Power Systems Laboratory

User Manual

Department of Electrical and Computer Engineering

University of Minnesota

Revised : July 22, 2008

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.

Laboratory Experiment 1

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

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)

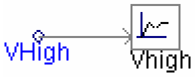
6) Setting Properties:
   To set the properties double click on any component and change the parameters.

   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

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.

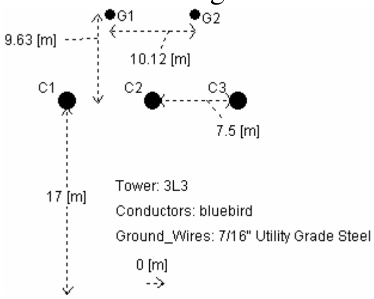


Fig. 4-8 A 345-kV, single-conductor per phase, transmission system.

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	$R(\Omega/km)$	$\omega L(\Omega/km)$	$\omega C(\mu$□$/km)$	$Z_c(\Omega)$	$SIL(MW)$
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 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). (For help regarding the program , type “help threeBusPowerFlow” in Command window)
    a. Annotate the files based on the material and equations of Chapter 5.
    b. Execute the 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.
    c. Add the line shunt capacitances as described in Chapter 5, execute the files to obtain power and reactive power flow using both methods, and compare the results with part (a).
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 program 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.

Newton Raphson Function

```
function [P Q] = runNewtonRaphson(voltage, angle, Psp, Qsp, CAPFLAG)

% ----------- initialize parameters ---------- %
MAX_ITER = 4;
TOL = 0.001;
convFlag = 1;
iter = 1;
deltaAngle = zeros(1,3);
deltaVoltage = zeros(1,3);

% ---------- initialize variables ------------ %

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

```
P = zeros(3,3);
Q = zeros(3,3);

J = zeros(3,3);
[Y Z Zc] = inputBusData(CAPFLAG);      % get conductance and
susceptance
G = real(Y);
B = imag(Y);

% ------- begin Newton-Raphson procedure ------- %

while( convFlag==1 && iter < MAX_ITER)

    angle = angle + deltaAngle;
    voltage = voltage + deltaVoltage;

    % ------ Jacobian equations ------------- %

    J(1,1)=voltage(2)*(voltage(1)*(-G(2,1)*sin(angle(2) - angle(1))+
B(2,1)*cos(angle(2) - angle(1))) + voltage(3)*(-G(2,3)*sin(angle(2) -
angle(3))+B(2,3)*cos(angle(2)-angle(3))));
    J(1,2)=voltage(2)*(voltage(3)*(G(2,3)*sin(angle(2) - angle(3)) -
B(2,3)*cos(angle(2) - angle(3))));
    J(1,3)=voltage(2)*((G(2,3)*cos(angle(2) - angle(3)) +
B(2,3)*sin(angle(2) - angle(3))));
    J(2,1)=voltage(3)*(voltage(2)*(G(3,2)*sin(angle(3) - angle(2)) -
B(3,2)*cos(angle(3) - angle(2))));
    J(2,2)=voltage(3)*(voltage(1)*(-G(3,1)*sin(angle(3) - angle(1)) +
B(3,1)*cos(angle(3) - angle(1))) + voltage(2)*(-G(3,2)*sin(angle(3)-
angle(2))+B(3,2)*cos(angle(3)-angle(2))));
    J(2,3)=2*G(3,3)*voltage(3) + voltage(1)*(G(3,1)*cos(angle(3)-
angle(1)) + B(3,1)*sin(angle(3) - angle(1))) +
voltage(2)*(G(3,2)*cos(angle(3) - angle(2))+B(3,2)*sin(angle(3) -
angle(2)));
    J(3,1)=voltage(3)*(voltage(2)*(G(3,2)*cos(angle(3) - angle(2)) -
B(3,2)*sin(angle(3) - angle(2))));
    J(3,2)=voltage(3)*(voltage(1)*(G(3,1)*cos(angle(3) - angle(1)) +
B(3,1)*sin(angle(3) - angle(1))) + voltage(2)*(G(3,2)*cos(angle(3) -
angle(2)) + B(3,2)*sin(angle(3) - angle(2))));
    J(3,3)= -2*B(3,3)*voltage(3) + voltage(1)*(G(3,1)*sin(angle(3) -
angle(1)) - B(3,1)*cos(angle(3) - angle(1))) +
voltage(2)*(G(3,2)*sin(angle(3) - angle(2)) - B(3,2)*cos(angle(3) -
angle(2)));

  % ---------Bus Voltages--------%
    V(1,1)=voltage(1)*exp(sqrt(-1)*angle(1));
    V(2,1)=voltage(2)*exp(sqrt(-1)*angle(2));
    V(3,1)=voltage(3)*exp(sqrt(-1)*angle(3));

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

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

    % -----Mismatch at PQ and PV buses-------%
    Mismatch(1,1) = Psp(2) - real(S(1,2));
    Mismatch(2,1) = Psp(3) - real(S(1,3));
    Mismatch(3,1) = Qsp(3) - imag(S(1,3));

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

    % -------Calculate Power Flow on the Transmission Lines-------%

    if(CAPFLAG)
        for i = 1:length(V)
            for j = 1:length(V)
                if(i~=j)
                    P(i,j) = real(V(i,1)*conj(V(i,1) - V(j,1)/Z(i,j)) +
V(i,1)/Zc(i,j));
                    Q(i,j) = imag(V(i,1)*conj(V(i,1) - V(j,1)/Z(i,j)) +
V(i,1)/Zc(i,j));
                else
                    P(i,j) = real(S(1,i));
                    Q(i,j) = imag(S(1,i));
                end;
            end;
        end;

    else
        for i = 1:length(V)
            for j = 1:length(V)
                if(i~=j)
                    P(i,j) = real(V(i,1)*conj(V(i,1) - V(j,1)/Z(i,j)));
                    Q(i,j) = imag(V(i,1)*conj(V(i,1) - V(j,1)/Z(i,j)));
                else
                    P(i,j) = real(S(1,i));
                    Q(i,j) = imag(S(1,i));
                end;
            end;
        end;
    end;

    displayResults(P,Q,V,iter);

     if max(abs(Mismatch)) > TOL,
        convFlag=1;
    else
        convFlag=0;
    end
    iter = iter + 1;
end

return;
```

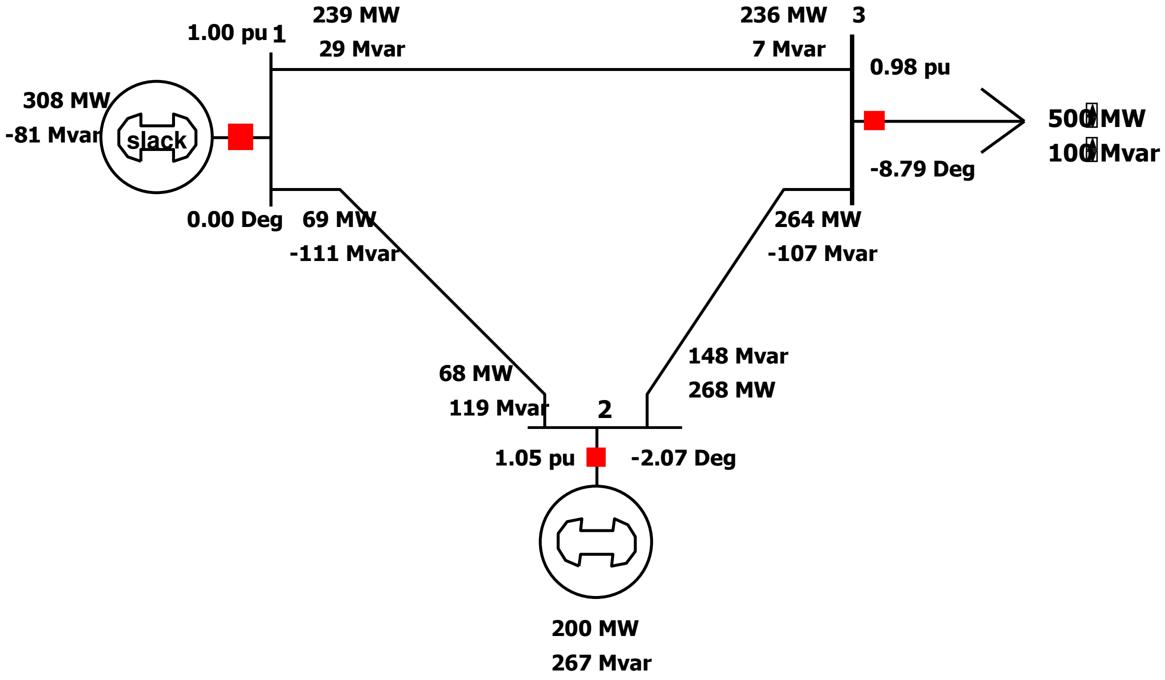
Gauss-Seidel Function

```
function [P Q] = runGaussSeidel(v, angle, Psp, Qsp, CAPFLAG)

MAX_ITER = 20;
iter = 1;
convFlag = 1;
TOL  = 0.001;

% ------- bus type and admittance ---------- %

[Y Z Zc] = inputBusData(CAPFLAG);
Bustype(1) = 'S';                          % PQ bus
Bustype(2) = 'G';                          % PV bus
Bustype(3) = 'L';                          % PQ bus


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

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


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

Psched = zeros(1,length(Z));
Qsched = zeros(1,length(Z));

% ------ Set up power schedule --------- %

for i = 1:length(Z)
    Psched(i)= Psp(i);
    Qsched(i)= Qsp(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));
    if(CAPFLAG)
         for j = 1:length(Z)
                    if(i~=j)
                        P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
                        Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
                    else
                        P(i,j) = real(S(i));
                        Q(i,j) = imag(S(i));
                    end;
            end;
    else
        for j = 1:length(Z)
                if(i~=j)
                    P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    else
                        P(i,j) = real(S(i));
                    Q(i,j) = imag(S(i));
                    end;
            end;
    end;
end;


    while(convFlag==1 && iter < MAX_ITER)
        MAXDP = 0.0;
        MAXDQ = 0.0;

        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));
            if(CAPFLAG)
         for j = 1:length(Z)
                    if(i~=j)
                        P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
                        Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
                    else
                        P(i,j) = real(S(i));
                        Q(i,j) = imag(S(i));
                    end;
            end;
        else
        for j = 1:length(Z)
                if(i~=j)
                    P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    else
                        P(i,j) = real(S(i));
                    Q(i,j) = imag(S(i));
                    end;
            end;
        end;
```

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

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

                if deltaq(i) > MAXDQ
                    MAXDQ = deltaq(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));
           if(CAPFLAG)
         for j = 1:length(Z)
                    if(i~=j)
                        P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
```

```
                        Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)) +
v(i)/Zc(i,j));
                    else
                        P(i,j) = real(S(i));
                        Q(i,j) = imag(S(i));
                    end;
            end;
       else
        for j = 1:length(Z)
                if(i~=j)
                    P(i,j) = real(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    Q(i,j) = imag(v(i)*conj(v(i) - v(j)/Z(i,j)));
                    else
                        P(i,j) = real(S(i));
                    Q(i,j) = imag(S(i));
                    end;
            end;
      end;


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

        displayResults(P,Q,v,iter);

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

        if MAXDP < TOL
            if MAXDQ < TOL
                convFlag = 0;
            end
        end
        iter = iter + 1;
    end
    return;
```

3Bus_PowerFlow.pwb



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

Laboratory Experiment 5

Including Transformers in Power Flow using PowerWorld and Confirmation by MATLAB

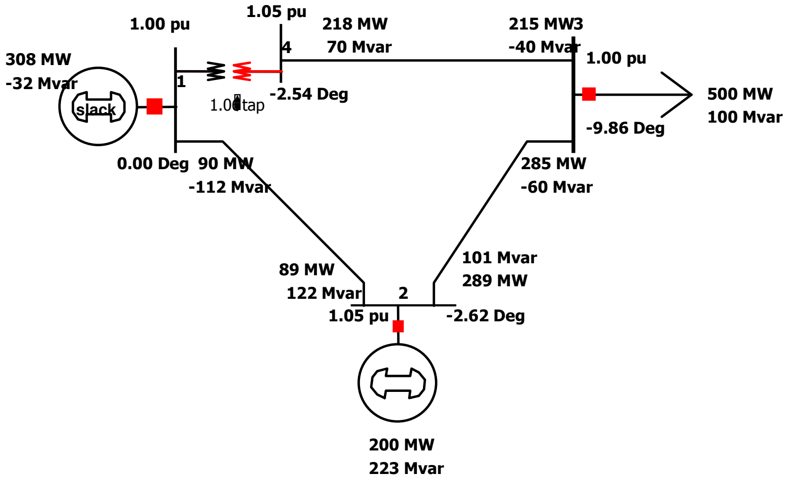
Objectives: To look at the influence of including a tap-changer and a phase-shifter on power flow and bus voltages.

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. Represent this auto-transformer by means of a pi-circuit of Fig. 6-17b in a MATLAB program, using the results of part a, to confirm the results of part a.

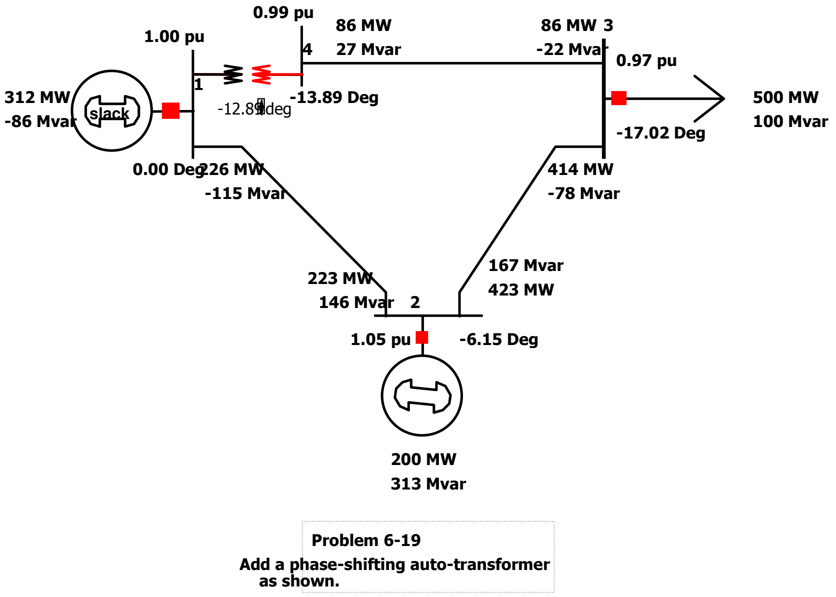
2. Including a Phase-Shifter **(PowerFlow_PhaseShift.pwb**; see video clip# 7**)**
   a. A phase-shift transformer is added between buses 1 and 4 (newly created) as shown in the *PowerWorld* file **PowerFlow_PhaseShift.pwb**, which is located in this Folder. Double click on this file or open it through *PowerWorld*. The phase-shift between buses 1 and 4 is such that $V_1 \angle -15^0$ results in $V_4 \angle 0$. 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. Represent this phase-shift transformer by means of Eq. 6-32 in a MATLAB program, using the results of part a, to confirm the results of part a.

PowerFlow_AutoTransformer.pwb



Problem 6-18
Add an tap-changing auto-transformer
as shown.

PowerFlow_PhaseShift.pwb



Problem 6-19

Add a phase-shifting auto-transformer as shown.

Laboratory Experiment 6

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

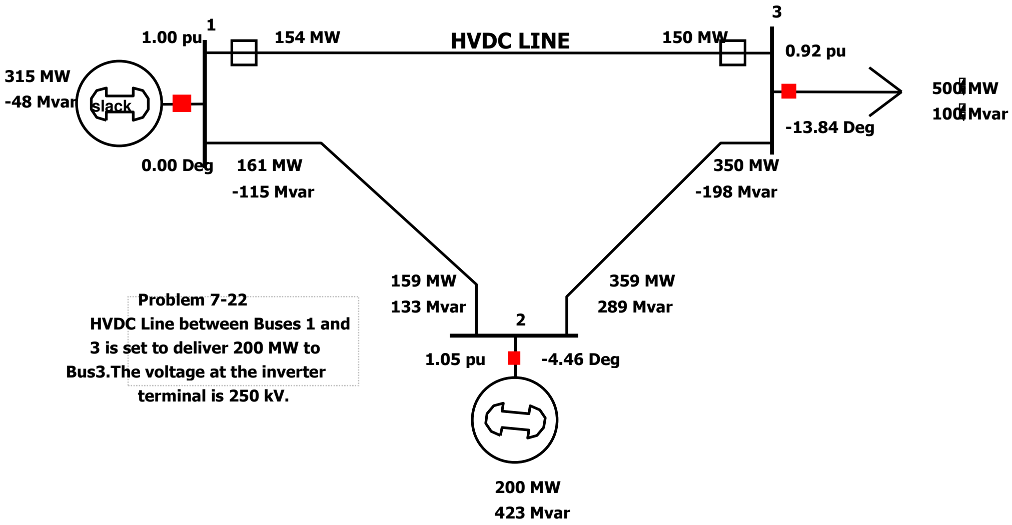
Objectives:

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.

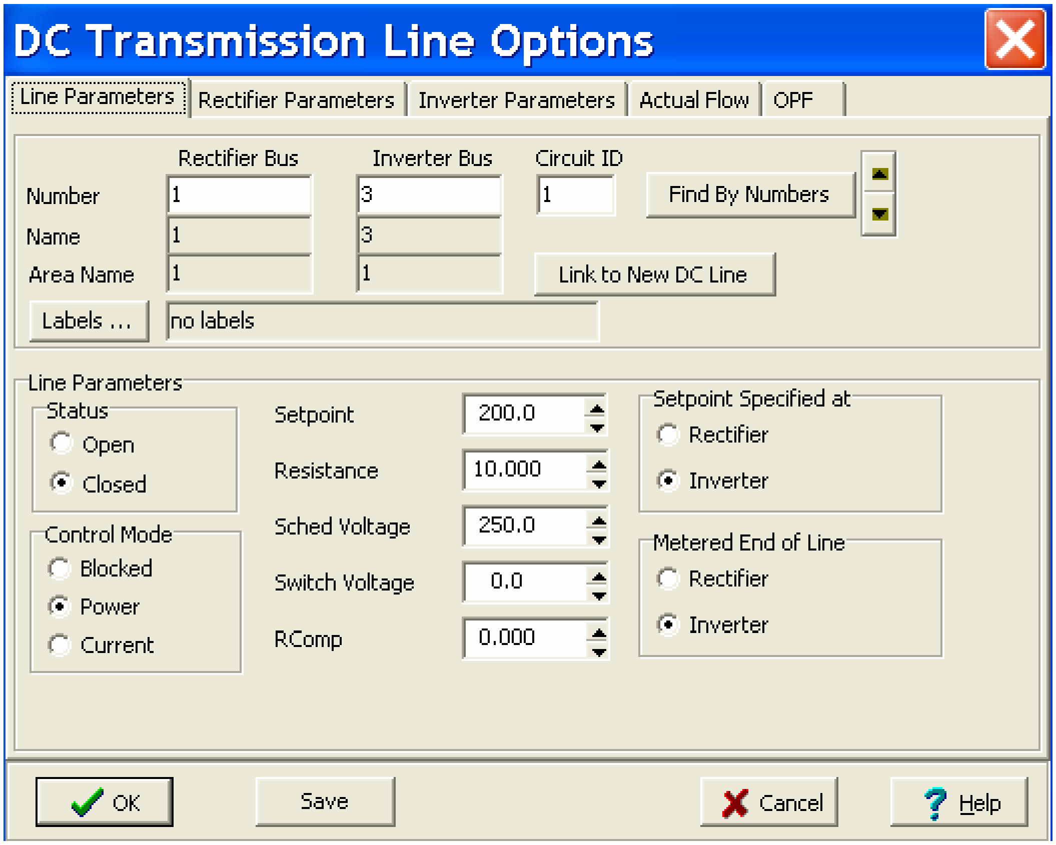
Laboratory Tasks and Report:

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.
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.
6. Obtain harmonic components of secondary line current of Wye-Delta Transformer and harmonic components of the source current in the rectifier and inverter. What is the significant difference between two of them? Why?

PowerFlow_HVDCline.pwb



Problem 7-22
HVDC Line between Buses 1 and 3 is set to deliver 200 MW to Bus3.The voltage at the inverter terminal is 250 kV.



DC Transmission Line Options

Line Parameters | Rectifier Parameters | Inverter Parameters | Actual Flow | OPF

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

Find By Numbers
Link to New DC Line
Labels ...

no labels

Line Parameters

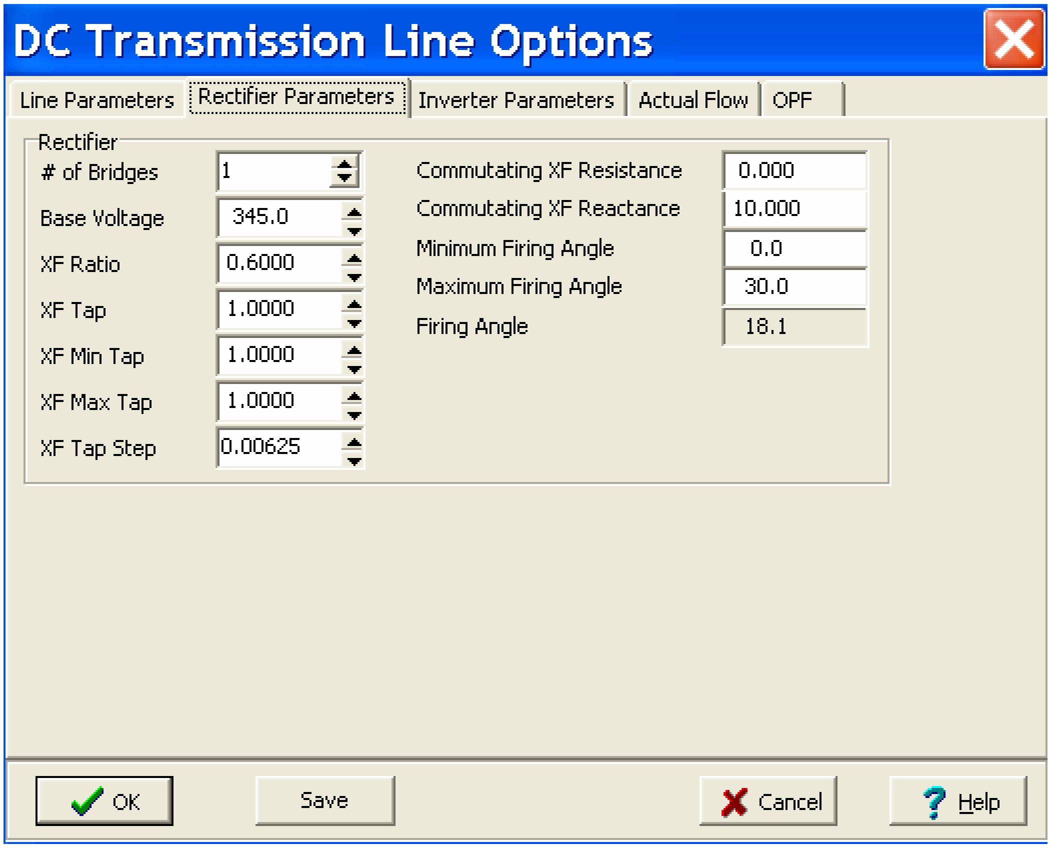
Status
○ Open
● Closed

Control Mode
○ Blocked
● Power
○ Current

Setpoint 200.0
Resistance 10.000
Sched Voltage 250.0
Switch Voltage 0.0
RComp 0.000

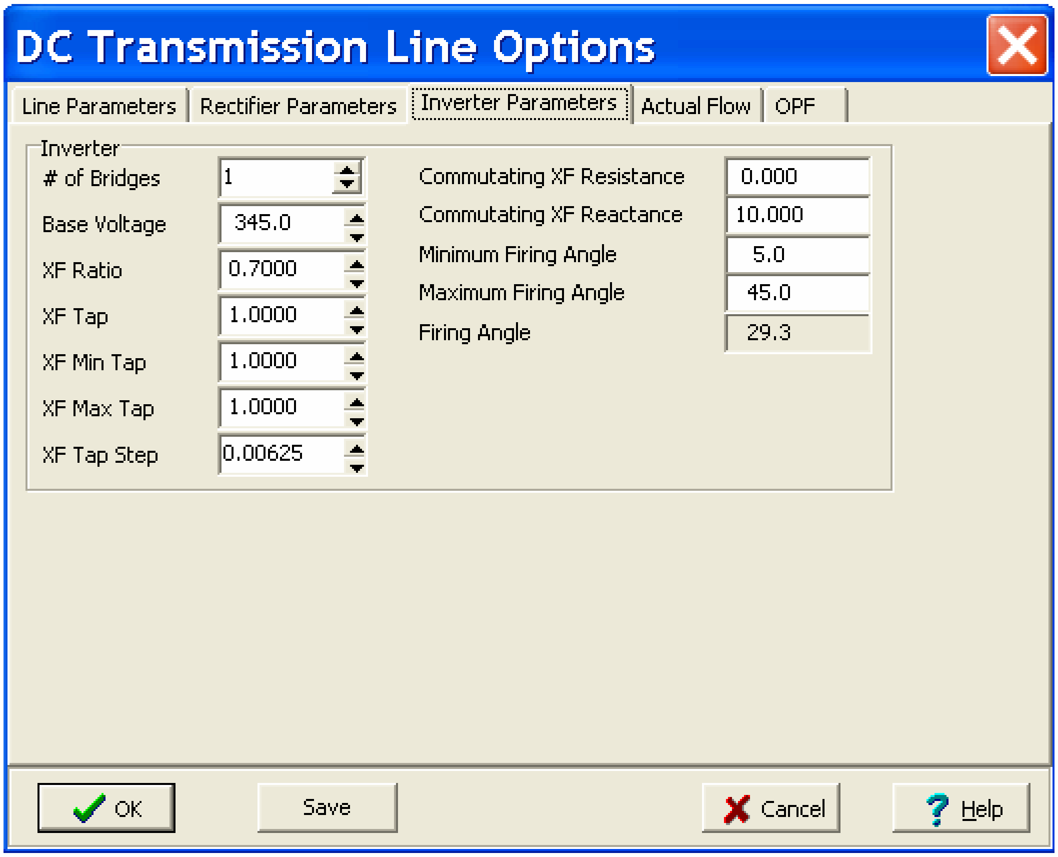
Setpoint Specified at
○ Rectifier
● Inverter

Metered End of Line
○ Rectifier
● Inverter



Rectifier

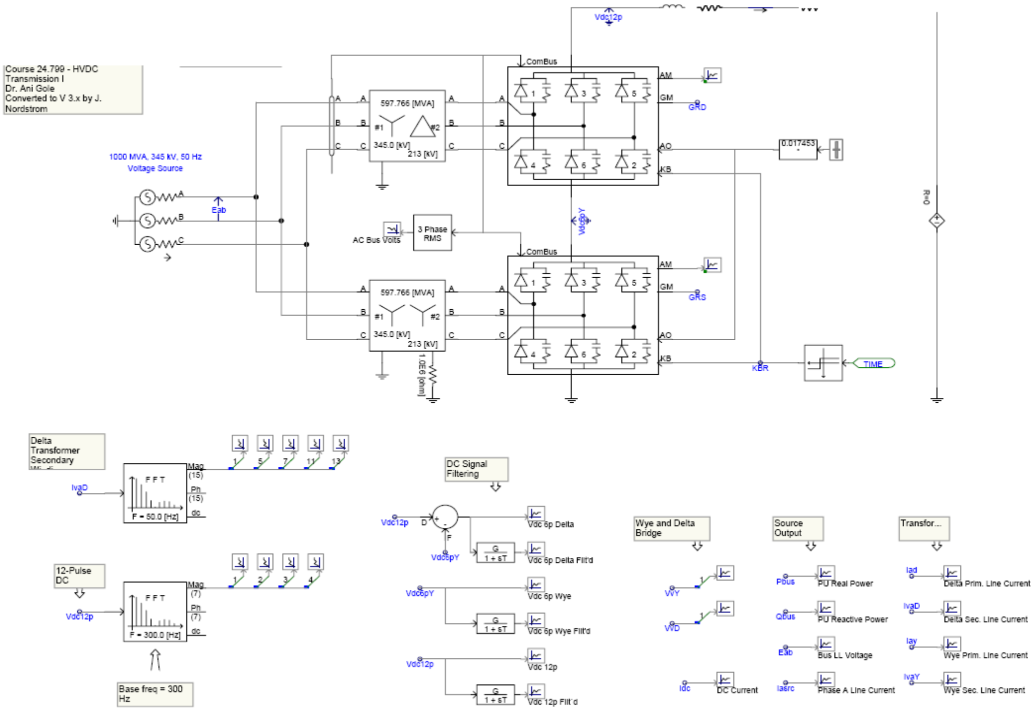
# of Bridges	1	Commutating XF Resistance	0.000
Base Voltage	345.0	Commutating XF Reactance	10.000
XF Ratio	0.6000	Minimum Firing Angle	0.0
XF Tap	1.0000	Maximum Firing Angle	30.0
XF Min Tap	1.0000	Firing Angle	18.1
XF Max Tap	1.0000		
XF Tap Step	0.00625		



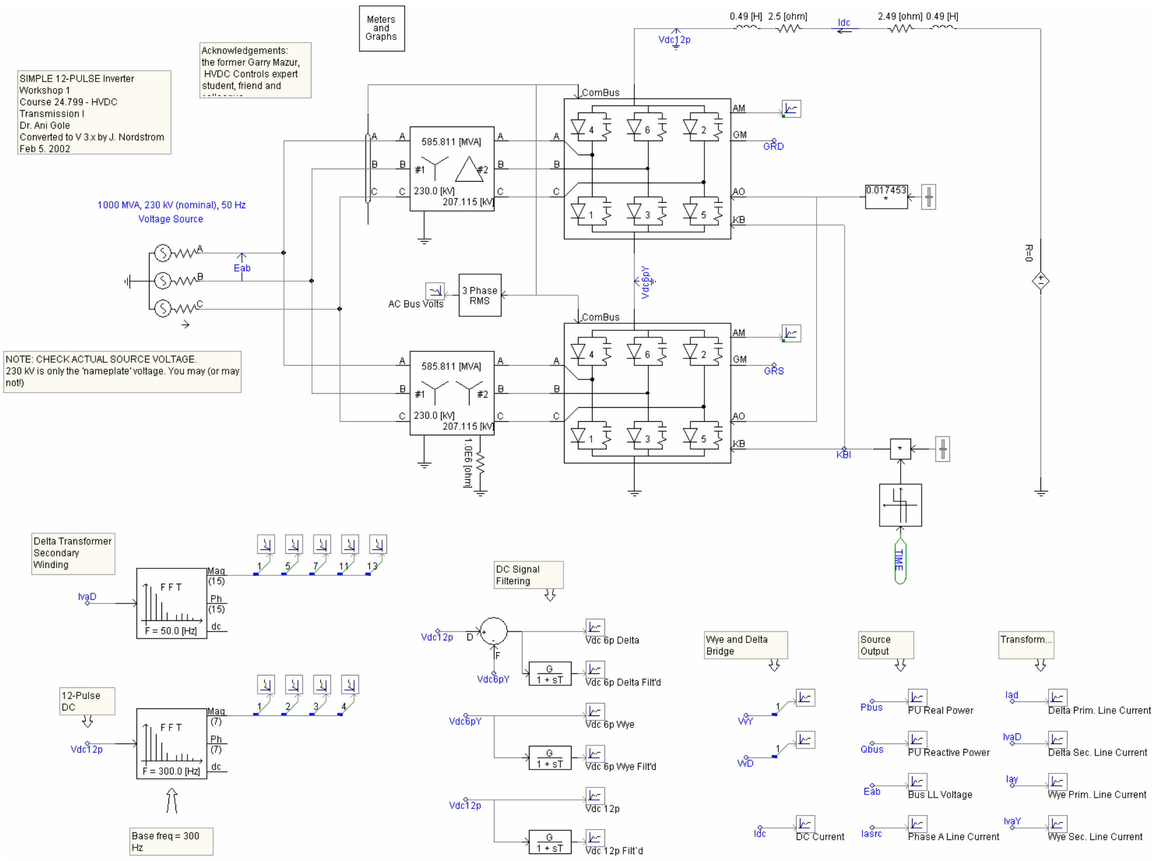
Inverter

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

HVDC_Rectifier.psc


Course 24.799 - HVDC Transmission I
Dr. Ani Gole
Converted to V 3.x by J. Nordstrom

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

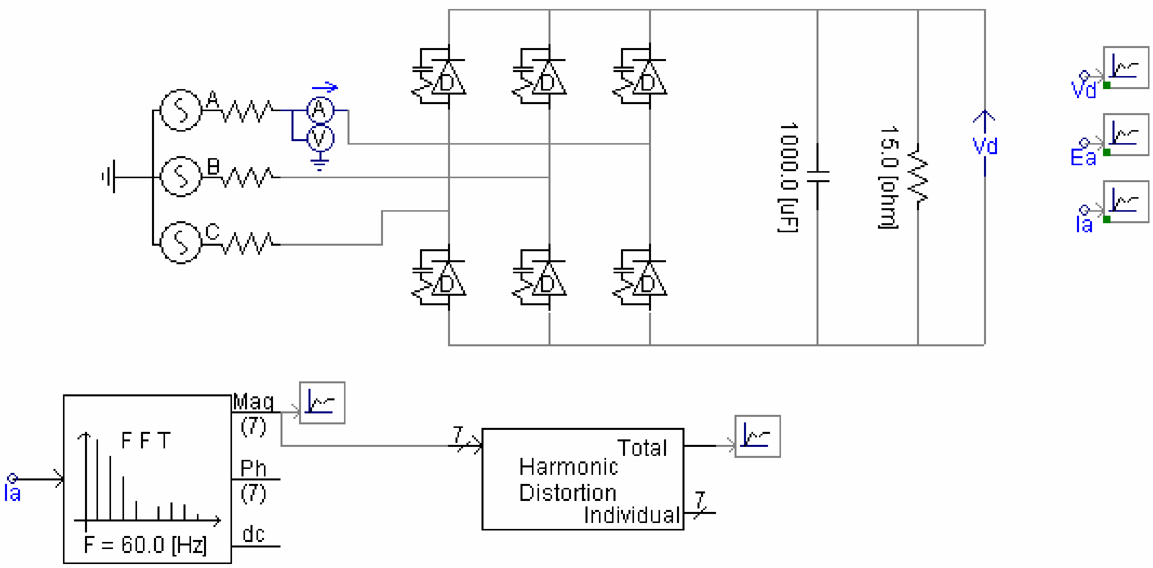
Power Quality

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.

Laboratory Experiment 8

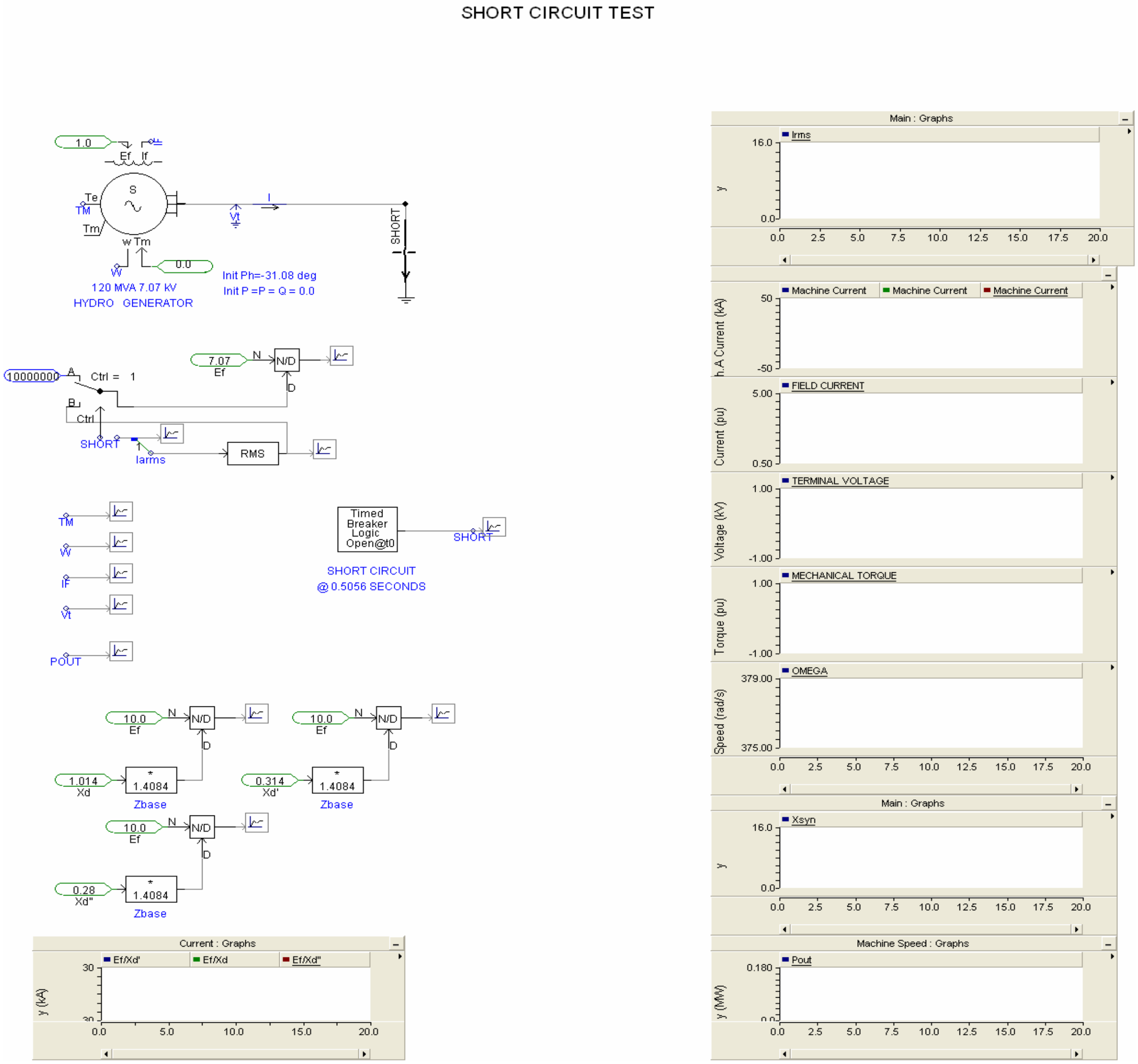
Synchronous Generators

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



Laboratory Experiment 9

Voltage Regulation

Objectives:

1) To study the effect of real and reactive powers on bus voltages.
2) Modeling of Thyristor Controlled Reactors (TCR).
3) Modeling of Thyristor Controlled Series Capacitors (TCSC).

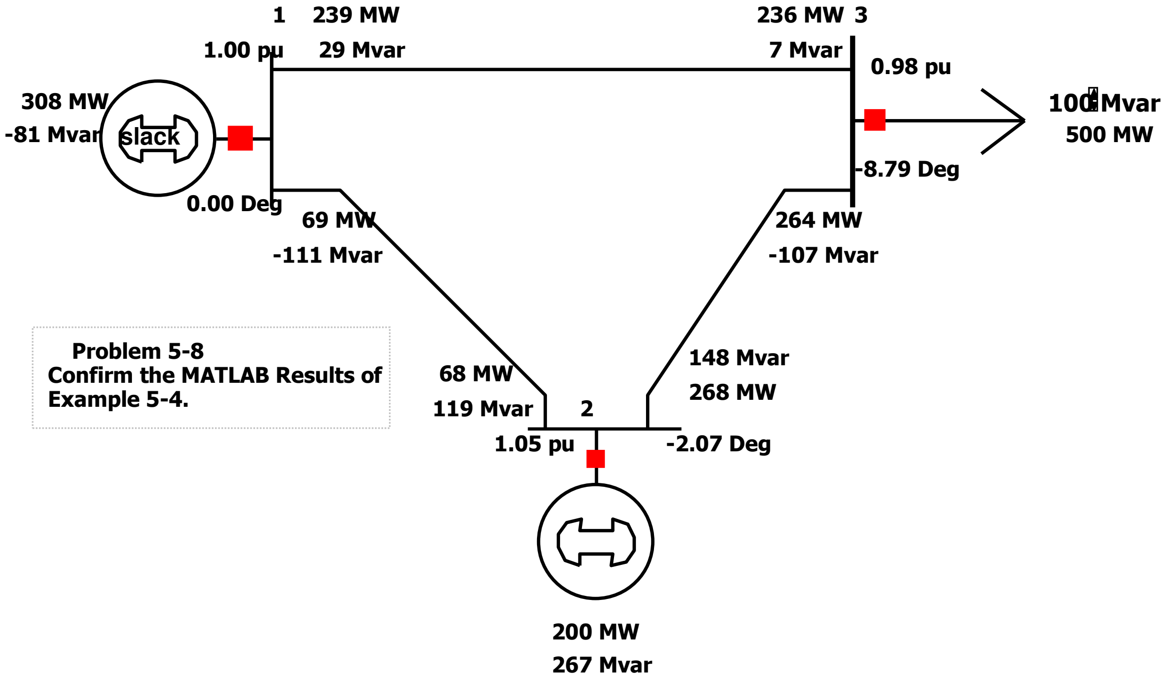
Laboratory Tasks and Report:

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.

2. Model a TCR as described by the *PSCAD/EMTDC* file **TCR.psc** (see video clip# 12). Obtain various waveforms and comment on them.

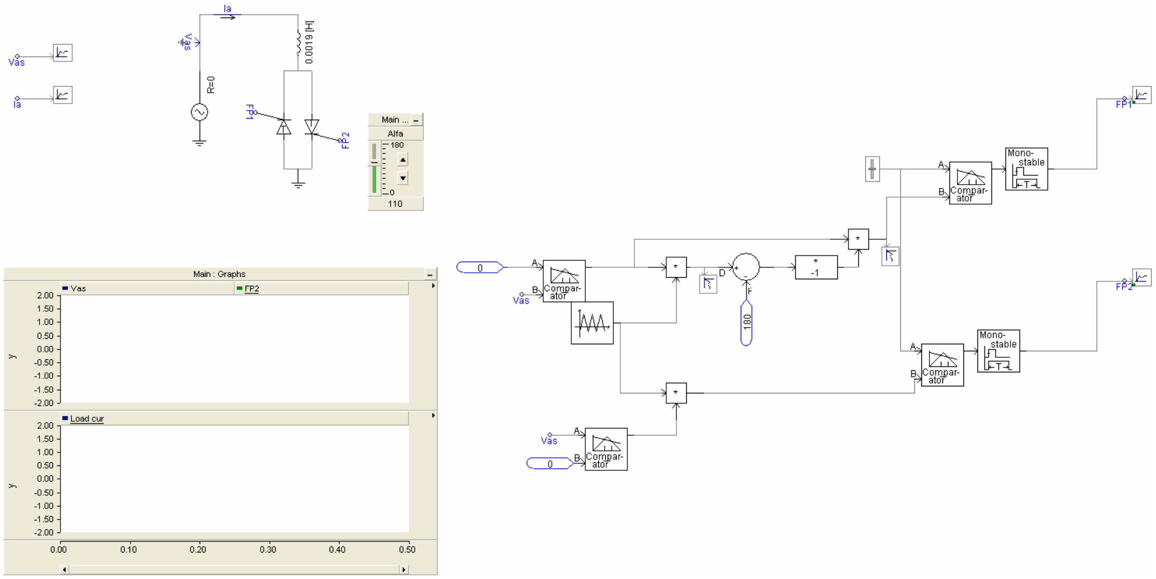
3. Model a TCSC as described by the *PSCAD/EMTDC* file **TCSC.psc** (see video clip# 13). Obtain various waveforms and comment on them.

VoltageRegulation.pwb

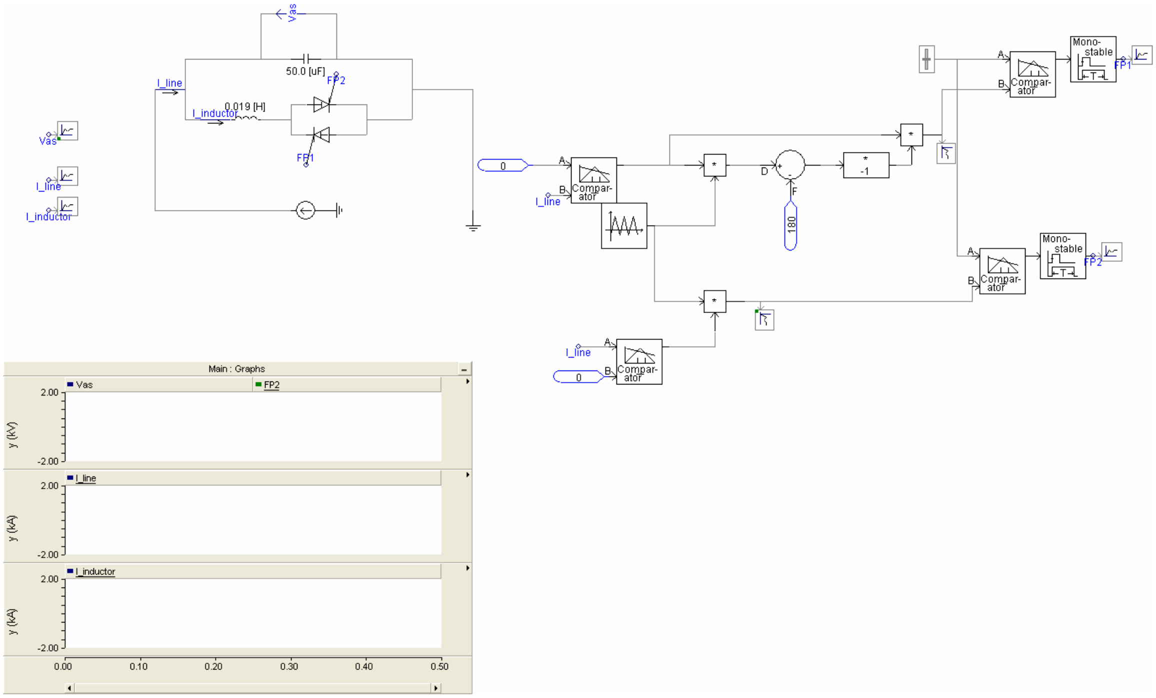


Problem 5-8
Confirm the MATLAB Results of Example 5-4.

TCR.psc



TCSC.psc



Laboratory Experiment 10

Transient Stability using MATLAB

Objectives: To calculate 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. See video clip# 14.

e. Annotate this file based on the material and equations of Chapter 11.
f. Execute this file get the plots of rotor angles of generators 1 and 2.
g. Double the fault time and observe the effect on stability.
h. Assume that a three-phase fault occurs on line between buses 2 and 3, one-third away from bus 2. The duration and the clearing time are the same as in the original case. Modify the program and calculate the transient stability.

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*(22/KV_LL)^2*(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*(22/KV_LL)^2*(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);
H2=H_Gen*(S_Gen2/MVA_Base);
X1=XdP1_PU+Xtr1_PU; X2=XdP2_PU+Xtr2_PU;

% Pre-Fault (Pre) steady state
EP1=V(1,1)+j*X1*Iinj(1,1);
EP2=V(2,1)+j*X2*Iinj(2,1);
EP1MAG=abs(EP1); EP2MAG=abs(EP2);
DT=0.0001;

% Pre-Fault Transient
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;
imax=1000
for i=2:imax
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1)*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2)*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;
    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;
    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

% During Fault Transient
kmax=5000
for k=1:kmax
    i=imax+k;
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1)*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2)*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;
    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;
    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

% Post Fault Transient
nmax=10000
for n=1:nmax
    i=imax+kmax+n;
    time1(i)=time1(i-1)+DT;
    w1(i)=w1(i-1)+(wsyn/(2*H1))*(Pm1-Pe1)*DT;
    w2(i)=w2(i-1)+(wsyn/(2*H2))*(Pm2-Pe2)*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;
    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;
    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

% plot(time1,w1,time1,w2)
plot(time1,DelDIFF1_DEG,time1,DelDIFF2_DEG)
%plot(time1,DelDIFF)
```

Laboratory Experiment 11

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:

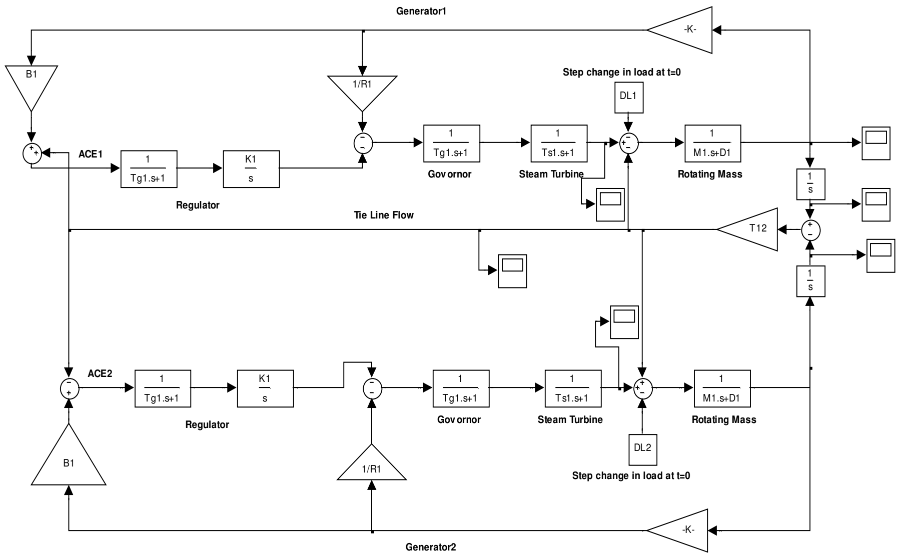
1. Study the dynamic interaction between two control areas using Simulink modeling. The MATLAB file for this is **AGC_Data.m**, which is located in this Folder. First launch MATLAB and open this file through it, and then execute it. Then double click on the *Simulink* file **AGC.mdl** located in this folder. Look at the various waveforms and comment on them. Adapted from Reference 6 in Chapter 12. See video clip# 15.

2. 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.

AGC_Data.m

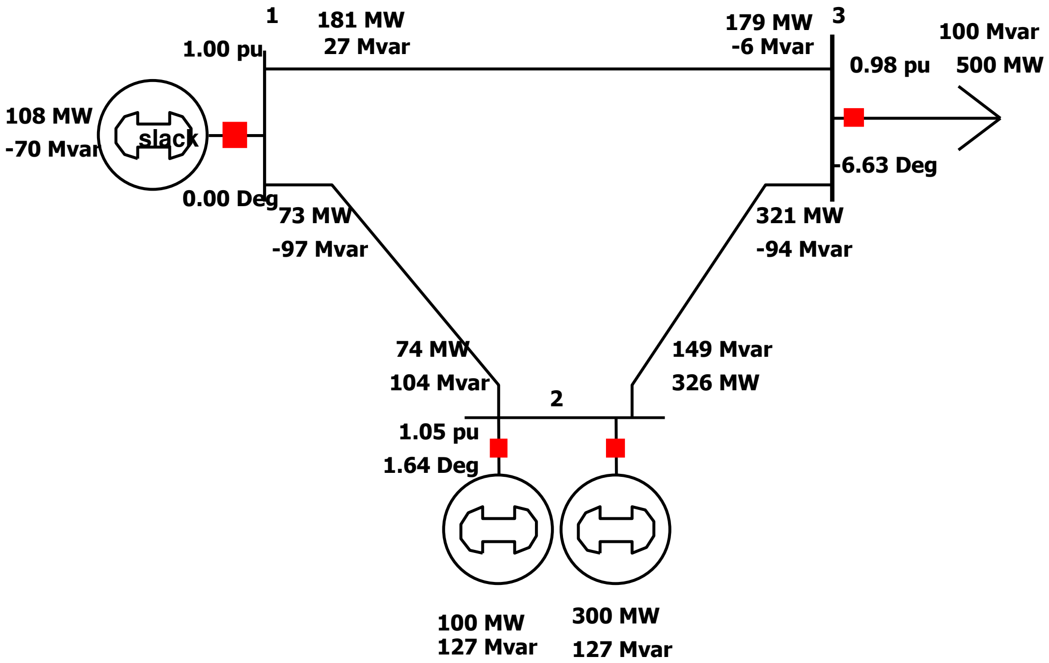
```
% Data for Example 12-3
H=5.3; % H=5.3 seconds
M1=(2*H)/(2*pi*60) % 2*pi*60 denotes the synchronous speed in rad/s
D1=0.75/(2*pi*60) % D1 is the damping coefficient
T12=0.1;
R1=0.167;
Tg1=.26;
Ts1=.26;
DL1=1; % 1 percent so the results are in per cent
DL2=0;
B1=1/R1+D1;
thetar=0;
K1=0.001*((2*pi*60));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

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:

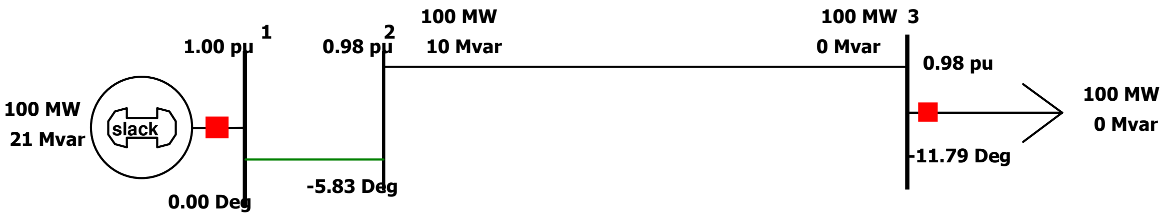
1. Simulate the fault in Example 13-2. The MATLAB file for this example is **SimpleSystemFault.m**, which is located in this Folder. First launch MATLAB and open this file through it, and then execute it. The *PowerWorld* file for this example is **SimpleSystemFault.pwb**; double click on it and compare results with that from the MATLAB simulation.

2. The PowerWorld file for this example is **ShortCircuitFault.pwb** (see video clips# 16 and 17), which is located in this Folder; double click on it and comment on results.

SimpleSystemFault.m

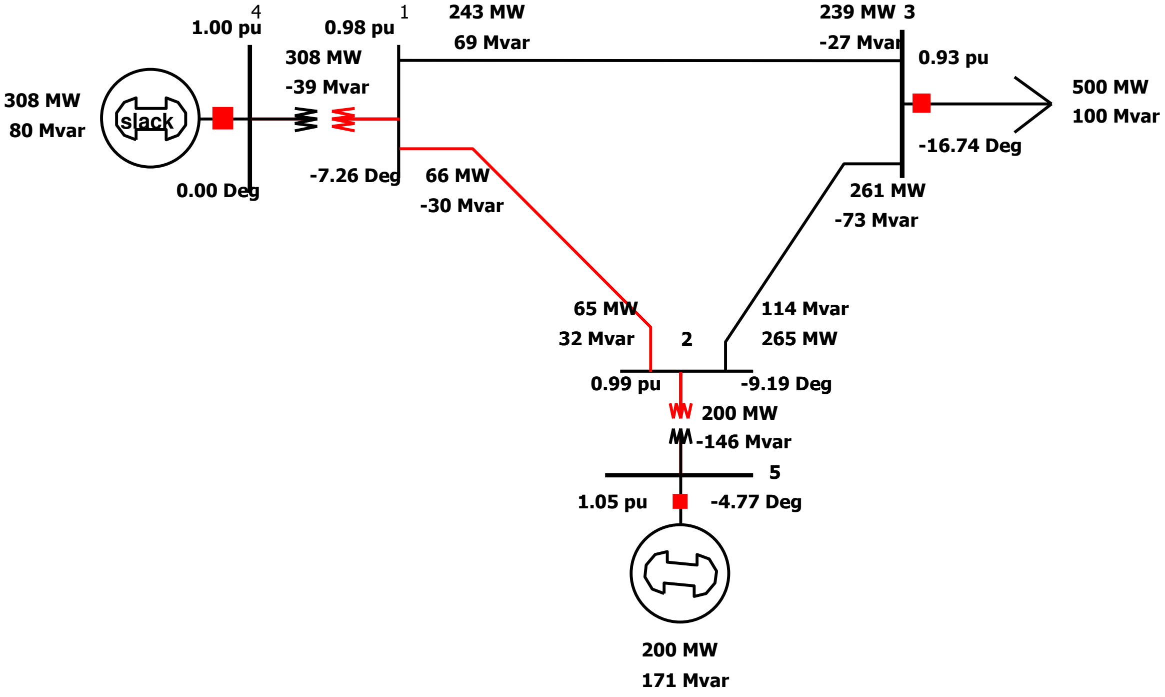
```
% Example 13-2; simple system
% Fault; 3-phase at bus 2
%Pre-Fault
V3_a1=0.98*exp(-j*11.79*pi/180);
I3_a1=(1/0.98)*exp(-j*11.79*pi/180);
Rload=0.98*0.98,
Ea=1+j*0.12*I3_a1,
[Th_Ea,Amp_Ea]=cart2pol(real(Ea),imag(Ea));
Th_Ea_deg=Th_Ea*180/pi,
Amp_Ea
Power=Ea*conj(I3_a1),
% 3-phase fault at bus 2
Ifault=Ea/(j*(0.12+0.1)),
[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



Example 13-2
Simple system Fault

ShortCircuitFault.pwb



Example 13-3

Laboratory Experiment 13

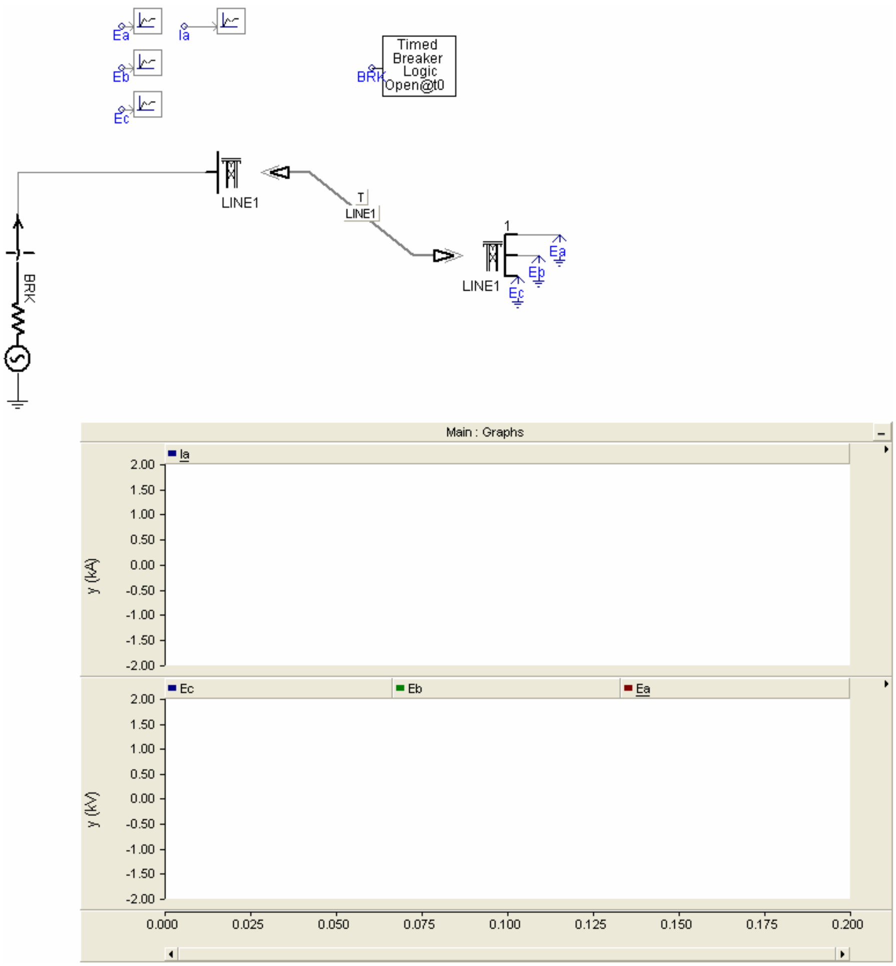
Switching Over-Voltages and Modeling of Surge Arresters using PSCAD/EMTDC

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

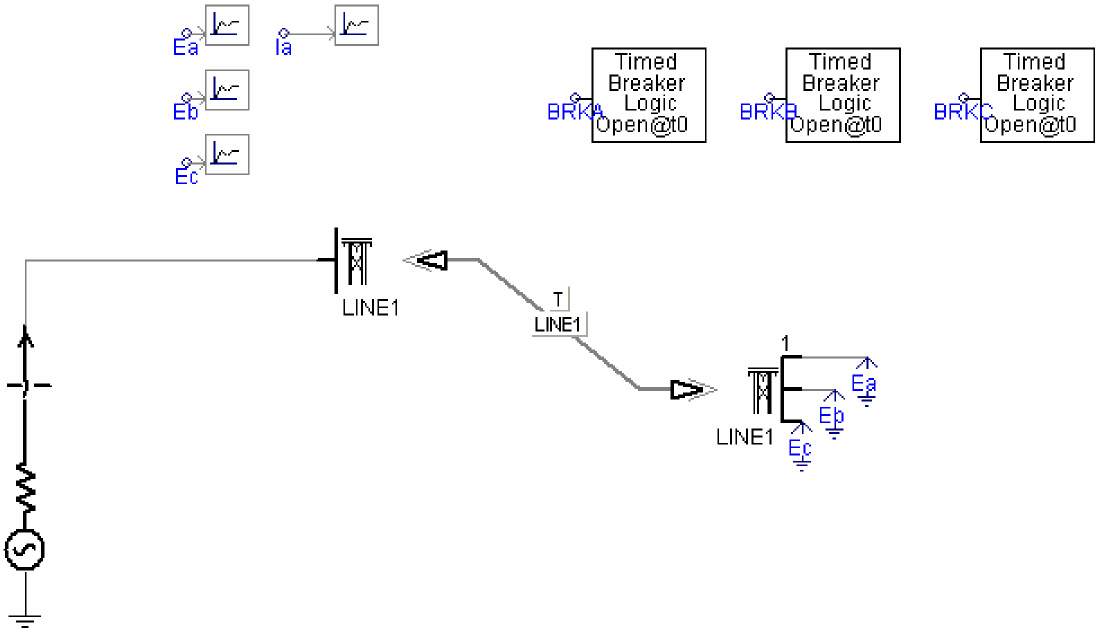
Laboratory Tasks and Report:

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)**.**

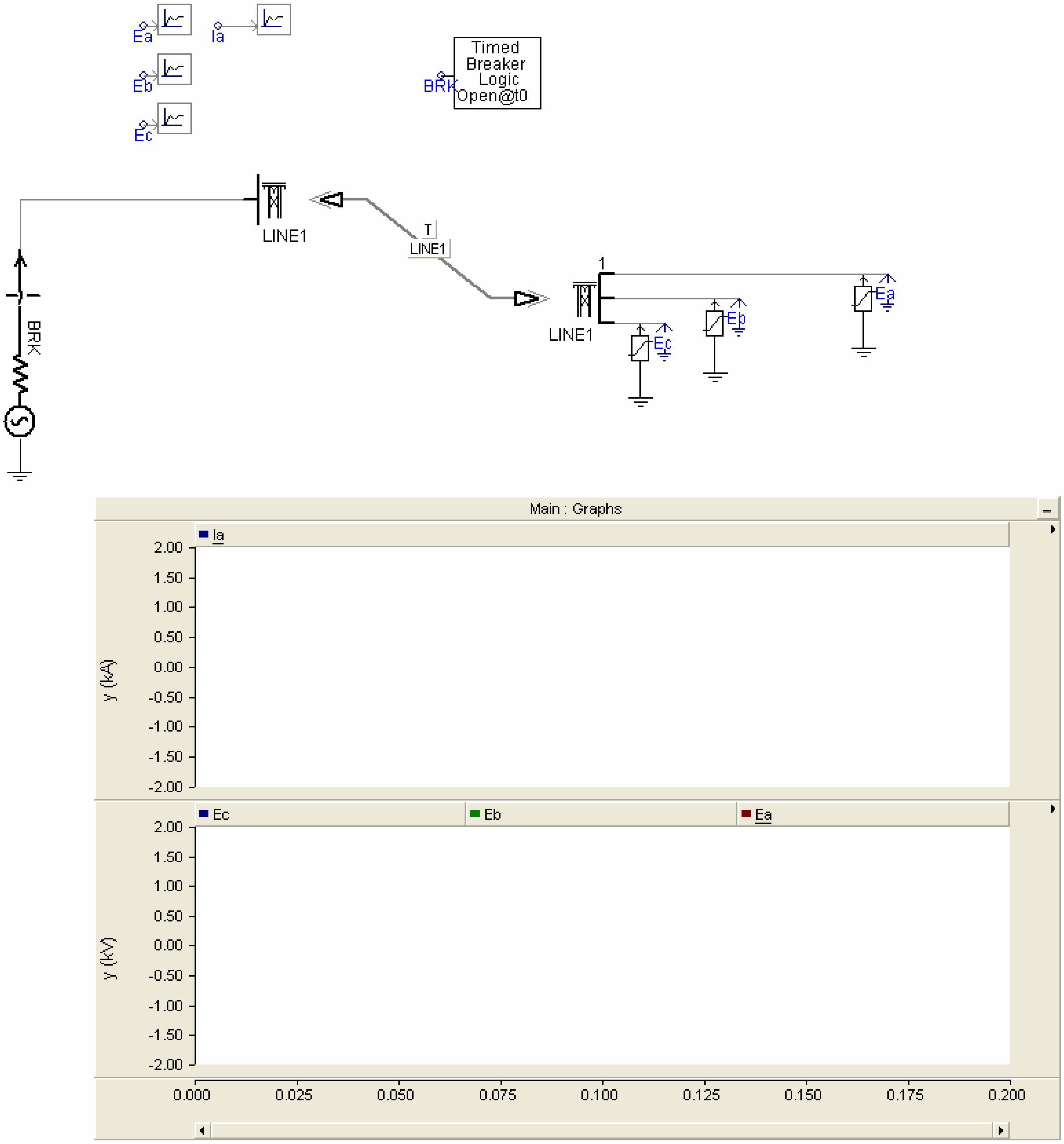
TL_Energization.psc



TL_Energization_Preinsertion.psc


Ea
Ia
Eb
Ec
Timed Breaker Logic Open@t0
BRKA
Timed Breaker Logic Open@t0
BRKB
Timed Breaker Logic Open@t0
BRKC
LINE1
T
LINE1
1
Ea
Eb
LINE1
Ec

TL_Energization_MOV.psc


Ea
Ia
Eb
Ec
Timed
Breaker
Logic
Open@t0
BRK
LINE1
T
LINE1
1
Ea
Eb
Ec
LINE1
BRK
Main : Graphs
Ia
y (kA)
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
Ec
Eb
Ea
y (kV)
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200