s(t)$, the instantaneous power $p(t) = v_s(t)i_s(t)$ and the average power P delivered to the load; all in the same plot.
3. Three-Phase AC Circuit in Steady State:
   a. Connect a three-phase circuit in a balanced-wye where the per-phase circuit is the same as the single-phase circuit above. Calculate $v_a(t)$, $i_a(t)$, $p_a(t) = v_a(t)i_a(t)$, $p_b(t) = v_b(t)i_b(t)$, $p_c(t) = v_c(t)i_c(t)$, and the sum of the three instantaneous power $p_s(t) = p_a(t) + p_b(t) + p_c(t)$.
   b. Connect the capacitive reactance X_C calculated in step 2b, in parallel with the load in each phase, to bring the per-phase power factor to unity. Plot $v_a(t)$ and $i_a(t)$, the instantaneous power $p(t) = v_s(t)i_s(t)$ and the average power P delivered to the load; all in the same plot.

Simple Guide to using PSCAD/EMTDC

1) Launch PSCAD (student version) from Start Menu.
2) Creating new project:
   Click on **File/New/Case** – A new project entitled ‘noname’ appears in the left workspace window, indicating that a new project is created.

3) Setting active project:
   In the workspace window, right click on the title of an inactive project and select **Set as active.**

4) Saving active project:
   Click on **File/Save Active Project.** Select appropriate folder and save the project as ‘Lab1’ or any other name.

5) Adding components to a project:
   Double click on master library in left top workspace. Navigate to the area containing desired component. Right click on component and select **Copy**. Open the project where you wish to add the component (double click on ‘project name’), right click over blank area and select **Paste**. (Note: There are many other ways to add a component to a project)

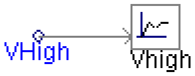
6) Setting Properties:
   To set the properties double click on any component and change the parameters.

   At the top of the parameter dialog is a drop list, which contains list of all parameter dialog pages. If only one page exists, then the drop list will be disabled. For e.g. if you double click on resistor, it will ask for only resistance value.

7) Making connections between components:
   Click **Wire Mode** button in the main toolbar. Move the mouse pointer onto the project page. The mouse pointer will have turned into a pencil, which indicates you are in Wire mode. To draw a wire, move the cursor to the node where you want line to start and left click. Move the cursor to where you want the line to end and right-click to complete the wire. Multi-segment Wires may be built by continuing to left click at different points. To turn off Wire Mode, press **Esc** key.

8) Measurement:
   To measure currents and voltages **ammeter** and **voltmeter** are provided on toolbar on right. Ammeter should be connected in series.

   To plot currents and voltages use **output channel** and **data signal label** on toolbar, as shown in the fig below



   where VHigh, is **data signal label** and is same signal name given in voltmeter or ammeter. Voltmeter /Ammeter signal name and data label signal name should match. In the **output channel** parameter dialog give title, unit, scale factor and min/max limits.

9) Adding a Graph Frame:
Right click on the Output Channel component. Select **Input/Output Reference/Add Overlay Graph with Signal.** This will create a new graph frame, overlay graph and a curve simultaneously. For adding more graphs on same graph frame, right click on graph frame and click **Add Overlay Graph (Analog).** This will add another graph on same frame. To put a curve on this graph Ctrl+click on output channel and drag it on the graph. Curve corresponding to that output channel will be added on to graph. When you run the simulation curves will be automatically plotted on this graph.
Press **Y** and **X** buttons to see complete curve (zoom out).

10) Setting time step and simulation time:
Right click on blank space in project, select **Project Settings**. In runtime tab you can set simulation time, time step and plot step.

11) To simulate the project:
Click on **Build/R.**

Laboratory Experiment 3: Transmission Line and Modeling

Obtaining Parameters of a 345 kV Transmission Line and Modeling it in PSCAD/EMTDC

Objectives: Obtaining the parameters of a 345 kV transmission line and modeling it in PSCAD/EMTDC.

Laboratory Tasks and Report:

Tasks

1. Consider a 345-kV transmission line consists of three-conductor-flat towers shown in Fig. 4-8. This transmission system consists of a single-conductor per phase, which is a Bluebird ACSR conductor with a diameter of 1.762 inches. The *PSCAD/EMTDC* file for this 345-kV single-conductor line is **LineParameters.psc** (see video clip# 3), which is located in this folder. Double click on it to open it and execute it to calculate line constants. Compare the results with those given in Example 4-2.

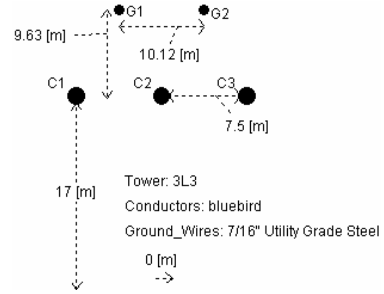


Fig. 4-8 A 345-kV, single-conductor per phase, transmission system.

2. The *PSCAD/EMTDC* file for a 345-kV double-conductor line is **LineParameters_Bundled.psc**, which is located in this folder. Double click on it to open it and execute it to calculate line constants. Compare the results with those in Task 1.
3. A 200 km long 345-kV line has the parameters given in the Table below. Neglect the resistance. Measure the reactive power at both ends under the following two levels of loading if both ends are held at the voltages of 1 per unit: (a) 1.5 times *SIL*, and (b) 0.75 times *SIL*. The *PSCAD/EMTDC* file for modeling this transmission line is **TransmissionLine.psc** (see video clip# 4), which is located in this folder. Double click on it to open it and execute it.

Table 4-1

Transmission Line Parameters with Bundled Conductors at 60 Hz

Nominal Voltage	$R(\Omega/km)$	$\omega L(\Omega/km)$	$\omega C(\mu\mho/km)$	$Z_c(\Omega)$	$SIL(MW)$
345 kV	0.037	0.376	4.518	280 (use 288.48)	425 MW (use 412.16)

Help with Transmission Line Constants in PSCAD/EMTDC:

Task 1:

1) Take **T-Line** i.e. transmission line from toolbar.
2) Double click on T-Line, in the configuration parameters dialog select Termination style as Direct connection
3) Set other parameters as per requirement
4) Click on edit to edit tower and conductor data
5) Select and delete 'frequency dependent model' block. Right click on blank area and select Bergeron model
6) Again right click on blank area to select type of tower. There are 12 tower types to choose from
7) Double click on Tower structure to edit the data as below

Component	Properties
Line constants 3 conductor flat tower	**Tower Data:** Here you can edit Height of conductors, Horizontal Spacing between conductors etc. Also you can specify no. of ground wires and transposed lines or untransposed lines. **Conductor Data:** In this either you can select conductor from a library or can specify conductor radius and DC resistance. Change the library path to this C:\Program Files\PSCAD420Eval\examples\Relay_Cases\conductor.clb **Ground Wire Data:** As in conductor data you can specify ground wire data using library or inputting radius and resistance of ground wire. Also you can specify sag for ground wires, height of ground wires and spacing between ground wires. Conductor bundling X, Y data: If bundled conductors are used, then their X, Y positions can be specified here.

8) Right click on blank space and click on additional options. After pasting it, double click on additional options and change the output file display settings in the dialog box appropriately.
9) To solve the line constants, right click on blank space and select 'solve constants'.
10) Click on 'output' at the bottom to see the results.

Laboratory Experiment 4: Power Flow

Power Flow using MATLAB and PowerWorld

Objective: To carry out power flow calculations using *MATLAB* and *PowerWorld* program.

Laboratory Tasks and Report:

1. The MATLAB files to calculate power flow in the example 3-bus power system using both Newton-Raphson and Gauss-Seidel methods are included in this folder (see video clip# 6).
    a. Execute the Newton_Raphson_task_1a.m file and obtain power and reactive power flow through all the transmission lines (both ends) and provided by the generators at buses 1 and 2 using both the methods. Draw a diagram of the three bus system and add the values of the P and Q flows and the P and Q bus injections to the diagram.
    b. Execute the Gauss_seidel_task_1a program. Does it get to the same solution? Why does it take so many more iterations?
    c. The line shunt capacitances have been added as described in Chapter 5, execute the Newton_Raphson_task_1b.m file to obtain power and reactive power flow and compare the results with part (a). Explain all differences.
2. The PowerWorld file to calculate power flow in the example 3-bus power system is **Three_Bus_PowerFlow.pwb** (see video clip# 5), which is included in this folder.
    a. Execute this file and obtain the results to confirm those from the MATLAB programs in step 1.
    b. Comment on the nature of buses 1 and 2.
    c. Add the line shunt capacitances as in part 1(c) and compare results.
    d. Limit the reactive power from generator 2 to be in a range $\pm 200\,MVA$ and see the influence on the bus 2 voltage and power flows on the lines.
    e. Raise the power delivered to bus 3 by clicking on the up arrow next to the real power displayed. How much can you deliver until voltage drops to 0.95 pu? How much can you deliver until it drops to 0.90 pu?

Newton Raphson Power Flow Program

```
% Example 5-4 Power Flow in a 3-bus Test System
clear
j = sqrt(-1);

V = zeros(3,1);
S = zeros(3,1);
Mismatch = zeros(3,1);
```

```
% ---------- Input line impedances ------------- %

Z = [0 0.0047 + 0.0474i 0.0062 + 0.0632i
     0.0047 + 0.0474i 0 0.0047 + 0.0474i
     0.0062 + 0.0632i 0.0047 + 0.0474i 0];

%---------Base Values ---------%
kVLL=345;
MVA3Ph=100;
Zbase=kVLL^2/MVA3Ph;

XL_km=0.376;                         % ohm/km at 60 Hz
RL_km= 0.037; B_km=4.5;              % B in micro-mho/km

%---------Line Susceptances--------%

B13_Micro_Mho=4.5*200;                %200 km long
B12_Micro_Mho=4.5*150;                %150 km long
B23_Micro_Mho=4.5*150;                %150 km long

%---------Line impedances------------%
Z13_ohm=(RL_km+j*XL_km)*200;          %200 km long
Z12_ohm=(RL_km+j*XL_km)*150;          %150 km long
Z23_ohm=(RL_km+j*XL_km)*150;          %150 km long

%------- line impedances in per unit--------%
Z13=Z13_ohm/Zbase;
Z12=Z12_ohm/Zbase;
Z23=Z23_ohm/Zbase;

%-------- susceptances in per unit----------%
B13=B13_Micro_Mho*Zbase*10^-6;
B12=B12_Micro_Mho*Zbase*10^-6;
B23=B23_Micro_Mho*Zbase*10^-6;

%---------- YBUS Creation-------------%
Y(1,1)=1/Z12 + 1/Z13;
Y(1,2)=-1/Z12;
Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12;
Y(2,2)=1/Z12 + 1/Z23;
Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13;
Y(3,2)=-1/Z23;
Y(3,3)=1/Z13 + 1/Z23;
Y                % Print Y=G+jB Admittance Matrix

%----------Conductance Values------------%
G(1,1)=real(Y(1,1));
G(1,2)=real(Y(1,2));
G(1,3)=real(Y(1,3));
G(2,1)=real(Y(2,1));
G(2,2)=real(Y(2,2));
G(2,3)=real(Y(2,3));
G(3,1)=real(Y(3,1));
```

```
G(3,2)=real(Y(3,2));
G(3,3)=real(Y(3,3));

%--------Susceptance Values----------%
B(1,1)=imag(Y(1,1));
B(1,2)=imag(Y(1,2));
B(1,3)=imag(Y(1,3));
B(2,1)=imag(Y(2,1));
B(2,2)=imag(Y(2,2));
B(2,3)=imag(Y(2,3));
B(3,1)=imag(Y(3,1));
B(3,2)=imag(Y(3,2));
B(3,3)=imag(Y(3,3));

%--------- Given Specifications in pu----------%
V1MAG=1.0;
ANG1=0;
V2MAG=1.05;
P2sp=2.0;
P3sp=-5.0;
Q3sp=-1.0;

% -------------Calulate ANG2, V3MAG and ANG3---------------%

% ----Solution Parameters----%
Tolerance= 0.001;
Iter_Max=10;

%----- Initialization-------%
Iter=0;
i=0;
ConvFlag=1;
ANG2=0;
ANG3=0;
V3MAG=1.0;
delANG2=0;
delANG3=0;
delMAG3=0;


%------------------- Start Iteration Process for N-R-----------------%
while( ConvFlag==1 & Iter < Iter_Max)
    Iter=Iter+1;
    i=i+1;
ANG2=ANG2+delANG2;
ANG3=ANG3+delANG3;
V3MAG=V3MAG+delMAG3;

%------- Creation of Jacobian J--------%
% J(1,1)=dP2/dAng2; Eq. 9-26; k=2, m=1,3
J(1,1)=V2MAG*(V1MAG*(-G(2,1)*sin(ANG2-ANG1)+B(2,1)*cos(ANG2-ANG1)) +
V3MAG*(-G(2,3)*sin(ANG2-ANG3)+B(2,3)*cos(ANG2-ANG3)));

% J(1,2)=dP2/dAng3; Eq. 9-27; k=2, j=3
J(1,2)=V2MAG*(V3MAG*(G(2,3)*sin(ANG2-ANG3)-B(2,3)*cos(ANG2-ANG3)));

% J(1,3)=dP2/dMAG3; Eq. 9-29; k=2, j=3
```

```
J(1,3)=V2MAG*((G(2,3)*cos(ANG2-ANG3)+B(2,3)*sin(ANG2-ANG3)));

% J(2,1)=dP3/dAng2; Eq. 9-27; k=3, j=2
J(2,1)=V3MAG*(V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2)));

% J(2,2)=dP3/dAng3; Eq. 9-26; k=3, m=1,2
J(2,2)=V3MAG*(V1MAG*(-G(3,1)*sin(ANG3-ANG1)+B(3,1)*cos(ANG3-ANG1)) +
V2MAG*(-G(3,2)*sin(ANG3-ANG2)+B(3,2)*cos(ANG3-ANG2)));

% J(2,3)=dP3/dMAG3; Eq. 9-28; k=3, m=1,2
J(2,3)=2*G(3,3)*V3MAG + V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-
ANG1)) + V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2));

% J(3,1)=dQ3/dAng2; Eq. 9-31; k=3, j=2
J(3,1)=V3MAG*(V2MAG*(G(3,2)*cos(ANG3-ANG2)-B(3,2)*sin(ANG3-ANG2)));

% J(3,2)=dQ3/dAng3; Eq. 9-30; k=3, m=1,2
J(3,2)=V3MAG*(V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-ANG1)) +
V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2)));

% J(3,3)=dQ3/dMAG3; Eq. 9-32; k=3, m=1,2
J(3,3)=- 2*B(3,3)*V3MAG + V1MAG*(G(3,1)*sin(ANG3-ANG1)-B(3,1)*cos(ANG3-
ANG1))  + V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2));

% ---------Bus Voltages--------%
V(1)=V1MAG*exp(j*ANG1);
V(2)=V2MAG*exp(j*ANG2);
V(3)=V3MAG*exp(j*ANG3);

% -------Injected currents into Buses-------%
Iinj=Y*V;

%------- P and Q Injected into Buses--------%
S(1)=V(1)*conj(Iinj(1));
S(2)=V(2)*conj(Iinj(2));
S(3)=V(3)*conj(Iinj(3));

% -----Mismatch at PQ and PV buses-------%
Mismatch(1)=P2sp-real(S(2));
Mismatch(2)=P3sp-real(S(3));
Mismatch(3)=Q3sp-imag(S(3));

% -------calculate new delta values for ANG2, ANG3, and MAG3------&
del=inv(J)*Mismatch;
delANG2=del(1);
delANG3=del(2);
delMAG3=del(3);

% -------Calculate Power Flow on the Transmission Lines-------%
P12=real(V(1)*conj((V(1)-V(2))/Z12));    Q12=imag(V(1)*conj((V(1)-
V(2))/Z12)); % at Bus 1
P13=real(V(1)*conj((V(1)-V(3))/Z13));    Q13=imag(V(1)*conj((V(1)-
V(3))/Z13)); % at Bus 1

P21=real(V(2)*conj((V(2)-V(1))/Z12));    Q21=imag(V(2)*conj((V(2)-
V(1))/Z12)); % at Bus 2
```

```
P23=real(V(2)*conj((V(2)-V(3))/Z23));    Q23=imag(V(2)*conj((V(2)-
V(3))/Z23)); % at Bus 2

P31=real(V(3)*conj((V(3)-V(1))/Z13));    Q31=imag(V(3)*conj((V(3)-
V(1))/Z13)); % at Bus 3
P32=real(V(3)*conj((V(3)-V(2))/Z23));    Q32=imag(V(3)*conj((V(3)-
V(2))/Z23)); % at Bus 3

P1=real(S(1));
Q1=imag(S(1));
P2=real(S(2));
Q2=imag(S(2));
P3=real(S(3));
Q3=imag(S(3));

%--------Display voltage and P,Q values at each iteration-------%

fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Mismatch 1   Mismatch 2
Mismatch 3 ')

fprintf('             %7.4f        %7.4f      %7.4f \n', Mismatch(1),
Mismatch(2), Mismatch(3) );


 if max(abs(Mismatch)) > Tolerance,
     ConvFlag=1;
 else
     ConvFlag=0;
 end
end

J,                                                  %  Print Final Jacobian
Matrix
ANG2DEG=ANG2*180/pi,
ANG3DEG=ANG3*180/pi,
V3MAG,




%-------Display Power Flows on the buses---------%

fprintf('\n %s','Flows on Line 1-2')
fprintf('\n  %s \n', ' P12       Q12        P21        Q21 ')
fprintf('   %7.4f   %7.4f   %7.4f   %7.4f  \n',  P12, Q12 ,P21, Q21)

fprintf('\n %s','Flows on Line 1-3')
fprintf('\n  %s \n', ' P13       Q13        P31        Q31 ')
fprintf(' %7.4f   %7.4f   %7.4f   %7.4f  \n', P13, Q13 ,P31, Q31)

fprintf('\n %s','Flows on Line 2-3')
fprintf('\n  %s \n',  ' P23       Q23        P32        Q32 ')
fprintf('  %7.4f   %7.4f   %7.4f   %7.4f  \n',  P23, Q23 ,P32, Q32)

fprintf('\n %s','Net P,Q at Buses')
```

```
fprintf('\n %s \n', '  P1           Q1           P2           Q2
P3          Q3   ')
fprintf('  %7.4f    %7.4f    %7.4f    %7.4f     %7.4f     %7.4f    \n',
P1, Q1, P2, Q2, P3, Q3)
```

Gauss-Seidel Power Flow Program

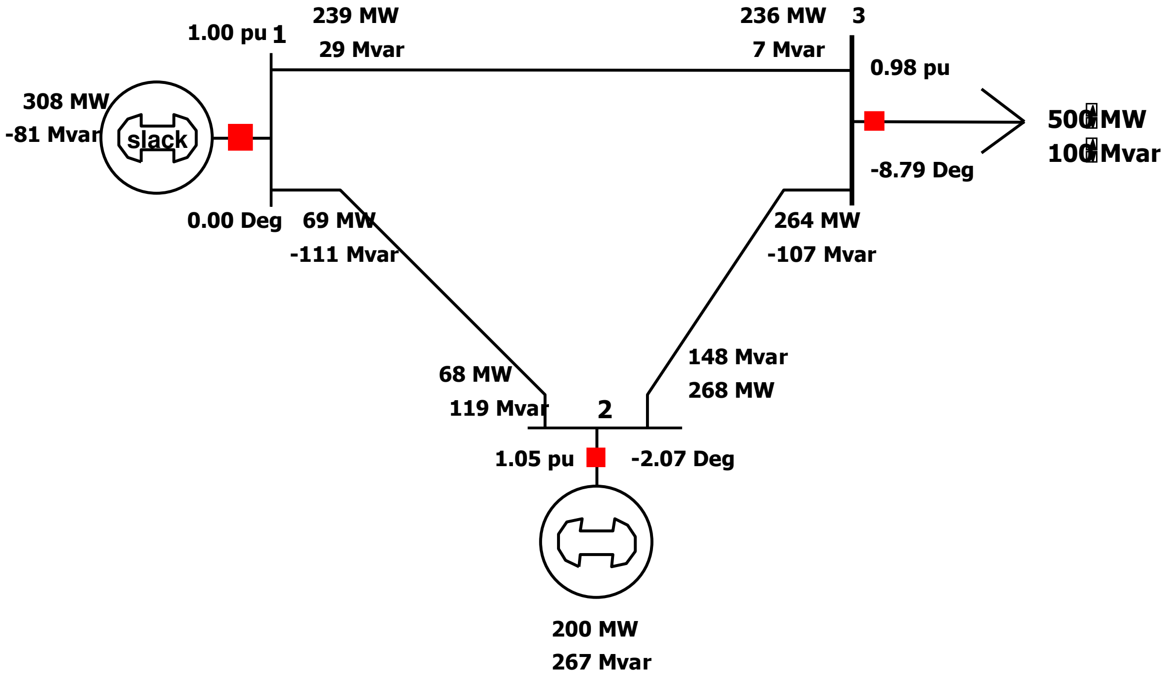
```
Iter = 0;
Maxiter = 20;
converge = 0;
tolerance  = 0.001;
j = sqrt(-1);

% ---------- Input line impedances ------------- %

Z = [0 0.0047 + 0.0474i 0.0062 + 0.0632i
     0.0047 + 0.0474i 0 0.0047 + 0.0474i
     0.0062 + 0.0632i 0.0047 + 0.0474i 0];

% ---------- Base Values ----------- %

kVLL=345;
MVA3Ph=100;
Zbase=kVLL^2/MVA3Ph;

XL_km=0.376;                                      % ohm/km at 60 Hz
RL_km= 0.037; B_km=4.5;                           % B in micro-mho/km

Z13_ohm=(RL_km+j*XL_km)*200;                      %200 km long
B13_Micro_Mho=4.5*200;                            %200 km long
Z12_ohm=(RL_km+j*XL_km)*150;                      %150 km long
B12_Micro_Mho=4.5*150;                            %150 km long
Z23_ohm=(RL_km+j*XL_km)*150;                      %150 km long
B23_Micro_Mho=4.5*150;                            %150 km long

%--------- line impedances in per unit ----------%

Z13=Z13_ohm/Zbase;
Z12=Z12_ohm/Zbase;
Z23=Z23_ohm/Zbase;

%--------- susceptances in per unit ----------%

B13=B13_Micro_Mho*Zbase*10^-6;
B12=B12_Micro_Mho*Zbase*10^-6;
B23=B23_Micro_Mho*Zbase*10^-6;

%-------Finding the admittance matrix---------%

Y(1,1)=1/Z12 + 1/Z13;
Y(1,2)=-1/Z12;
Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12;
Y(2,2)=1/Z12 + 1/Z23;
```

```
Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13;
Y(3,2)=-1/Z23;
Y(3,3)=1/Z13 + 1/Z23;
G = real(Y);
B = imag(Y);

% ------- Initialize line variables --------- %

I = zeros(1,length(Z));
P = zeros(1,length(Z));
Q = zeros(1,length(Z));
S = zeros(1,length(Z));

% -------- Input bus parameters --------- %

v = ones(1,length(Z));
%angle = zeros(length(Z),1);
Pload = zeros(1,length(Z));
Qload = zeros(1,length(Z));
Psched = zeros(1,length(Z));
Qsched = zeros(1,length(Z));

Bustype(1)  = 'S';                          % PQ bus
Bustype(2)  = 'G';                          % PV bus
Bustype(3)  = 'L';                          % PQ bus

v(2) = 1.05;
Pload(2)  = -2;
Pload(3)  = 5;
Qload(3)  = 1;
% ------ Set up power schedule --------- %

totload = 0.0;
for i = 1:length(Z)
   Psched(i)= -Pload(i);
   Qsched(i)= -Qload(i);
   totload = totload + Pload(i);
end

% --- Calculate P and Q at each bus ----- %

for i = 1:length(Z)
   I(i)  = 0.0 + sqrt(-1)*0.0;
   for j = 1:length(Z)
      I(i)  = I(i)  + Y(i,j)*v(j);
   end
   S(i)  = v(i)  * conj(I(i));
   P(i)  = real(S(i));
   Q(i)  = imag(S(i));
end

% ------- Display bus voltages ---------- %

fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Vreal   Vimag   Vmag
Vangle     P      Q ')
for i = 1:length(Z)
```

```
   vitermag = abs(v(i));
   viterang = (180./pi)*atan2(imag(v(i)),real(v(i)));
   fprintf(' %s %2d %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f \n',' Bus ',i,
real(v(i)), imag(v(i)), vitermag, viterang, P(i), Q(i) )
end

while converge == 0
    Iter = Iter + 1;
    MAXDP = 0.0;
    MAXDPbus = 0;
    MAXDQ = 0.0;
    MAXDQbus = 0;

    for i = 1 : length(Z)
%---------calculate net P and Q at bus i--------%
        I(i) = 0.0 + sqrt(-1)*0.0;
        for j = 1:length(Z)
            I(i) = I(i) + Y(i,j)*v(j);
        end
        S(i) = v(i) * conj(I(i));
        P(i) = real(S(i));
        Q(i) = imag(S(i));

        if Bustype(i) == 'G'
            Qsched(i) = Q(i);
        end
        deltap(i) = abs(P(i) - Psched(i));
        deltaq(i) = 0.0;

        if Bustype(i) == 'L'
            deltaq(i) = abs(Q(i) - Qsched(i));
        end

        if Bustype(i) == 'S'
            deltap(i) = 0.0;
            deltaq(i) = 0.0;
        end

        if Bustype(i) ~= 'S'
            if deltap(i) > MAXDP
                MAXDP = deltap(i);
                MAXDPbus = i;
            end

            if deltaq(i) > MAXDQ
                MAXDQ = deltaq(i);
                MAXDQbus = i;
            end

%--------- Y * V for row i of Y matrix without Yii term---------%

            sum = 0.0;
            for j = 1:length(Z)
                if j ~= i
                    sum = sum + Y(i,j)*v(j);
                end
            end
```

```
            vnew = (1.0/Y(i,i))*( ((Psched(i) - sqrt(-
1)*Qsched(i))/(conj(v(i)))) - sum);
            v(i) = vnew;
        end
    end

%------------ Print and save result from last iteration-----------%

%-----calculate net P and Q at bus i-----%

    for i = 1:length(Z)
        I(i) = 0.0 + sqrt(-1)*0.0;
        for j = 1:length(Z)
            I(i) = I(i) + Y(i,j)*v(j);
        end
        S(i) = v(i) * conj(I(i));
        P(i) = real(S(i));
        Q(i) = imag(S(i));

        if Bustype(i) == 'G'
            Qsched(i) = Q(i);
        end
    end

    fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Vreal   Vimag   Vmag
Vangle     P      Q ')
    for i = 1:length(Z)
        vitermag = abs(v(i));
        viterang = (180./pi)*atan2(imag(v(i)),real(v(i)));
        fprintf(' %s %2d %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f \n',' Bus
',i, real(v(i)), imag(v(i)), vitermag, viterang, P(i), Q(i) )
    end

% -------- check for convergence ---------- %

    if MAXDP < tolerance
        if MAXDQ < tolerance
            converge = 1;
        end
    end

    if Iter > Maxiter
        converge = 1;
    end
end
```

3Bus_PowerFlow.pwb



Problem 5-8
Confirm the MATLAB Results of
Example 5-4.

Laboratory Experiment 5: Transformers in Power Flow

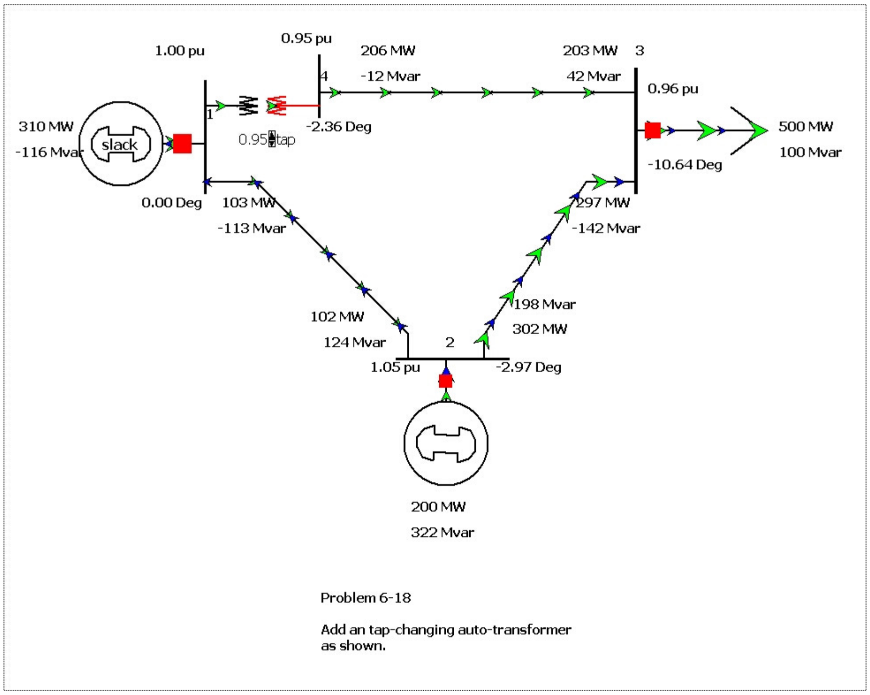
Including Transformers in Power Flow using PowerWorld and Confirmation by MATLAB

Objectives: To look at the influence of including a tap-changer and a phase-shifter on power flow and bus voltages.

Laboratory Tasks and Report:

1. Including a Tap Changer (**PowerFlow_AutoTransformer.pwb**; see video clip# 7)
    a. An Autotransformer is added between buses 1 and 4 (newly created) as shown in the *PowerWorld* file **PowerFlow_AutoTransformer.pwb**, which is located in this Folder. Double click on this file or open it through *PowerWorld*. The tap-ratio between buses 1 and 4 is such that $n_1 / n_4 = 0.95$. Compare this case with that in Example 5-4 for the various bus voltages and the power flow on various lines due to this tap ratio.
    b. An autotransformer is used to control voltage on one bus of the transformer. You should click on the arrows to change this tap until the voltage at bus 4 is raised to 1.05 pu. What does raising this tap do to the reactive flows? How does raising the tap affect the reactive output of each generator?
    c. Represent this auto-transformer by means of a pi-circuit of Fig. 6-17b in a MATLAB program, using the results of part a (that is with tap at 0.95), to confirm the results of part a.

PowerFlow_AutoTransformer.pwb

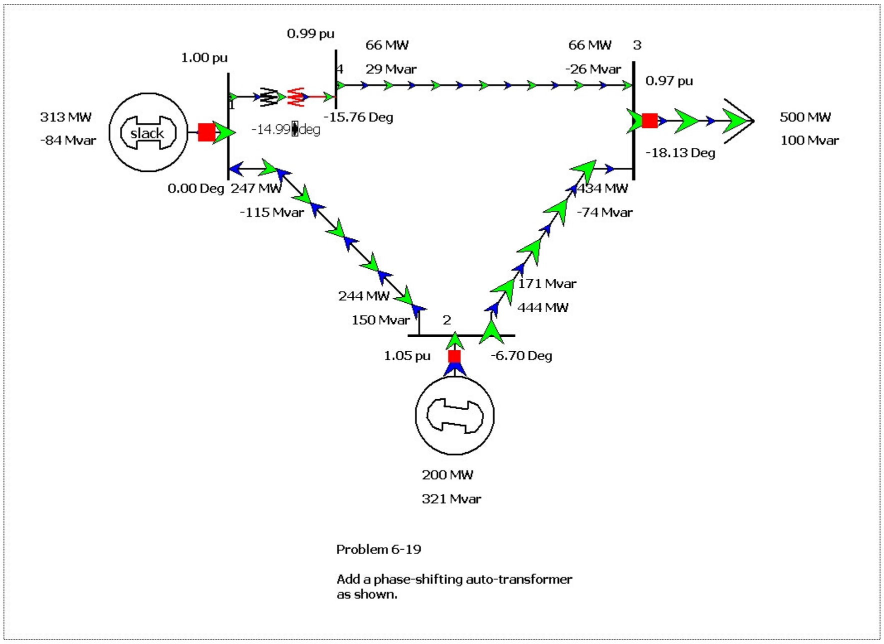


Problem 6-18

Add an tap-changing auto-transformer as shown.

2. Including a Phase-Shifter **(PowerFlow_PhaseShift.pwb**; see video clip# 7**)**
   a. A phase-shift transformer is added between buses 1 and 4 (newly created) as shown in the *PowerWorld* file **PowerFlow_PhaseShift.pwb**, which is located in this Folder. Double click on this file or open it through *PowerWorld*. The phase-shift between buses 1 and 4 is such that $V_1\angle 0^\circ$ results in $V_4\angle -15.76^\circ$ when the tap is at -14.99°. Compare this case with that in Example 5-4 for the various bus voltages and the power flow on various lines due to this phase shift.
   b. The phase shift transformer is used to control power flowing through the transformer. Click on the tap adjustment until the MW flowing from bus 4 to bus 3 is exactly 100 MW at the bus 4 end of the 4-3 line. What is the effect on the MW flows on the other two lines, what if any, is the effect of this tap change on the reactive flows on the other lines.
   c. Represent this phase-shift transformer by means of Eq. 6-32 in a MATLAB program, using the results of part a (that is with the tap at the initial value of -14.99°), to confirm the results of part a.

PowerFlow_PhaseShift.pwb



Add a phase-shifting auto-transformer as shown.

Laboratory Experiment 6: Including an HVDC Transmission Line for Power Flow

Including an HVDC Transmission Line for Power Flow Calculations in PowerWorld and Modeling of Thyristor Converters in PSCAD/EMTDC

Objectives:

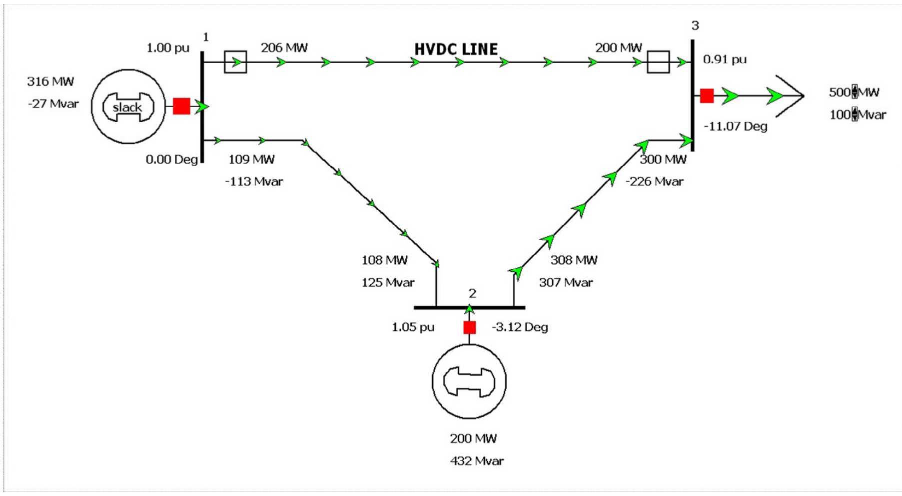
1. To include an HVDC transmission line and see its effect on power transfer on other transmission line.
2. To understand the operating principle of 12-pulse thyristor converters used in HVDC transmission systems.

Laboratory Tasks and Report:

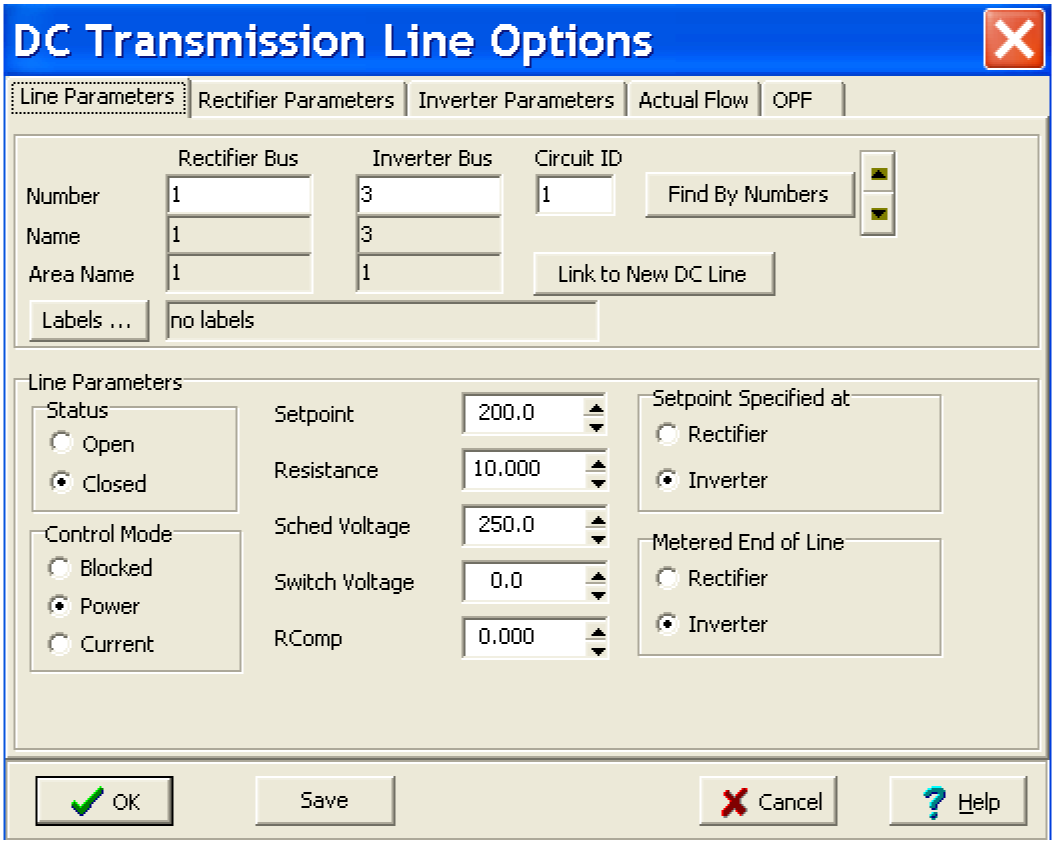
1. The transmission line between buses 1 and 3 is an HVDC line, as described in the *PowerWorld* file **PowerFlow_HVDCline.pwb** (see video clip# 8), which is located in this Folder. Double click on this file or open it through *PowerWorld*. Look at various characteristics of this HVDC system by examining its parameters; see dialog boxes below. Compare this case with that in Example 5-4 for the various bus voltages and the power flow on various lines due to this HVDC line. Change the set point from 200 MW to 300 MW and then to 400 MW. Explaing what you see.
2. Obtain the waveforms of individual Rectifier DC voltage and combined 12-pulse DC voltage output , for different firing angles , in a 12-pulse thyristor converter operating in a rectifier-mode described by the PSCAD/EMTDC file in this folder called **HVDC_Rectifier.psc** (see video clip# 9). Source: Courtesy of Prof. Ani Golé of the University of Manitoba.
3. Obtain the waveforms of individual inverter DC voltage and combined 12-pulse DC voltage input ,for different firing angles in a 12-pulse thyristor converter operating in the inverter-mode described by the PSCAD/EMTDC file in this folder called **HVDC_Inverter.psc** (see video clip# 9). Source: Courtesy of Prof. Ani Golé of the University of Manitoba.
4. By using the formula (7-12) and (7-13), for different firing angles, calculate the DC voltage and match with the value obtained from the waveform. For Rectifier: w*Ls = 13.6791 ohm, VLL=213 kV, Id= Obtain from simulation For Inverter: w*Ls=13.1843 ohm, VLL=207 kV, Id= Obtain from simulation.
5. Obtain the waveforms of the input and output currents for both the transformers in rectifier and inverter. Observe the phase shift between the primary and secondary of Wye-Delta transformer. Explain what you see.

6. Obtain harmonic components of secondary line current of Wye-Delta Transformer and harmonic components of the DC line voltage in the rectifier and inverter. What is the significance of the harmonics that appear?

PowerFlow_HVDCline.pwb



Problem 7-22 HVDC Line between Buses 1 and 3 is set to deliver 200 MW to Bus 3. The voltage at the inverter terminal is 250 kV



	Rectifier Bus	Inverter Bus	Circuit ID
Number	1	3	1
Name	1	3	
Area Name	1	1	

Labels: no labels

Line Parameters

Status: ○ Open ● Closed

Control Mode: ○ Blocked ● Power ○ Current

Setpoint 200.0

Resistance 10.000

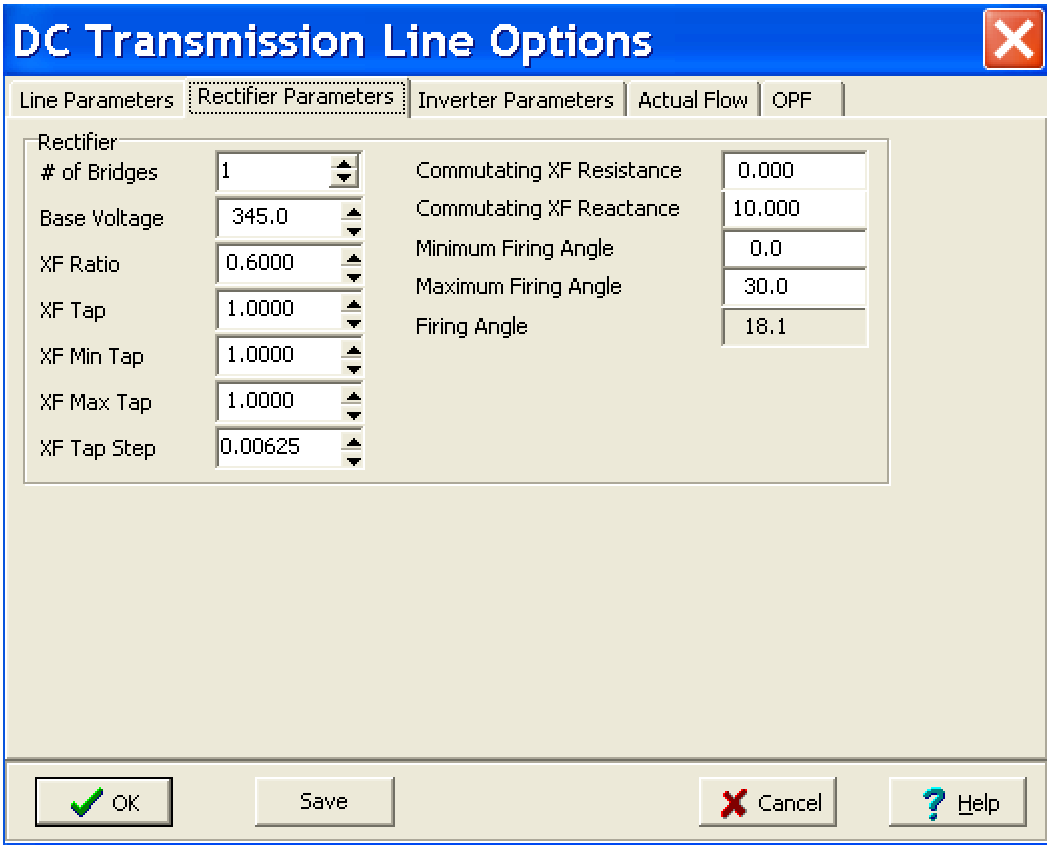
Sched Voltage 250.0

Switch Voltage 0.0

RComp 0.000

Setpoint Specified at: ○ Rectifier ● Inverter

Metered End of Line: ○ Rectifier ● Inverter



Rectifier

# of Bridges 1

Base Voltage 345.0

XF Ratio 0.6000

XF Tap 1.0000

XF Min Tap 1.0000

XF Max Tap 1.0000

XF Tap Step 0.00625

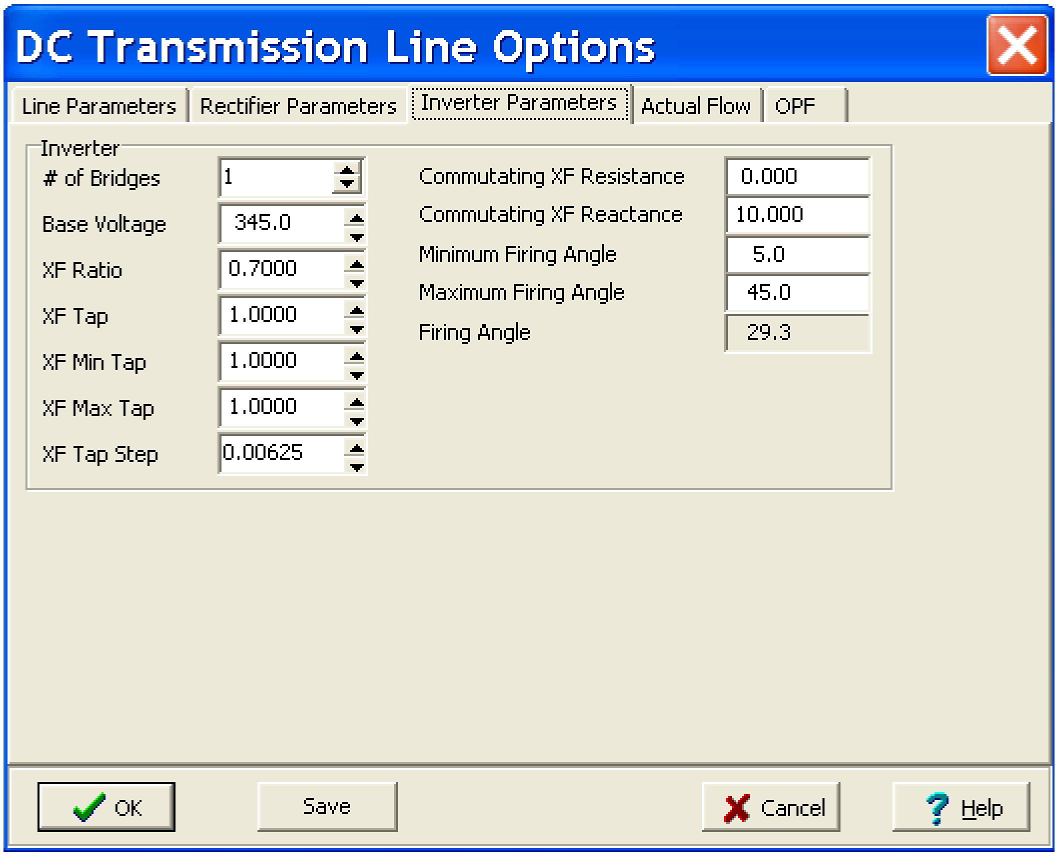
Commutating XF Resistance 0.000

Commutating XF Reactance 10.000

Minimum Firing Angle 0.0

Maximum Firing Angle 30.0

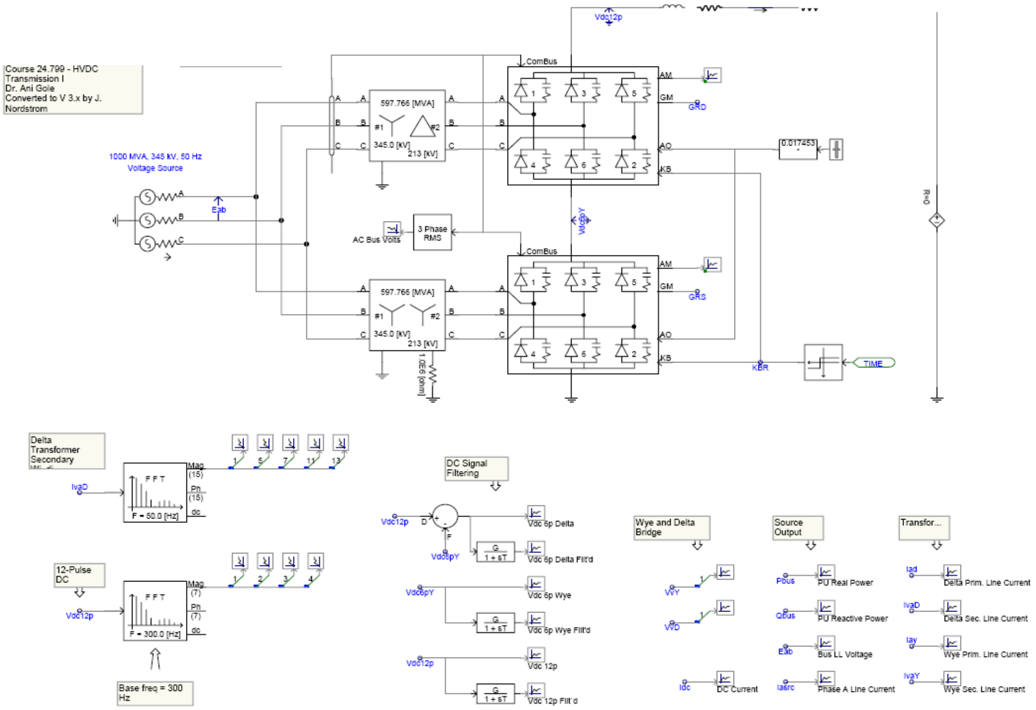
Firing Angle 18.1



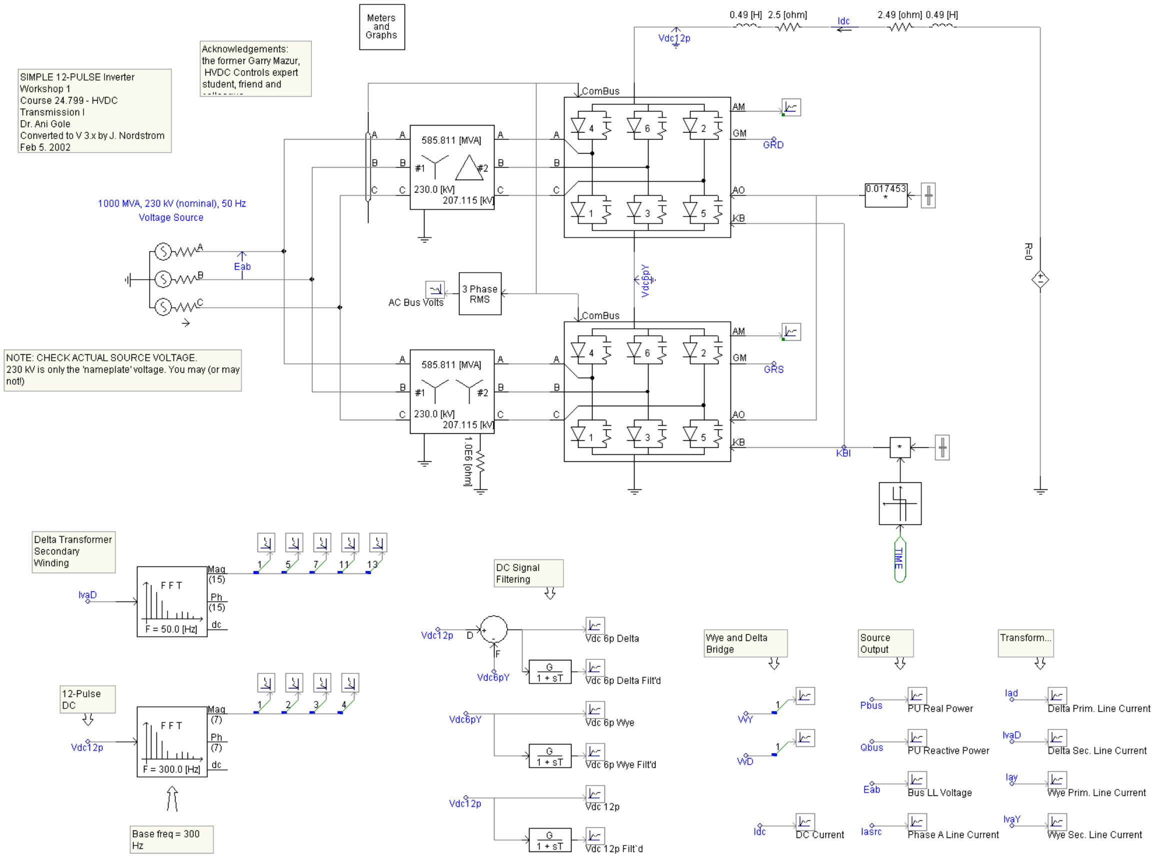
Inverter

# of Bridges	1
Base Voltage	345.0
XF Ratio	0.7000
XF Tap	1.0000
XF Min Tap	1.0000
XF Max Tap	1.0000
XF Tap Step	0.00625
Commutating XF Resistance	0.000
Commutating XF Reactance	10.000
Minimum Firing Angle	5.0
Maximum Firing Angle	45.0
Firing Angle	29.3

HVDC_Rectifier.psc



HVDC_Inverter.psc



SIMPLE 12-PULSE Inverter
Workshop 1
Course 24.799 - HVDC
Transmission I
Dr. Ani Gole
Converted to V 3.x by J. Nordstrom
Feb 5. 2002

Acknowledgements:
the former Garry Mazur,
HVDC Controls expert
student, friend and

NOTE: CHECK ACTUAL SOURCE VOLTAGE.
230 kV is only the 'nameplate' voltage. You may (or may not!)

Laboratory Experiment 7: Power Quality

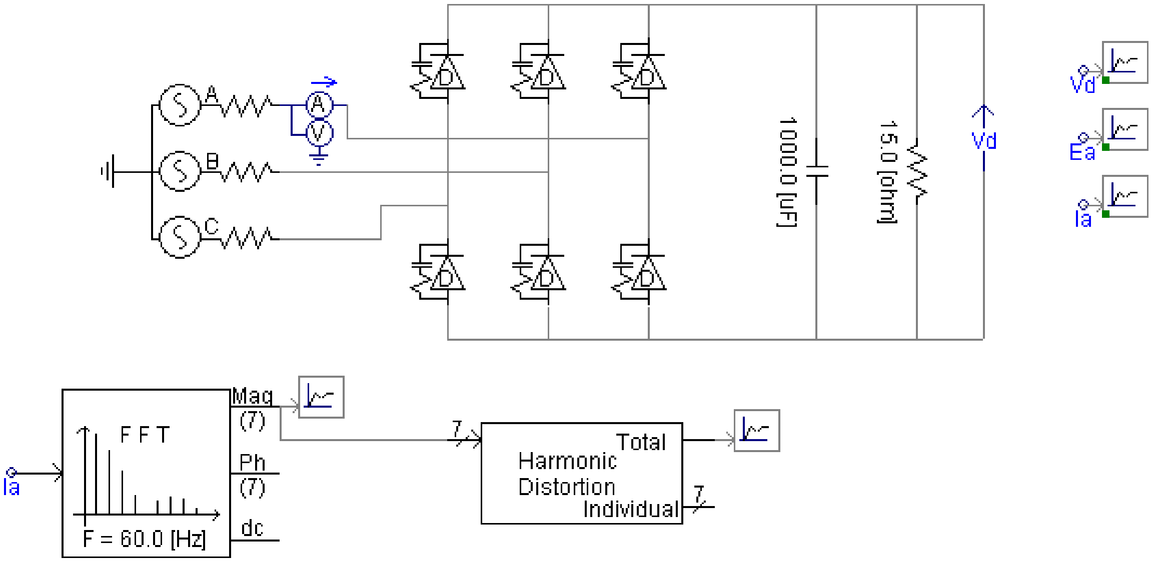
Power Quality

Objectives: To obtain the current harmonics drawn by power electronics interface.

Laboratory Tasks and Report:

1. Calculate the displacement power factor, power factor and the total harmonic distortion associated with the power-electronics interface described in the *PSCAD/EMTDC* file **PowerQuality.psc**. See video clip# 10.

Help with PowerQuality.psc
Build the circuit as shown below



FFT and harmonic distortion blocks are taken from CSMF library.
Voltage and currents are plotted as described in first few experiments.

The class textbook goes into how to calculate THD in Chapter 8. The important thing to remember is that THD is the ratio of the sum of all powers at frequencies above the fundamental to the power supplied at the fundamental frequency:

$$THD = \frac{\sum \text{harmonic frequency powers}}{\text{fundamental frequency power}} = \frac{P_2 + P_3 + \cdots + P_n}{P_1}$$

The displacement power factor, DPF is the cosine of the angle between the fundamental voltage and the current at the fundamental frequency. When the THD approaches zero the DPF approaches the usual power factor which is the cosine of the angle between the voltage and the net current.

Explanation of THD

In most cases, the transfer function of a system is linear and time-invariant. When a signal passes through a non-linear device, additional content is added at the harmonics of the original frequencies. THD is a measurement of the extent of that distortion.

The measurement is most commonly the ratio of the sum of the powers of all harmonic frequencies *above* the fundamental frequency to the power of the fundamental:

$$\mathrm{THD} = \frac{\sum \text{harmonic powers}}{\text{fundamental frequency power}} = \frac{P_2 + P_3 + P_4 + \cdots + P_n}{P_1}$$

Other calculations for amplitudes, voltages, currents, and so forth are equivalent. For a voltage signal, for instance, the ratio of the squares of the RMS voltages is equivalent to the power ratio:

$$\mathrm{THD} = \frac{V_2^2 + V_3^2 + V_4^2 + \cdots + V_n^2}{V_1^2}$$

In this calculation, V_n means the RMS voltage of harmonic n, where n=1 is the fundamental harmonic. One can also calculate THD using all harmonics (n=□):

$$\mathrm{THD} = \frac{V_{RMS}^2 - V_1^2}{V_1^2}$$

Other definitions may be used. Many authors define THD as an amplitude ratio rather than a power ratio. This results in a definition of THD which is the square root of that given above. For example in terms of voltages the definition would be:

$$\mathrm{THD} = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \cdots + V_n^2}}{V_1}$$

This latter definition is commonly used in audio distortion (percentage THD) specifications. It is unfortunate that these two conflicting definitions of THD (one as a power ratio and the other as an amplitude ratio) are both in common usage. Fortunately, if the THD is expressed in dB, then both definitions are equivalent. This is not the case if the THD is expressed as a percentage. The power THD can be higher than 100% and is known as IEEE, but for audio measurements 100% is preferred as maximum, thus the IEC version is used (Rohde & Schwartz, Bruel and Kjær use it).

A measurement must also specify how it was measured. Measurements for calculating the THD are made at the output of a device under specified conditions. The THD is usually expressed in percent as distortion factor or in dB as distortion attenuation. A meaningful measurement must include the number of harmonics included.

Laboratory Experiment 8: Synchronous Generators

Synchronous Generators

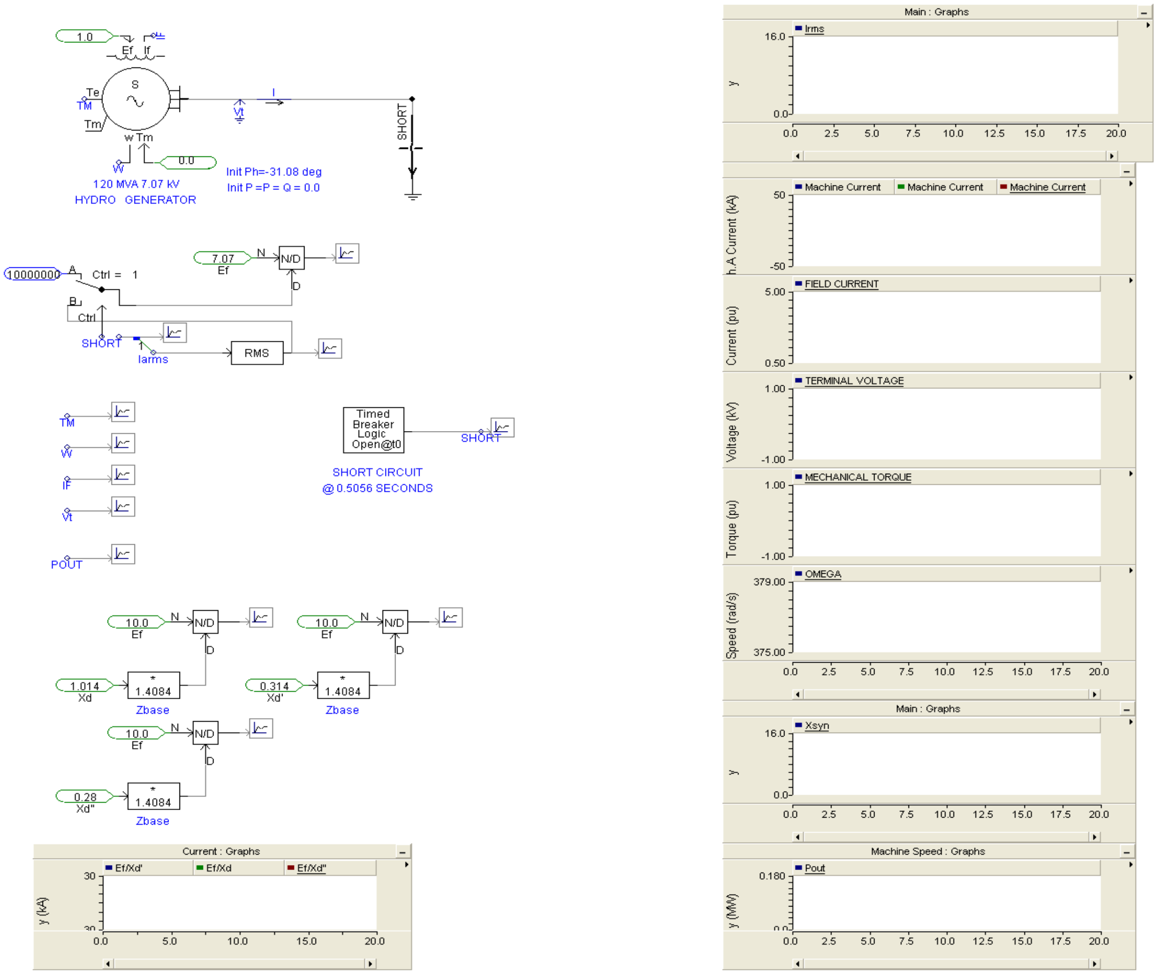
Objectives: To obtain the effect of sudden short-circuit on a synchronous generator output.

Laboratory Tasks and Report:

1. Model a short-circuit on a synchronous generator as described by the *PSCAD/EMTDC* file **SynchGen.psc**. Obtain various waveforms and comment on them. See video clip# 11.
2. Plot Ef/Xd” , Ef/Xd’, Ef/Xd and Phase A line current together, to verify the peak current transition at the three transient modes.

SynchGen.psc

SHORT CIRCUIT TEST



Laboratory Experiment 9: Voltage Regulation

Voltage Regulation

Objectives:

1) To study the effect of real and reactive powers on bus voltages.
2) Understanding the operation of a Thyristor Controlled Reactor (TCR).

Laboratory Tasks and Report:

1. In the PowerWorld example **VoltageRegulation.pwb**, vary the reactive power consumed at Bus 3 in a range from 300 MAVR to -300 MVAR and plot its effect on voltage magnitudes at Buses 3 and 2. Both line MW (green arrows) and MVAR (blue arrows) are shown. Note the direction of line MVAR flow on lines 1-3 and 2-3 as the load MVAR is changed.

   The generator at bus 2 has an upper MVAR limit of +250 MVAR and a lower MVAR limit of -200 MVAR. When does it hit the upper MVAR limit, what happens after the limit is hit (note the bus voltage and the MVAR output of the generator at bus 2).

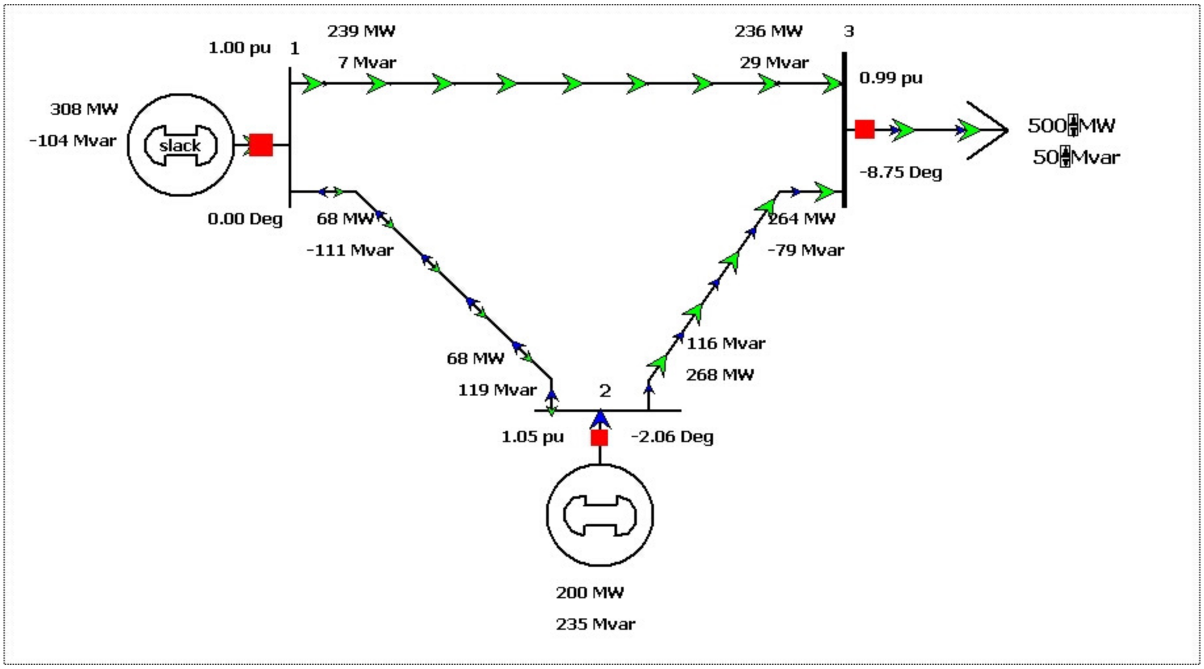
2. The TCR is modeled in the *PSCAD/EMTDC* file **TCR.psc** (see video clip# 12).

   A TCR is a "variable reactor" which can be used in a power system to vary the amount of inductive reactance connected to a bus.

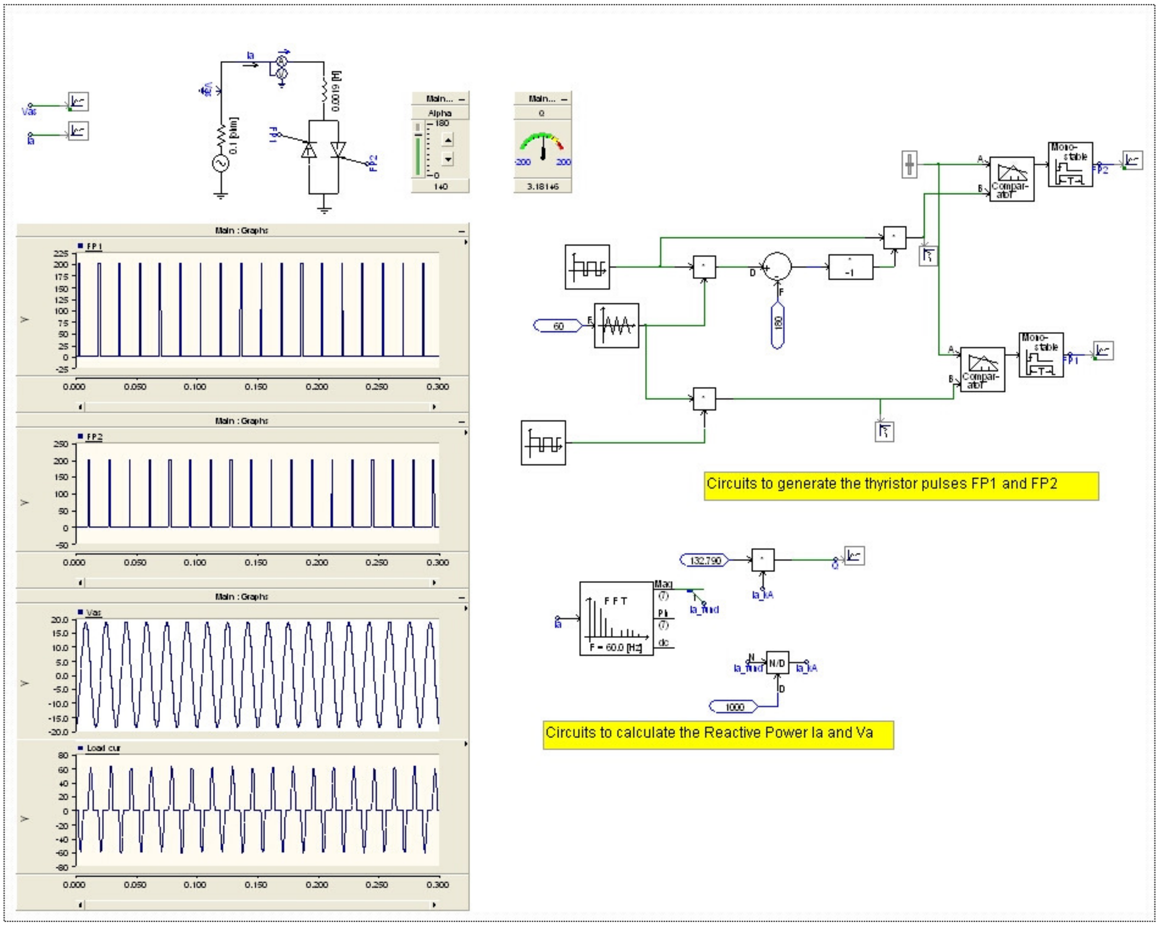
   The TCR is to be connected to a bus in a power system where it can absorb reactive power from the bus. In the model in **TCR.psc** the bus is represented by a voltage source with a resistance of 0.1 ohm. The amount of reactive power drawn by the variable reactor is controlled by the angle "Alpha" which can be adjusted using the mouse. The MVAR is seen on the display next to the Alpha adjustment box.

   Plot the reactive output versus the angle Alpha from Alpha = 90 deg to Alpha = 180 deg. Show the pulse plots for Alpha = 90 deg and for Alpha = 180 deg and explain how changes in the current waveform are related to changes in reactive power.

VoltageRegulation.pwb



TCR.psc



Circuits to generate the thyristor pulses FP1 and FP2

Circuits to calculate the Reactive Power Ia and Va

Laboratory Experiment 10: Transient Stability

Transient Stability using MATLAB

Objectives: To simulate transient stability in a 3-bus example power system.

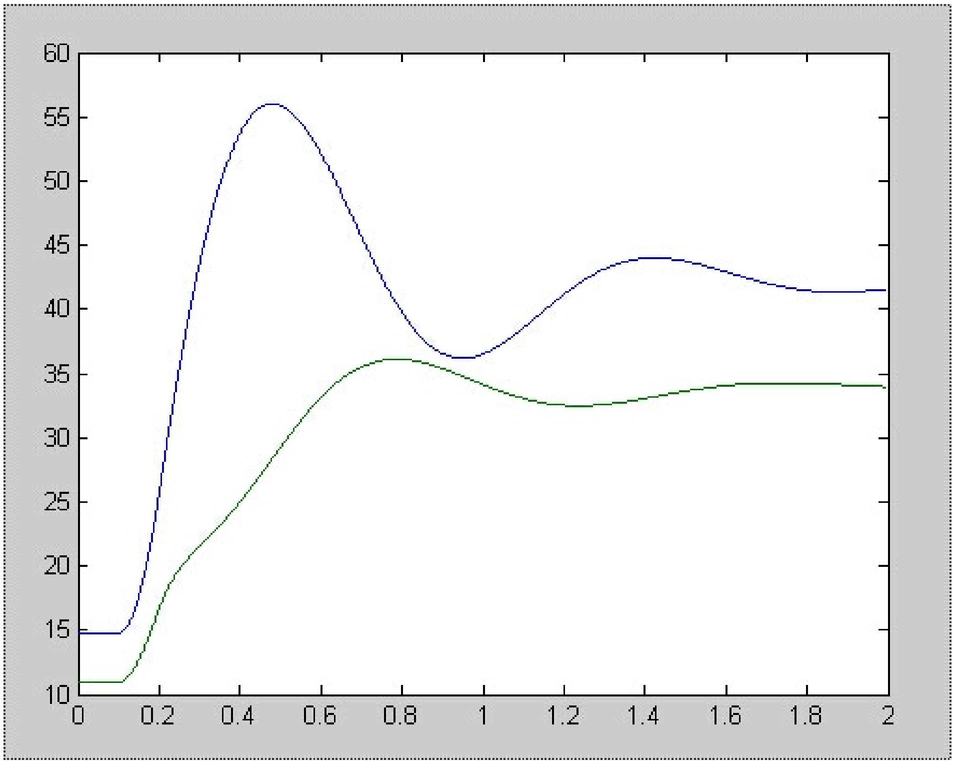
Laboratory Tasks and Report:

The MATLAB file to calculate Transient Stability in the example 3-bus power system is **TransientStability.m**, which is included in this folder. This file has a power flow to initialize the simulation and then three separate simulations for prefault, during the fault, and post fault dynamics. The program initially simulates a three phase fault on line 1-2 at 1/3 of the distance from bus 1 to bus 2. See video clip# 14.

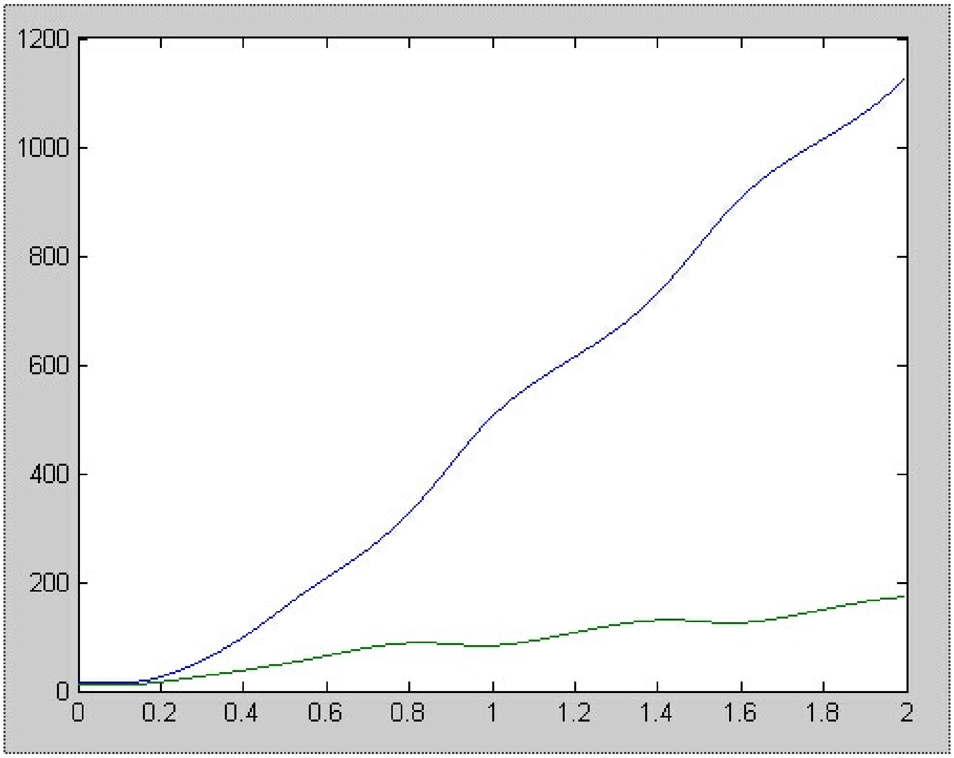
a. Annotate this file based on the material and equations of Chapter 11.
b. Execute this file get the plots of rotor angles of generators 1 and 2 (see figures below)
    i. Note that the clearing time is shown in seconds at line 125 of the program.
    ii. When you get the program the clearing time is set to 0.2 second which results in a stable system.
    iii. Start increasing the clearing time until the system goes unstable. Try to determine the "critical clearing time" which is the amount of time that can be allowed before the system goes unstable. Save the plots and note the maximum swing angle for each clearing time you try.
c. Next you are going to convert generator 2 to an "infinite" generator by setting its H constant to a very large number. You do this by putting a % sign in front of the code where H2 is calculated and then removing the % sign in front of the next line thereby making H2 = 50000.0. Now start the clearing time at 0.2 seconds and find the critical clearing time with gen 2 as an infinite generator.
d. Assume that a three-phase fault occurs on line between buses 2 and 3, one-third away from bus 2. Modify the program to represent the fault on line 2-3 instead of line 1-2. Run with generator 2 as a normal generator (not an infinite generator). Start the clearing time at 0.2 second and then find the critical clearing time for this case.

Note that the program you have for this lab has included some damping into the differential equations of the generators. The textbook does not show this damping but damping is always present in real world systems.

Here is a plot of the two generator rotor angles when the fault is cleared quickly:



Here is a plot when the clearing time is too long:



TransientStability.m

```
% Example 11-3 Swing Curves
clear all
j = sqrt(-1);

XL_km=0.367; % ohm/km at 60 Hz
RL_km= 0.1*XL_km; % Resistance in ohm/km

KV_LL= 345; MVA_Base=100; % common 3-phase base
Z_Base=KV_LL^2/MVA_Base; % common base

% YBUS Creation
Z13_ohm=(RL_km+j*XL_km)*200; B13_Micro_Mho=4.5*200; % Line 1-3 is 200 km long
Z12_ohm=(RL_km+j*XL_km)*150; B12_Micro_Mho=4.5*150; % Line 1-2 is 150 km long
Z23_ohm=(RL_km+j*XL_km)*150; B23_Micro_Mho=4.5*150; % Line 2-3 is 150 km long
Z13=Z13_ohm/Z_Base; Z12=Z12_ohm/Z_Base; Z23=Z23_ohm/Z_Base; % line impedances in per unit

Y(1,1)=1/Z12 + 1/Z13; Y(1,2)=-1/Z12; Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12; Y(2,2)=1/Z12 + 1/Z23; Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13; Y(3,2)=-1/Z23; Y(3,3)=1/Z13 + 1/Z23;
G(1,1)=real(Y(1,1)); B(1,1)=imag(Y(1,1)); G(1,2)=real(Y(1,2)); B(1,2)=imag(Y(1,2));
G(1,3)=real(Y(1,3)); B(1,3)=imag(Y(1,3));
G(2,1)=real(Y(2,1)); B(2,1)=imag(Y(2,1)); G(2,2)=real(Y(2,2)); B(2,2)=imag(Y(2,2));
G(2,3)=real(Y(2,3)); B(2,3)=imag(Y(2,3));
G(3,1)=real(Y(3,1)); B(3,1)=imag(Y(3,1)); G(3,2)=real(Y(3,2)); B(3,2)=imag(Y(3,2));
G(3,3)=real(Y(3,3)); B(3,3)=imag(Y(3,3));

% Given Specifications
V1MAG=1.0; ANG1=0; V2MAG=1.03; P2sp=5.0; P3sp=-9.0; Q3sp=-4.0;

% Solution Parameters
Tolerance= 0.001; Iter_Max=10;

% Initialization
Iter=0; ConvFlag=1;
ANG2=0; ANG3=0; V3MAG=1.0;
delANG2=0; delANG3=0; delMAG3=0;

% Start Iteration Process for N-R
while( ConvFlag==1 & Iter < Iter_Max)
    Iter=Iter+1;
ANG2=ANG2+delANG2;
ANG3=ANG3+delANG3;
V3MAG=V3MAG+delMAG3;

% Creation of Jacobian J
% J(1,1)=dP2/dAng2; k=2, m=1,3
J(1,1)=V2MAG*(V1MAG*(-G(2,1)*sin(ANG2-ANG1)+B(2,1)*cos(ANG2-ANG1)) + V3MAG*(-
G(2,3)*sin(ANG2-ANG3)+B(2,3)*cos(ANG2-ANG3)));
% J(1,2)=dP2/dAng3; k=2, j=3
J(1,2)=V2MAG*(V3MAG*(G(2,3)*sin(ANG2-ANG3)-B(2,3)*cos(ANG2-ANG3)));
% J(1,3)=dP2/dMAG3; k=2, j=3
J(1,3)=V2MAG*((G(2,3)*cos(ANG2-ANG3)+B(2,3)*sin(ANG2-ANG3)));
% J(2,1)=dP3/dAng2; k=3, j=2
J(2,1)=V3MAG*(V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2)));
% J(2,2)=dP3/dAng3; k=3, m=1,2
J(2,2)=V3MAG*(V1MAG*(-G(3,1)*sin(ANG3-ANG1)+B(3,1)*cos(ANG3-ANG1)) + V2MAG*(-
G(3,2)*sin(ANG3-ANG2)+B(3,2)*cos(ANG3-ANG2)));
% J(2,3)=dP3/dMAG3; k=3, m=1,2
J(2,3)=2*G(3,3)*V3MAG + V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-ANG1)) +
V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2));
% J(3,1)=dQ3/dAng2; k=3, j=2
J(3,1)=V3MAG*(V2MAG*(G(3,2)*cos(ANG3-ANG2)-B(3,2)*sin(ANG3-ANG2)));
% J(3,2)=dQ3/dAng3; k=3, m=1,2
J(3,2)=V3MAG*(V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-ANG1)) +
V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2)));
% J(3,3)=dQ3/dMAG3; k=3, m=1,2
J(3,3)=- 2*B(3,3)*V3MAG + V1MAG*(G(3,1)*sin(ANG3-ANG1)-B(3,1)*cos(ANG3-ANG1))  +
V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2));
```

```
% Voltages
V(1,1)=V1MAG*exp(j*ANG1);
V(2,1)=V2MAG*exp(j*ANG2);
V(3,1)=V3MAG*exp(j*ANG3);

% Injected currents
Iinj=Y*V;

% P and Q Injected
S(1,1)=V(1,1)*conj(Iinj(1)); S(2,1)=V(2,1)*conj(Iinj(2)); S(3,1)=V(3,1)*conj(Iinj(3));

% Mismatch at PQ and PV buses
Mismatch(1,1)=P2sp-real(S(2,1)); Mismatch(2,1)=P3sp-real(S(3,1)); Mismatch(3,1)=Q3sp-
imag(S(3,1));

% calculate new delta values for ANG2, ANG3, and MAG3
del=inv(J)*Mismatch;
delANG2=del(1); delANG3=del(2); delMAG3=del(3);
 if max(abs(Mismatch)) > Tolerance,
     ConvFlag=1;
 else
     ConvFlag=0;
 end
end
Pm1=real(S(1,1))
Pm2=real(S(2,1))
P3=-real(S(3,1))
Q3=-imag(S(3,1))
ZLoad=V(3,1)/(-Iinj(3,1));

Xtr1_PU=0.12*MVA_Base/500;
% Transformer base is 500 MVA
S_Gen1=500;

XdP1_PU=0.23*(MVA_Base/S_Gen1);
% Gen XdP is 0.23pu on the base of 500MVA and 22kVLL
S_Gen2=600;

Xtr2_PU=0.12*MVA_Base/600;
% Transformer base is 600 MVA

XdP2_PU=0.23*(MVA_Base/S_Gen2);
% Gen XdP is 0.23pu on the base of 600MVA and 22kVLL

H_Gen=3.5; wsyn=377;

H1=H_Gen*(S_Gen1/MVA_Base);
damp1 = 0.2;
X1=XdP1_PU+Xtr1_PU; % internal plus transformer impedances for gen 1

H2=H_Gen*(S_Gen2/MVA_Base);
%H2 = 50000.;
damp2 = 2.0;
X2=XdP2_PU+Xtr2_PU; % internal plus transformer impedances for gen 2

% Pre-Fault (Pre) steady state
EP1=V(1,1)+j*X1*Iinj(1,1);
EP1MAG=abs(EP1);
EP2=V(2,1)+j*X2*Iinj(2,1);
EP2MAG=abs(EP2);


DT=0.0001;               % Time step is 1*10^-4 seconds
Time_fault_start = 0.1; % start the fault at 0.1 seconds
Time_fault_clear = 0.2; % user can change the clear time, it must be greater than 0.1 sec
and less than 2.0 sec
Time_max = 2.0;          % maximum time is 2 seconds

% set up initial conditions
Pe1=Pm1; Pe2=Pm2;
```

```
Del1(1)=angle(EP1); Del2(1)=angle(EP2); w1(1)=wsyn; w2(1)=wsyn; time1(1)=0; DelREF=0;
DelDIFF1_DEG(1)=(Del1(1)-DelREF)*180/pi;
DelDIFF2_DEG(1)=(Del2(1)-DelREF)*180/pi;

% Pre-Fault Transient
% build Y matrix for all lines in, no fault
Y(1,1)=1/(j*X1)+1/Z12 + 1/Z13; Y(1,2)=-1/Z12; Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12; Y(2,2)=1/(j*X2)+1/Z12 + 1/Z23; Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13; Y(3,2)=-1/Z23; Y(3,3)=1/ZLoad+1/Z13 + 1/Z23;
imax = round( Time_fault_start/DT );  % imax is the number of steps to use in the pre-
fault simulation
for i=2:imax
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1 - damp1*(w1(i-1) - wsyn))*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2 - damp2*(w2(i-1) - wsyn))*DT;
    Del1(i)=Del1(i-1)+w1(i-1)*DT;
    Del2(i)=Del2(i-1)+w2(i-1)*DT;
    EP1=EP1MAG*(cos(Del1(i))+j*sin(Del1(i)));
    EP2=EP2MAG*(cos(Del2(i))+j*sin(Del2(i)));
    I_Norton(1,1)=EP1/(j*X1); I_Norton(2,1)=EP2/(j*X2); I_Norton(3,1)=0;
    V=inv(Y)*I_Norton;
    Pe1=real(V(1,1)*conj(I_Norton(1,1)));
    Pe2=real(V(2,1)*conj(I_Norton(2,1)));
    DelREF=DelREF+DT*wsyn;
    DelDIFF1_DEG(i)=(Del1(i)-DelREF)*180/pi;
    DelDIFF2_DEG(i)=(Del2(i)-DelREF)*180/pi;
    DelDIFF(i)=DelDIFF1_DEG(i)-DelDIFF2_DEG(i);
end
fprintf(' %s  %5d \n',' Number of time steps in prefault = ',imax);

% During Fault Transient
% build Y matrix with fault on line 1-2
Y(1,1)=1/(j*X1)+1/(Z12/3) + 1/Z13; Y(1,2)=0; Y(1,3)=-1/Z13;
Y(2,1)=0; Y(2,2)=1/(j*X2)+1/(2*Z12/3)+ 1/Z23; Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13; Y(3,2)=-1/Z23; Y(3,3)=1/ZLoad+1/Z13 + 1/Z23;
kmax = round( (Time_fault_clear - Time_fault_start)/DT );
%kmax is the number of time steps used during the fault simulation
for k=1:kmax
    i=imax+k;
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1 - damp1*(w1(i-1) - wsyn))*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2 - damp2*(w2(i-1) - wsyn))*DT;
    Del1(i)=Del1(i-1)+w1(i-1)*DT;
    Del2(i)=Del2(i-1)+w2(i-1)*DT;
    EP1=EP1MAG*(cos(Del1(i))+j*sin(Del1(i)));
    EP2=EP2MAG*(cos(Del2(i))+j*sin(Del2(i)));
    I_Norton(1,1)=EP1/(j*X1); I_Norton(2,1)=EP2/(j*X2); I_Norton(3,1)=0;
    V=inv(Y)*I_Norton;
    Pe1=real(V(1,1)*conj(I_Norton(1,1)));
    Pe2=real(V(2,1)*conj(I_Norton(2,1)));
    DelREF=DelREF+DT*wsyn;
    DelDIFF1_DEG(i)=(Del1(i)-DelREF)*180/pi;
    DelDIFF2_DEG(i)=(Del2(i)-DelREF)*180/pi;
    DelDIFF(i)=DelDIFF1_DEG(i)-DelDIFF2_DEG(i);
end
fprintf(' %s  %5d \n',' Number of time steps during fault = ',kmax);

% Post Fault Transient (breakers on faulted line have opened)
% build Y matrix with line 1-2 removed
Y(1,1)=1/(j*X1)+ 1/Z13; Y(1,2)=0; Y(1,3)=-1/Z13;
Y(2,1)=0; Y(2,2)=1/(j*X2)+ 1/Z23; Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13; Y(3,2)=-1/Z23; Y(3,3)=1/ZLoad+1/Z13 + 1/Z23;
nmax = round( (Time_max - Time_fault_clear)/DT );
%nmax is the number of time steps used to complete the simulation
for n=1:nmax
    i=imax+kmax+n;
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1 - damp1*(w1(i-1) - wsyn))*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2 - damp2*(w2(i-1) - wsyn))*DT;
    Del1(i)=Del1(i-1)+w1(i-1)*DT;
    Del2(i)=Del2(i-1)+w2(i-1)*DT;
```

```
    EP1=EP1MAG*(cos(Del1(i))+j*sin(Del1(i)));
    EP2=EP2MAG*(cos(Del2(i))+j*sin(Del2(i)));
    I_Norton(1,1)=EP1/(j*X1); I_Norton(2,1)=EP2/(j*X2); I_Norton(3,1)=0;
    V=inv(Y)*I_Norton;
    Pe1=real(V(1,1)*conj(I_Norton(1,1)));
    Pe2=real(V(2,1)*conj(I_Norton(2,1)));
    DelREF=DelREF+DT*wsyn;
    DelDIFF1_DEG(i)=(Del1(i)-DelREF)*180/pi;
    DelDIFF2_DEG(i)=(Del2(i)-DelREF)*180/pi;
    DelDIFF(i)=DelDIFF1_DEG(i)-DelDIFF2_DEG(i);
end
fprintf(' %s  %5d \n',' Number of time steps after fault = ',nmax);

plot(time1,DelDIFF1_DEG,time1,DelDIFF2_DEG)
```

Laboratory Experiment 10A: Making a Power System Reliable

Making a Power System Reliable

Objectives:

3. To understand the planning/design process that goes into making a power system reliable.
4. To understand the editing tools in PowerWorld

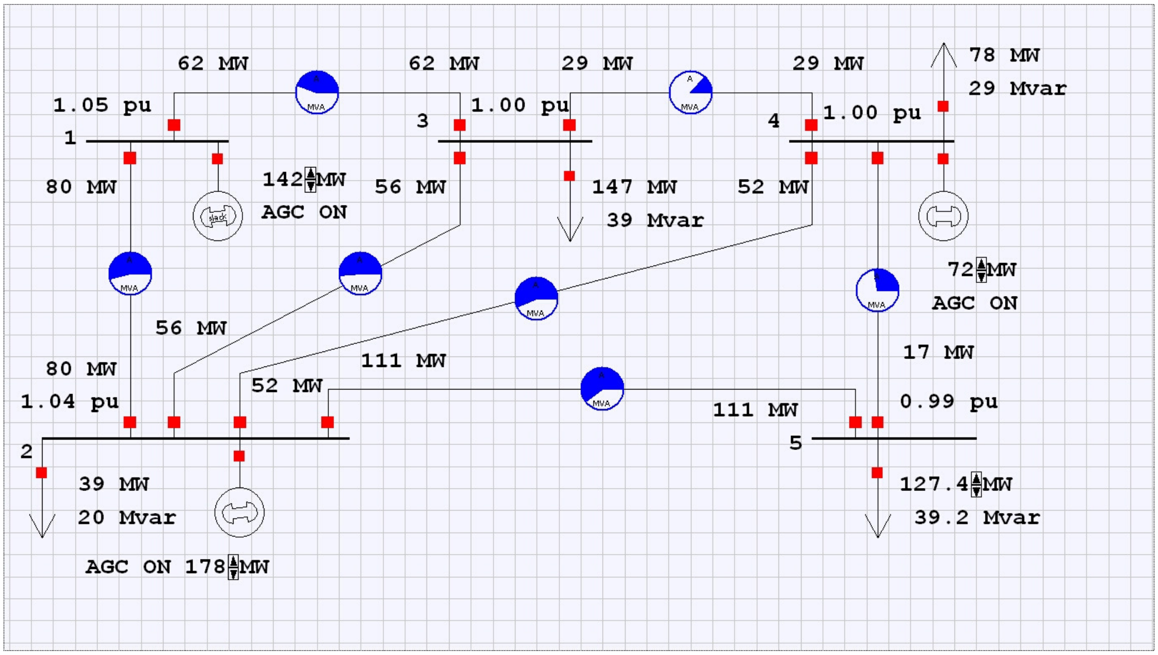
Laboratory Tasks and Report:

7. To test the reliability of a system we will apply the following rule:

   No single line or generator outage will leave the system with lines overloaded or bus voltages violating limits.

8. Given the PowerWorld case **PowerReliabilityCase.pwb** with the drawing file **PowerReliabilityCase.pwd**
   a. Read the case into PowerWorld and start it running
   b. Open each line (open a line by clicking on one breaker at the line's end) one at a time and observe the loading on all other lines.If another line is operating at or above 95% of its limits that outage is considered to have caused the system to fail. Similarly, if an open line causes any bus voltages to go outside the range 0.95 pu to 1.05 pu that case will also be considered to have caused the system to fail.
   c. Drop each generator one at a time (again just open the generator's breaker). PowerWorld will reallocate the lost generation to other generators. Again, any lines that are overloaded beyond 85% or buses which have a voltage outside the 0.95-1.05 limits will be considered to have caused the system to fail.
9. As a power system engineer you are going to make this system reliable. To do this you can add new equipment such as new transmission lines, new generators, capacitors on buses, etc. You will need to stop the simulator and put it into edit mode and then use the drawing tools and the information tools that can set the parameters for each new component added. If you add a transmission line that is in parallel with an existing line give the new line the same parameters as the existing line. If you add a generator, limits its capacity to 200 MW.
10. Add as many lines, generators, capacitors, etc as necessary so that at the end of the process you can run the tests above in 2b and 2c and no system failures are seen. Try to achieve this will as few new components added as necessary.

This is the five bus PowerReliabilityCase:



Laboratory Experiment 11: AGC and Economic Dispatch

AGC using *Simulink* and Economic Dispatch using *PowerWorld*

Objectives: Study the dynamic interaction between two control areas using *Simulink* modeling and economic dispatch using *PowerWorld*.

Laboratory Tasks and Report:

1. Study the dynamic interaction between two control areas using Simulink modeling. The MATLAB file for this is **AGC_Data.m**, which is located in this Folder. First launch MATLAB and open this file through it, and then execute it. Then double click on the *Simulink* file **AGC.mdl** located in this folder. Look at the various waveforms and comment on them. Adapted from Reference 6 in Chapter 12. See video clip# 15.
2. To run the simulation, first go to the Matlab workspace and enter AGC_Data which runs this simple bit of code to enter all the parameters into the workspace. Then run AGC and click on the arrow to run it.
    a. First run the simulation and capture a copy of each “scope” in the simulink window by double clicking on the scope and then clicking on the “binoculars” icon.
    b. Next you are to run the simulation with the ACE1 and ACE2 connections broken. This means that only the governors are controlling frequency. Again capture all the scope plots and compare them to those found in part a above.
    c. In parts a and b the generators are identical. You are now going to change the rotating mass of the lower generator. Left click on the box having 1/M1s+D1 for the lower generator and change the M1 parameter to M2, (Use TransferFcn Parameters) which is 10 times the size. Now rerun the simulation and get the scope plots. Explain the differences.
3. In PowerWorld, assume that the generation at Bus 2 is by two generators with different marginal costs, as shown in **Load_Sharing.pwb**. Justify the load sharing between the generators.

    In this system the generator on the left at bus 2 has a different cost curve from that on the right on bus 2. You can find the cost functions by first clicking on one of the generators, then left clicking and selecting “Generator Information Dialog”, and then clicking on “Costs” tab, and then the “Output Cost Model” tab. These generator use a cubic cost model. That is:

$$Cost(P_i) = A_i + B_iP_i + C_iP_i^2 + D_iP_i^3$$

However, the value of the D parameter is zero, so they are actually quadratic functions.

The generation is allocated so that the "Incremental Cost" of each generator is equal.

That is: $\frac{dCost_1(P_1)}{dP_1} = \frac{dCost_2(P_2)}{dP_2}$ or $B_1 + 2C_1P_1 = B_2 + 2C_2P_2$

You are to obtain the values for B and C for each generator on bus 2, note that the generation starts out at 100 MW and 300 MW but this changes.

First open the breaker on the generator at bus 1 so that the generators at bus 2 are the only ones supplying the load. Now start up the PowerWorld simulation and note how it allocates the generation to each generator on bus 2.

Now, while it is running, click on the 500 MW load and change it to 600 MW and again record the generation allocation on bus 2.

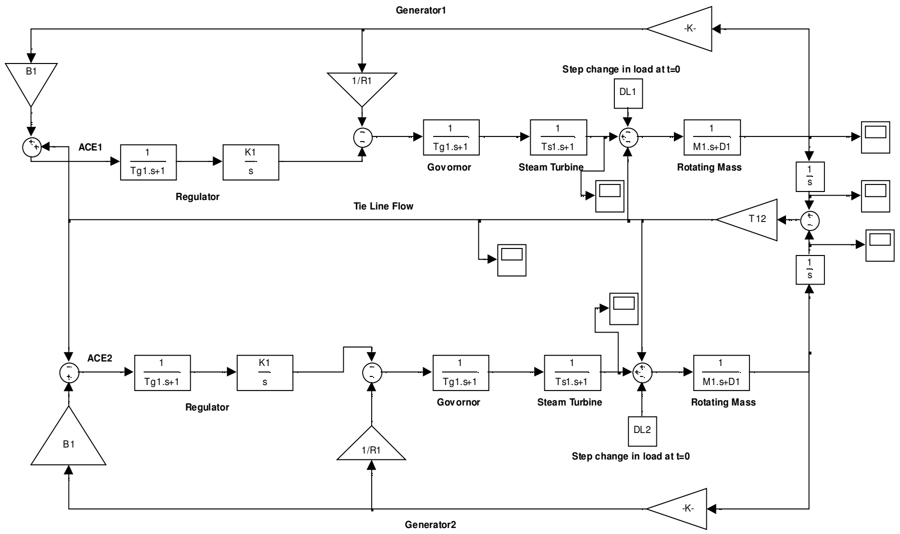
Equal incremental cost means that two equations are solved:

$B_1 + 2C_1P_1 = B_2 + 2C_2P_2$ and $P_1 + P_2 = P_{total}$ where P_{total} is the total generation being supplied by the two generators. Solve these two equations for each case you ran and indicate whether PowerWorld is loading them according to incremental loading.

AGC_Data.m

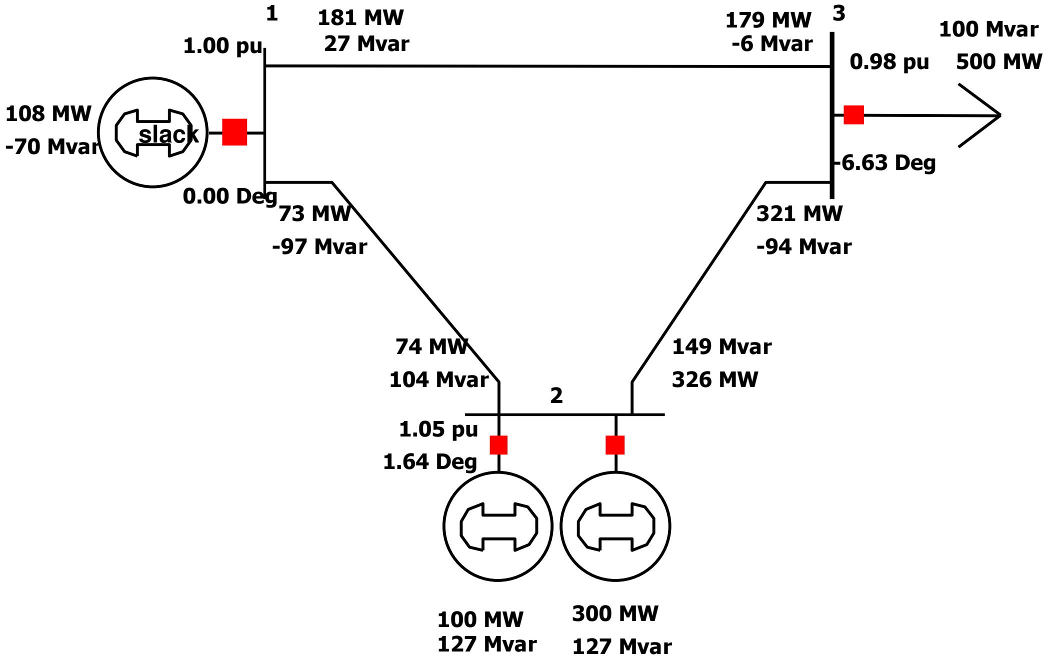
```
% Data for Example 12-3
H1=5.3; % H=5.3 seconds
M1=(2*H1)/(2*pi*60) % 2*pi*60 denotes the synchronous speed in
rad/s
H2=500.
M2=(2*H2)/(2*pi*60)
D1=0.75/(2*pi*60) % D1 is the damping coefficient
T12=0.1;
R1=0.167;
Tg1=.26;
Ts1=.26;
DL1=1; % 1 percent so the results are in per cent
DL2=0;
B1=1/R1+D1;
thetar=0;
K1=0.001*((2*pi*60)); % Controller gain for the ACE loop
```

AGC.mdl



The above simulation is with D1=0.75/wsyn and K1=0.001/wsyn with timeconstants Tg1,Ts1 in seconds

Load_Sharing.pwb



Problem 5-8
Confirm the MATLAB Results of Example 5-4.

Laboratory Experiment 12: Short Circuit Faults and Overloading of Transmission Lines

Transmission Line Short Circuit Faults using MATLAB and PowerWorld, and Overloading of Transmission Lines using *PowerWorld*

Objectives: To study the effect of short-circuit faults and overloading of transmission lines.

Laboratory Tasks and Report:

1. Simulate the fault in Example 13-2. The MATLAB file for this example is **SimpleSystemFault.m**, which is located in this Folder. First launch MATLAB and open this file through it, and then execute it. Using the program code given, change the code for a three phase and a single line to ground fault at bus 3 instead of bus 2. Show the code for the bus 3 fault in your report.
2. The *PowerWorld* file for Example 13-2 is **SimpleSystemFault.pwb**; double click on it and compare results with that from the MATLAB simulation. Under Run Mode find Fault Analysis and click on it, then select bus 2 in the window marked "Choose the faulted bus." Then select "Bus Fault", and either three phase or single line to ground. When you are done with selections hit "calculate" in the lower left. You can cause the results to be displayed on the one line by selecting "All Phases" in the one line box and you can show results in either pu or in Amps. You should do both three phase and one line to ground faults at both bus 2 and bus 3 and compare to the Matlab results you got.
3. The PowerWorld file for this example is **ShortCircuitFault.pwb** (see video clips# 16 and 17), which is located in this Folder. Open the file and run it. Calculate three phase and single line to ground faults at bus 1, on the 1 to 3 line half way between bus 1 and 3 and on bus 3.

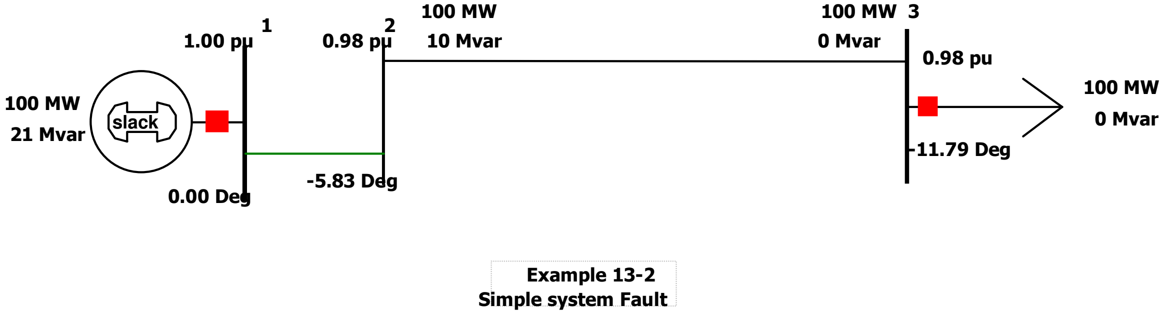
   For the line fault you need to click on the line fault tab and tell the program where on line 1 to 3 to place the fault.

   For each fault type and location you are to capture the results on the diagram showing the currents in amperes for all three phases. Which fault has the largest fault currents.

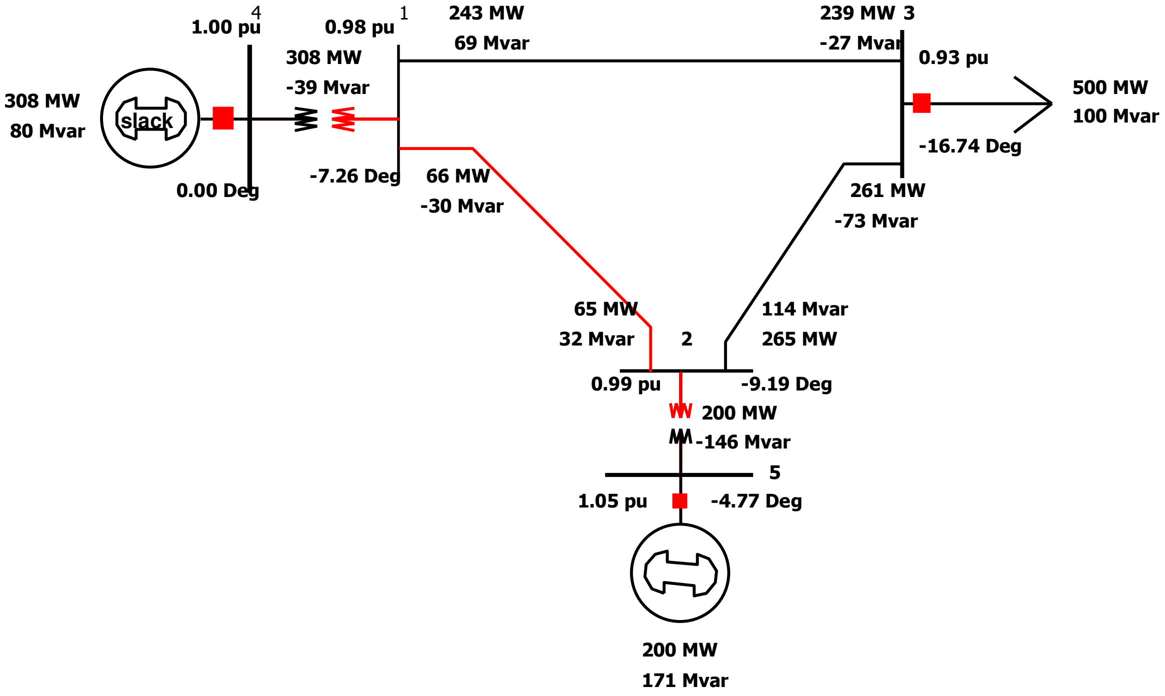
SimpleSystemFault.m

```
% Example 13-2; simple system
% Fault; 3-phase at bus 2
%Pre-Fault
V3_a1=0.98*exp(-j*11.79*pi/180);
I3_a1=(1/0.98)*exp(-j*11.79*pi/180);
Rload=0.98*0.98,
Ea=1+j*0.12*I3_a1,
[Th_Ea,Amp_Ea]=cart2pol(real(Ea),imag(Ea));
Th_Ea_deg=Th_Ea*180/pi,
Amp_Ea
Power=Ea*conj(I3_a1),
% 3-phase fault at bus 2
Ifault=Ea/(j*(0.12+0.1)),
[ang,Ifault_Mag]=cart2pol(real(Ifault),imag(Ifault));
Ifault_AngDEG=ang*180/pi,
Ifault_Mag
% SLG fault at bus 2
ITH=Ea/(j*0.32+Rload);
VTH=(j*0.1+Rload)*ITH;
ZTH=(j*0.1+Rload)*(j*0.22)/((j*0.1+Rload)+(j*0.22));
Z2=(j*0.1+Rload)*(j*0.22)/((j*0.1+Rload)+(j*0.22));
Z0=(j*0.2+Rload)*(j*0.10)/((j*0.2+Rload)+(j*0.10)); % delta wye-grounded transfomer bypasses X0 of generator
Ia1=VTH/(ZTH+Z2+Z0);
Ifault=3*Ia1;
[ang,Ifault_Mag]=cart2pol(real(Ifault),imag(Ifault));
Ifault_AngDEG=ang*180/pi,
Ifault_Mag
```

SimpleSystemFault.pwb



ShortCircuitFault.pwb



Example 13-3

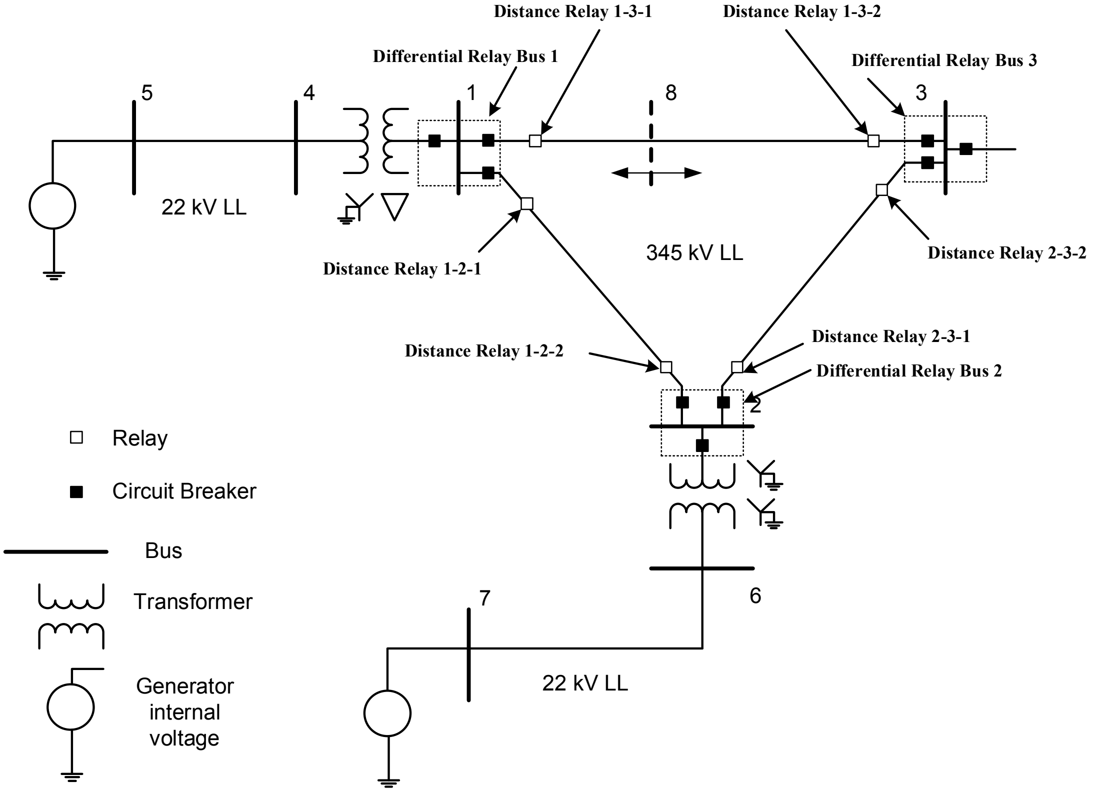
Laboratory Experiment 12A: Fault Analysis with Relay Settings

Fault Analysis with Relay Settings

Objectives: To study a power system with faults and determine relay settings based on calculated fault currents

Laboratory Tasks and Report:

1. You are given the two generator system below. You will use the program to analyze relay settings. The network you are to solve is shown below:



2. You are going to run several faults on this system and then see what various relays would have as input given the fault voltages and currents that the program outputs. The program is in Matlab and is called **FaultAnalysis_RelaySettings.m** and the data describing the network is in a separate file called **NetworkData.m** and you will need both to run the program.
3. The program will print out the Line to Neutral and Line to Line voltages at all buses in the network, as well as the zero, positive and negative sequence currents and the abc currents on each branch of the network. (voltages are in kV currents in amperes)
4. The program allows you to select three phase or line to ground fault, it allows you to select a bus or line fault, and which bus the fault is on. If it is a line fault it allows you to

select how far down the line the fault appears and adds a new bus at that point where the fault takes place.

The positive sequence impedances in per unit for lines 1-3 and 2-3 are: Z13 = 0.005+0.044j and Z23=0.0025+0.022j the system has two voltages, 22 kV and 345 kV. The 22 kV is the voltage of the generators (buses 4, 5, 6 and 7 only) and buses 1, 2, and 3 are at 345 kV. When you run the program it prints out the Sbase, Vbase_LL and Vbase_LN, Ibase and Zbase for all regions. The 22 kV regions are considered region 1 (Vbase1_LL, Vbase1_LN, Ibase1 and Zbase1) and the 345 kV region is region 2 (Vbase2_LL, Vbase2_LN, Ibase2 and Zbase2).

You are going to capture the kV and ampere reading seen by the distance relays 1-3-1, 1-3-2, 2-3-1 and 2-3-2 as shown on the diagram. The program gives you kV LN and amperes flowing during a fault. You will use the following equation to calculate the impedance "seen" by each relay during a set of faults (given below), you will then divide the impedance seen by the actual impedance of each line to determine the distance to the fault. Note that you have to use the line impedance in ohms, not per unit, so use the Zbase2 to convert to actual impedance (use the impedance magnitude):

$$\left|Z_{measured}\right|\angle\theta_Z = \frac{\left|V\right|\angle\theta_V}{\left|I\right|\angle\theta_I}$$

Or

$$\left|Z_{measured}\right| = \frac{\left|V\right|}{\left|I\right|}$$

Where V is in volts (not kV not pu) and I is in amps (not pu) result is Z in ohms

Now calculate the distance to the fault as $d = \frac{Z_{measured}}{Z_{line}}$ where both $Z_{measured}$ and Z_{line} are in ohms, then d is the fraction of the total line's impedance as measured by the relay, which should be the same as the fraction of the line distance where the fault happens.

Task 1: Run the following faults in this order:

Three phase fault at 26 % of distance from bus 1 to bus 3 on line 1-3
Three phase fault at 50 % of distance from bus 1 to bus 3 on line 1-3
Three phase fault at 75 % of distance from bus 1 to bus 3 on line 1-3
Three phase fault on bus 3
Three phase fault at 75 % of distance from bus 2 to bus 3 on line 2-3 (i.e., 25 % of distance from bus 3 to bus 2 on line 2-3)
Three phase fault at 50 % of distance from bus 2 to bus 3 on line 2-3 (i.e., 50 % of distance from bus 3 to bus 2 on line 2-3)
Three phase fault at 25 % of distance from bus 2 to bus 3 on line 2-3 (i.e., 75 % of distance from bus 3 to bus 2 on line 2-3)

Calculate the distance to the fault as calculated by each relay for each case. Note that inline faults result in a new bus called bus 8, so the fault currents seen on relay 1-3-1 are calculated from the line to neutral voltage at bus 1 and the current on one of the phases as seen on line 1-8. Do these calculations for distance calculated at the other relays (1-3-2, 2-3-1, and 2-3-2) for all the faults.

Are the faults measuring the distance correctly? Note that when the fault is in line 2-3 the relay at 1-3-1 sees the entire line impedance Z13 and part of the impedance of line 2-3 – so you need to add them together.

Task 2: Repeat the above with a single line to ground fault instead of a three phase fault, once again, calculate the distance measured using phase a(using Va_LN and Ia) as well as phases b and c using their respective LN voltages and phase currents). Can you still get the distance to the fault in the case of a single line to ground fault.

Laboratory Experiment 13: Switching Over-Voltages and Modeling of Surge Arresters

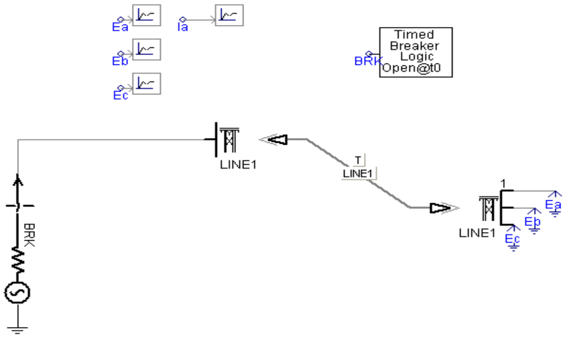
Switching Over-Voltages and Modeling of Surge Arresters using PSCAD/EMTDC

Objectives: To study over-voltages resulting from switching of transmission lines and limiting them by sing ZnO arresters.

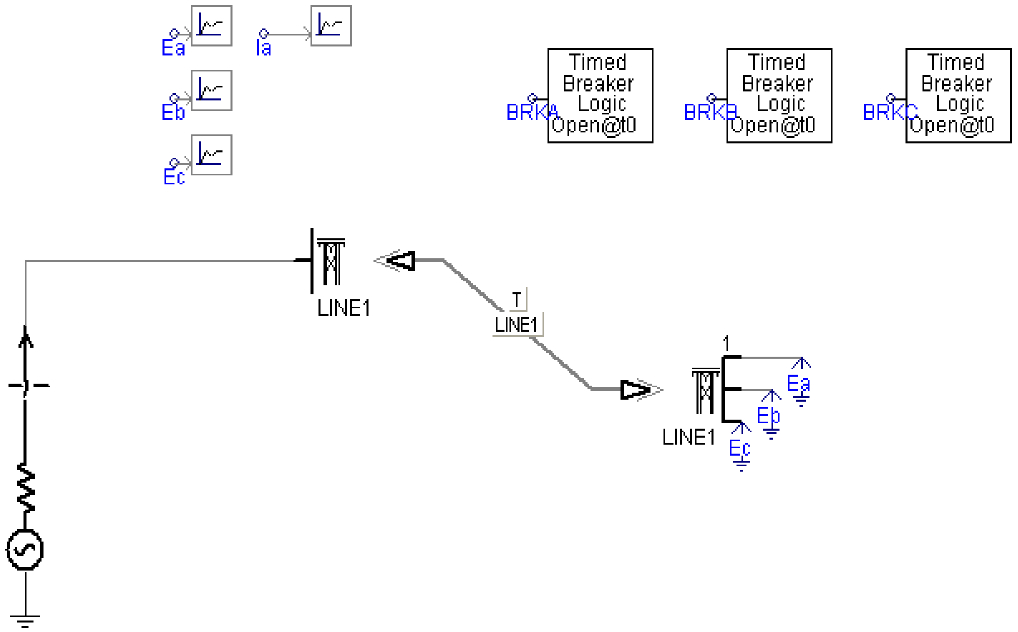
Laboratory Tasks and Report:

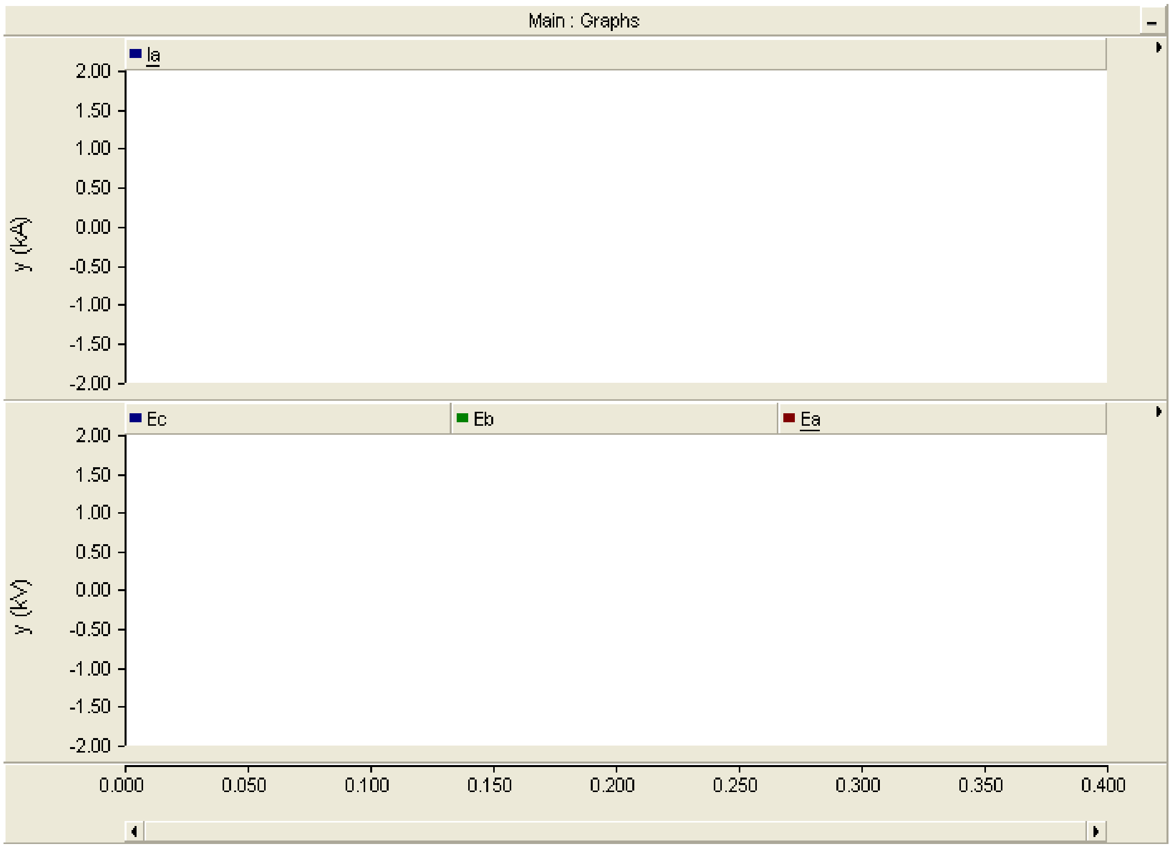
1. Model the reclosing of a transmission line as described by the *PSCAD/EMTDC* file **TL_Energization.psc** (see video clip# 18). Obtain various waveforms and comment on them.
2. Include the pre-insertion resistors, as described by the *PSCAD/EMTDC* file **TL_Energization_Preinsertion.psc** (see video clip# 18). Obtain various waveforms and comment on them.
3. Model the surge arresters to limit the over-voltages at the receiving end to 400 kV. Obtain various waveforms and comment on them; **TL_Energization_MOV.psc** (see video clip# 18)**.**

TL_Energization.psc



TL_Energization_Preinsertion.psc





TL_Energization_MOV.psc