ECE 523 Exam 1 November 2019

EXAMINATION RULES

1. This is an open-book/open-note take-home exam.
2. I can e-mail a MathCAD file with the exam problems to you if you would like a copy.
3. Do your own work on this examination. You are on your honor. Therefore, you will neither give nor receive aid on this examination, except from the *course* instructor. If you violate this trust, you will receive the grade of zero for this examination.
4. Show all of your work! Make it neat. *No* partial credit will be given if I can not easily follow your work.
5. Cite the sources of information you use for your answers
6. You have 3 days to complete the exam from the time you receive it from your proctor.
7. ***Do NOT e-mail your completed exam to the course TA.***
8. Pease read and sign the following statement when you finish the exam:

I certify that I have neither given nor have I received any help on this examination, except from the course instructor.

SIGNED: Joe Stanley

PRINTED NAME: Joe Stanley

DATE: 11/5/2019

1. ________________/25 pts

2. ________________/35 pts

3. ________________/40 pts

Total: ________________/100 pts

Joe Stanley

ECE523 - EXAM1

In [1]:

```python
# Import Necessary Libraries
import numpy as np
import matplotlib.pyplot as plt
from tabulate import tabulate as tab
import electricpy as ep
from electricpy.constants import *
```

Problem 1 (25 pts) Consider a case where an equivalent load of 80 MW and 20 MVAR lagging is to be connected BUS 3 in the system below.

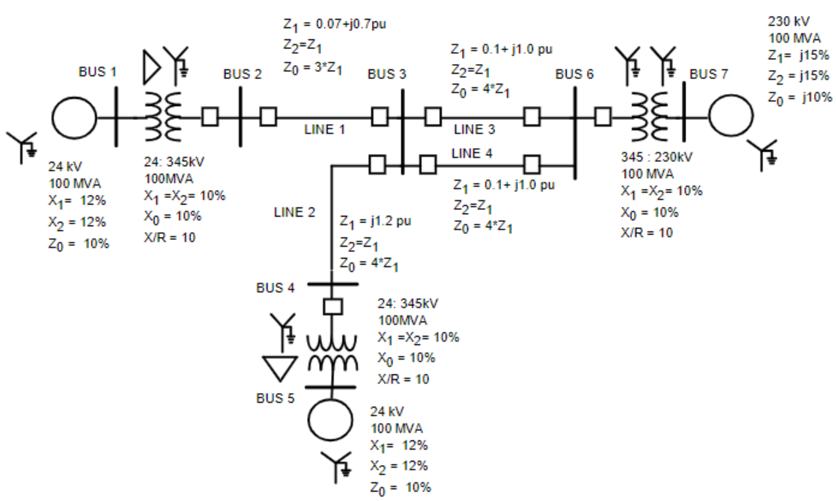
a). If the load is represented as a connection to a neighboring power system, and is treated as balanced positive sequence, how would model it for fault studies in the sequence domain, and how would you include the effect of the power flow in your fault studies? What other information do you need, if any?

b). How would the answer from part A change if you instead treated the load as a constant impedance load?

c). How would the answer from part A change if you instead treated the load as a constant current load?

d). How would the answer from part A change if you instead treated the load as a constant power load?

e). How would your answer from part A change if the load was 40% constant power, 40% constant current and 20% constant impedance?



Solution:

a) For loads described only by their positive sequence impedance, it is common to assume the negative sequence impedance as equal (and opposite where applicable in transformer phase shifts) to the positive sequence impedance, and it is also common to assume the zero sequence impedance as three times the positive sequence. That is to say:

$$Z_2 \approx Z_1 \qquad Z_0 \approx 3 \cdot Z_1$$

For fault analysis, short-circuit MVA or short-circuit current are also valuable pieces of information. Power flow information could be calculated at a steady-state value for unfaulted conditions using the positive sequence impedance to evaluate the power-flow results and treating the "load" as an infinite bus whose voltage magnitude and angle was fixed.

b) Treating the load as constant impedance load would ultimately remove the fixation of voltage magnitude and angle for the "load", however, it would largely leave the impedance relationships unchanged.

c) Treating the load as a constant current load may allow the assumption that the load's impedance will not affect the fault analysis, and that the fault studies may be conducted with impedance matrices that do not include the "load". The "load" current may not be neglected, however. Instead, the load current (being fixed) would be added to the fault current and any other relational power flow currents in much the same way that power-flow currents are treated in textbook-fault studies.

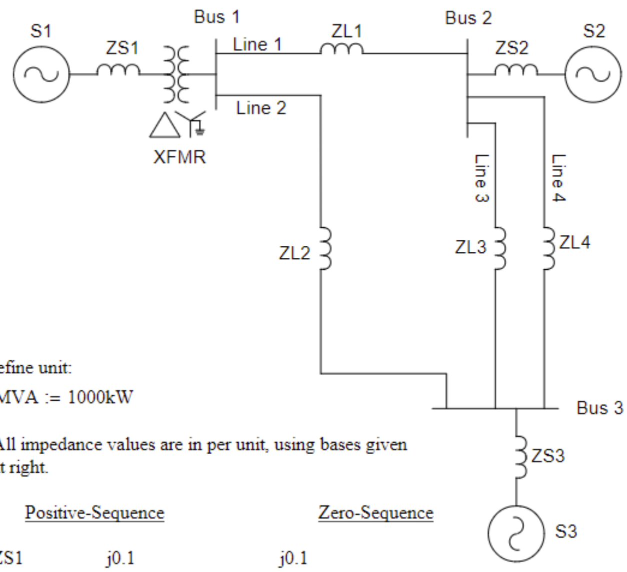
d) To fully account for a fixed power (or constant power) load, the solution method of solution should incorporate sequence component fault analysis prior to a power-flow calculation that includes the fictitious fault bus, accounting for the power consumption for the fault. This will allow for proper accounting of both the fault currents and the constant power currents, which will (inevitably) be affected by the fault's impact on bus voltages.

e) To account for a load that incorporates 40% constant power consumption, 40% constant current consumption, and 20% constant impedance would require decomposing the single load into three discrete "component" loads that represent each type of load characteristic. This would allow the composition of the various methods previously described to effectively represent the system.

Problem 2 (30 pts) The owner of the industrial co-gen facility connected to Bus 1 (point of common coupling or PCC) of the system below needs system equivalent impedance information in order to set their local protection equipment. You can use a fault program if you wish to do so.

I. Calculate the minimum and maximum short circuit MVA (both 3 phase and single line to ground fault cases -- hint: MVAsc was discussed in one of the early lectures) based on the following combinations:
   A. Everything in the system on-line.
   B. Any one of the lines out of service, but S2 and S3 both in service
   C. Either S2 or S3 out of service, but all lines in service.

II. Comment on whether your results willchange if the point of common coupling were on the low side of the transformer.



Define unit:
MVA := 1000kW

All impedance values are in per unit, using bases given at right.

	Positive-Sequence	Zero-Sequence
ZS1	j0.1	j0.1
XFMR	j0.2	j0.2
ZS2	j0.2	j0.35
ZS3	j0.2	j0.7
ZL1	0.01 + j0.15	0.06 + j0.5
ZL2	0.05 + j0.5	0.20 + j2.0
ZL3	0.02 + j0.2	0.07 + j0.7
ZL4	0.02 + j0.2	0.07 + j0.7

$S_B := 100MVA$

$V_{S1} := 24kV$

Transmission system: 230kV

Transformer: 24kV:230 kV, 100MVA

Solution:

I.

Description:	1ϕ MVA $_{sc}$	3ϕ MVA $_{sc}$
All On Line	60.24	774.2
Line 1 Fail	44.56	486.31
Line 2 Fail	55.79	666.1
Line 3 Fail	60.35	780.29
Line 4 Fail	60.35	780.29
Source 2 Off Line	47.57	549.32
Source 3 Off Line	55.62	637.48

II. The results will change if the point of common coupling was refferred to the low-voltage side of the transformer. This is beacuse the system would then have to be treated as a (minimum) 4 bus network and thus include the transformer impedance and phase shift in the thevenin impedance calculations.

Numerical Analysis and Solution:

In [5]:

```python
# Define Per-Unit Terms
Sbase = 100*M
VbaseT = 230*k
VbaseG = 24*k
ZbaseT = ep.zpu(Sbase,VLL=VbaseT)
ZbaseG = ep.zpu(Sbase,VLL=VbaseG)
IbaseT = ep.ipu(Sbase,VLL=VbaseT)
IbaseG = ep.ipu(Sbase,VLL=VbaseG)

# Define Sequence Impedances
ZS11 = 0.1J
ZS10 = 0.1J
XFM1 = 0.2J
XFM0 = 0.2J
ZS21 = 0.2J
ZS20 = 0.35J
ZS31 = 0.2J
ZS30 = 0.7J
ZL11 = 0.01+0.15J
ZL10 = 0.06+0.5J
ZL21 = 0.05+0.5J
ZL20 = 0.20+2.0J
ZL31 = 0.02+0.2J
ZL30 = 0.07+0.7J
ZL41 = 0.02+0.2J
ZL40 = 0.07+0.7J

# Define Function to Provide Impedance Matrices
def zmatrix(ZL1=True,ZL2=True,ZL3=True,ZL4=True,S2=True,S3=True,verbose=False):
    # Define Pos/Neg Seq Y-Bus Matrix
    y12 = np.array([
        [1/(ZS11+XFM1)+int(ZL1)/ZL11+int(ZL2)/ZL21, -int(ZL1)/ZL11, -int(ZL2)/ZL21],
        [-int(ZL1)/ZL11, int(ZL1)/ZL11+1/(int(ZL3)*ZL31+int(ZL4)*ZL41)+int(S2)/ZS21, -1/(int(ZL3)*ZL31+int(ZL4)*ZL41)],
        [-int(ZL2)/ZL21, -1/(int(ZL3)*ZL31+int(ZL4)*ZL41), int(ZL2)/ZL21+1/(int(ZL3)*ZL31+int(ZL4)*ZL41)+int(S3)/ZS31],
    ])
    # Define Zero Seq Y-Bus Matrix
    y0 = np.array([
        [1/(ZS10+XFM0)+int(ZL1)/ZL10+int(ZL2)/ZL20, -int(ZL1)/ZL10, -int(ZL2)/ZL20],
        [-int(ZL1)/ZL10, int(ZL1)/ZL10+1/(int(ZL3)*ZL30+int(ZL4)*ZL40)+int(S2)/ZS20, -1/(int(ZL3)*ZL30+int(ZL4)*ZL40)],
        [-int(ZL2)/ZL20, -1/(int(ZL3)*ZL30+int(ZL4)*ZL40), int(ZL2)/ZL20+1/(int(ZL3)*ZL30+int(ZL4)*ZL40)+int(S3)/ZS30],
    ])
    # Calculate Z-Bus Matrices
    zbus1 = np.linalg.inv(y12)
    zbus2 = np.linalg.inv(y12)
    zbus0 = np.linalg.inv(y0)
    # Print if Necessary
    if verbose:
        print("\nPositive Sequence Z-Bus:\n",tab(np.asarray(np.around(zbus1,3),dtype=str),tablefmt="fancy_grid"),sep='')
        print("\nNegative Sequence Z-Bus:\n",tab(np.asarray(np.around(zbus2,3),dtype=str),tablefmt="fancy_grid"),sep='')
        print("\nZero Sequence Z-Bus:\n",tab(np.asarray(np.around(zbus0,3),dtype=str),tablefmt="fancy_grid"),sep='')
    # Return Results
    return(zbus1,zbus2,zbus0)

# Define BUSnn Formulator
def BUSn(tup,ind=0):
    z1, z2, z0 = tup
    z1n = z1[ind][ind]
    z2n = z2[ind][ind]
    z0n = z0[ind][ind]
    return([z0n,z1n,z2n])

# Demonstrate Validity
print("Z-Bus, All On-Line")
z1,z2,z0 = zmatrix(verbose=True)

# A) Evaluate MVAsc for All On-Line
a_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix()),Sbase=Sbase),2)
a_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix()),Sbase=Sbase),2)

# B) Evaluate MVAsc for Each One Line Failure
b1_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(ZL1=False)),Sbase=Sbase),2)
b1_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(ZL1=False)),Sbase=Sbase),2)
b2_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(ZL2=False)),Sbase=Sbase),2)
b2_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(ZL2=False)),Sbase=Sbase),2)
b3_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(ZL3=False)),Sbase=Sbase),2)
b3_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(ZL3=False)),Sbase=Sbase),2)
b4_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(ZL4=False)),Sbase=Sbase),2)
b4_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(ZL4=False)),Sbase=Sbase),2)

# C) Evaluate MVAsc for Each Source Off-Line
c1_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(S2=False)),Sbase=Sbase),2)
c1_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(S2=False)),Sbase=Sbase),2)
c2_3ph = round(ep.fault.phs3mvasc(1.0,BUSn(zmatrix(S3=False)),Sbase=Sbase),2)
c2_1ph = round(ep.fault.phs1mvasc(1.0,BUSn(zmatrix(S3=False)),Sbase=Sbase),2)
```

Z-Bus, All On-Line

Positive Sequence Z-Bus:

(0.003+0.129j)	(-0+0.069j)	(-0.002+0.045j)
(-0+0.069j)	(0.002+0.111j)	(-0.002+0.044j)
(-0.002+0.045j)	(-0.002+0.044j)	(0.003+0.126j)

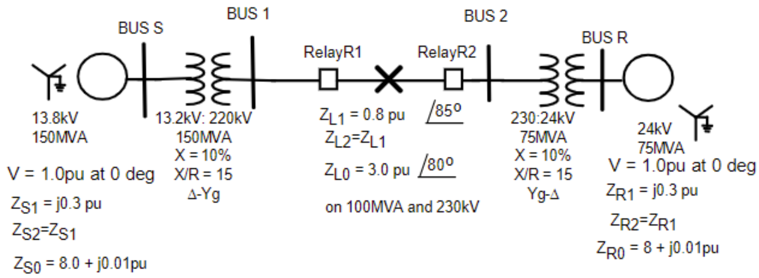
Negative Sequence Z-Bus:

(0.003+0.129j)	(-0+0.069j)	(-0.002+0.045j)
(-0+0.069j)	(0.002+0.111j)	(-0.002+0.044j)
(-0.002+0.045j)	(-0.002+0.044j)	(0.003+0.126j)

Zero Sequence Z-Bus:

(0.005+0.205j)	(-0.004+0.081j)	(-0.003+0.06j)
(-0.004+0.081j)	(0.006+0.218j)	(-0.003+0.074j)
(-0.003+0.06j)	(-0.003+0.074j)	(0.014+0.411j)

Problem 3. (40 pts) Given the network described below:



A. Calculate the sequence domain voltages and currents seen at BUS1 (RelayR1) for SLG, LL, and DLG faults with $R_f = 0$ at the following locations (you can use a fault program if you wish to do so).

(1) Bus 1
(2) 33% of the way down the line starting from BUS1 and going toward BUS2
(3) 67% of the way down the line starting from BUS 1.
(3) A fault at BUS2

B. Plot the sequence voltage magnitudes seen at BUS1 (RelayR1) versus fault location between BUS1 and BUS2 if $R_f = 0$ for the cases from part **A**.

C. For quantities measured at RelayR1, plot phase angle relationships between (1) the positive sequence voltage and positive sequence current, (2) the negative sequence voltage and negative sequence current, (3) the zero sequence voltage and zero sequence current, and (4) between the negative sequence current and zero sequence current for the fault cases from part **A.** versus fault location between BUS1 and BUS2.

D. Comment on your results

Solution:

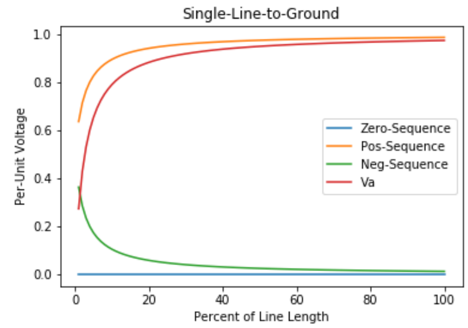
ASSUMPTIONS:

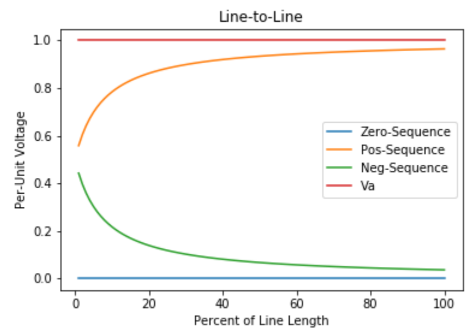
- The relays (R1/R2) are connected to a discrete section of line as close as possible to the bus, but not on it. Therefore, the assumption is made that if a fault occurs on the bus, that bus's relay sees the fault as behind itself.
- Voltage from the sources is regulated at the terminals, not at the internal voltage point.
- The Relays are "looking into" the line

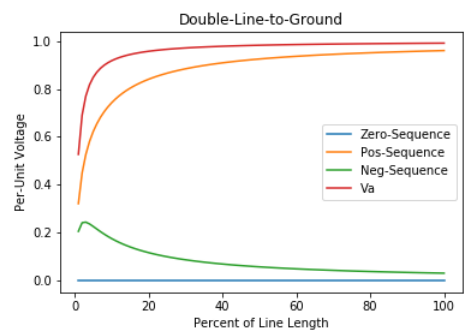
A. All calculations performed in following code section (Numerical Analysis and Solution).

Location	Single-Line-to-Ground (Per-Unit)	Line-to-Line (Per-Unit)	Double-Line-to-Ground (Per-Unit)
Bus 1	V: $\begin{bmatrix}0.617\angle 178.017^\circ\\0.808\angle -0.757^\circ\\0.192\angle -176.813^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.206\angle 98.017^\circ\\0.206\angle 98.017^\circ\\0.206\angle 98.017^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.5\angle 0.0^\circ\\0.5\angle 0.0^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.0\angle -180.0^\circ\\0.536\angle 94.83^\circ\\0.536\angle -85.17^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.433\angle -0.696^\circ\\0.433\angle -0.696^\circ\\0.433\angle -0.696^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.144\angle -80.696^\circ\\0.608\angle 95.361^\circ\\0.464\angle -85.865^\circ\end{bmatrix}$
33% From Bus 1	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.963\angle -0.159^\circ\\0.037\angle -175.884^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.61\angle -82.07^\circ\\0.61\angle -82.07^\circ\\0.61\angle -82.07^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.906\angle -0.1^\circ\\0.094\angle 0.963^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.0\angle 0.0^\circ\\1.538\angle -85.223^\circ\\1.538\angle 94.777^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.893\angle -0.182^\circ\\0.081\angle 0.227^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.434\angle 99.264^\circ\\1.754\angle -84.668^\circ\\1.322\angle 94.041^\circ\end{bmatrix}$
67% From Bus 1	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.981\angle -0.083^\circ\\0.019\angle -175.722^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.312\angle -81.908^\circ\\0.312\angle -81.908^\circ\\0.312\angle -81.908^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.949\angle -0.057^\circ\\0.051\angle 1.065^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.0\angle 0.0^\circ\\0.837\angle -85.121^\circ\\0.837\angle 94.879^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.942\angle -0.097^\circ\\0.045\angle 0.403^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.217\angle 99.338^\circ\\0.945\angle -84.611^\circ\\0.729\angle 94.217^\circ\end{bmatrix}$
Bus 2	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.987\angle -0.057^\circ\\0.013\angle -175.668^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.212\angle -81.854^\circ\\0.212\angle -81.854^\circ\\0.212\angle -81.854^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.965\angle -0.041^\circ\\0.035\angle 1.102^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.0\angle 0.0^\circ\\0.581\angle -85.084^\circ\\0.581\angle 94.916^\circ\end{bmatrix}$	V: $\begin{bmatrix}0.0\angle -116.565^\circ\\0.96\angle -0.066^\circ\\0.031\angle 0.464^\circ\end{bmatrix}$, I: $\begin{bmatrix}0.146\angle 99.362^\circ\\0.653\angle -84.588^\circ\\0.508\angle 94.278^\circ\end{bmatrix}$

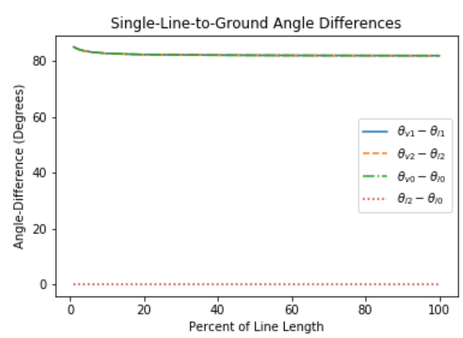
B.

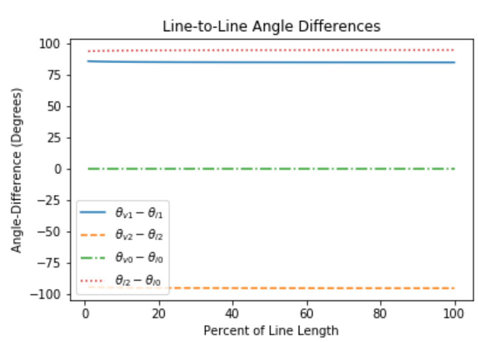


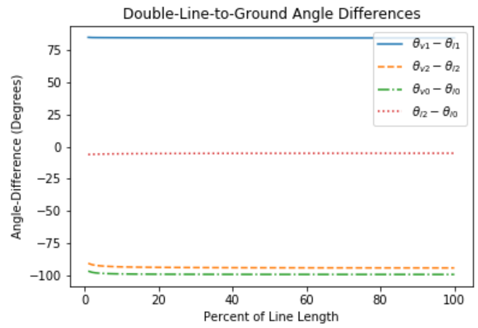




C. All calculations performed in following code section (Numerical Analysis and Solution).







D. These results exemplify what one would expect for the various types of faults on a simple single line system. It is encouraging to see the relationships between the voltage magnitudes as shown in section B, and it is also interesting to see the "locked" variance between angles as shown in section C. It is interesting to see that these angle relations seem to remain largely unchanged over the entire length of the line. This implies that the angle can be used as a factor of protection logic as it is not affected by the fault location. Then, of course, the supervisory control decision may be made by magnitude detection.

Numerical Analysis and Solution:

In [4]:

```
# Define Per-Unit Bases
Sbase = 100*M
VbaseT = 230*k
VbaseS = (VbaseT*13.2/220)*k
VbaseR = 24*k
ZbaseT = ep.zpu(Sbase,VLL=VbaseT)
ZbaseS = ep.zpu(Sbase,VLL=VbaseS)
ZbaseR = ep.zpu(Sbase,VLL=VbaseR)

# Define System Parameters
ZL1 = ep.phasor(0.8,85)
ZL0 = ep.phasor(3,80)
ZT1 = ep.puchgbase(ep.rxrecompose(0.1,15),ep.zpu(150*M,220*k),ZbaseT)
ZT2 = ep.puchgbase(ep.rxrecompose(0.1,15),ep.zpu(75*M,230*k),ZbaseT)
ZS1 = ep.puchgbase(0.3j,ep.zpu(150*M,13.8*k),ZbaseS)
ZS0 = ep.puchgbase(8.0+0.01j,ep.zpu(150*M,13.8*k),ZbaseS)
ZR1 = ep.puchgbase(0.3j,ep.zpu(75*M,24*k),ZbaseR)
ZR0 = ep.puchgbase(8.0+0.01j,ep.zpu(75*M,24*k),ZbaseR)

# A.1)
# General Terms
Vth = 1
Zth = np.array([ZL0,ZL1+ZT2,ZL1+ZT2]) # Sequence Impedances
Zbus = np.array([ZL0,ZL1+ZT2,ZL1+ZT2])
# Single Line to Ground
# Faulted Terms
i = ep.fault.phs1g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zth * i
# Printable Terms
i1_slg = ep.clatex(-i)
v1_slg = ep.clatex(v)
# Line to Line
# Faulted Terms
i = ep.fault.phs2(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zth * i
# Printable Terms
i1_ll = ep.clatex(-i)
v1_ll = ep.clatex(v)
# Double Line to Ground
# Faulted Terms
i = ep.fault.phs2g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zth * i
# Printable Terms
i1_dlg = ep.clatex(-i)
v1_dlg = ep.clatex(v)

# A.2)
# General Terms
Vth = 1
Zbus = np.array([0,ZT1,ZT1])
z0 = 0.33*ZL0
z1 = (0.33*ZL1+ZT1)
z2 = z1
Zth = np.array([z0,z1,z2]) # Sequence Impedances
# Single Line to Ground
# Faulted Terms
i = ep.fault.phs1g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i2_slg = ep.clatex(i)
v2_slg = ep.clatex(v)
# Line to Line
# Faulted Terms
i = ep.fault.phs2(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i2_ll = ep.clatex(i)
v2_ll = ep.clatex(v)
# Double Line to Ground
# Faulted Terms
i = ep.fault.phs2g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i2_dlg = ep.clatex(i)
v2_dlg = ep.clatex(v)

# A.3)
# General Terms
z0 = 0.67*ZL0
z1 = (0.67*ZL1+ZT1)
z2 = z1
Zth = np.array([z0,z1,z2]) # Sequence Impedances
# Single Line to Ground
# Faulted Terms
i = ep.fault.phs1g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i3_slg = ep.clatex(i)
v3_slg = ep.clatex(v)
# Line to Line
# Faulted Terms
i = ep.fault.phs2(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i3_ll = ep.clatex(i)
```

```
v3_ll = ep.clatex(v)
# Double Line to Ground
# Faulted Terms
i = ep.fault.phs2g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i3_dlg = ep.clatex(i)
v3_dlg = ep.clatex(v)

# A.4)
# General Terms
z0 = ZL0
z1 = ZL1+ZT1
z2 = z1
Zth = np.array([z0,z1,z2]) # Sequence Impedances
# Single Line to Ground
# Faulted Terms
i = ep.fault.phs1g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i4_slg = ep.clatex(i)
v4_slg = ep.clatex(v)
# Line to Line
# Faulted Terms
i = ep.fault.phs2(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i4_ll = ep.clatex(i)
v4_ll = ep.clatex(v)
# Double Line to Ground
# Faulted Terms
i = ep.fault.phs2g(Vth,Zth)
v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
# Printable Terms
i4_dlg = ep.clatex(i)
v4_dlg = ep.clatex(v)

# B)
LineLen = np.arange(1,101,1)
# Define Simple Plotter Function
def VoverLinePlot(faultfunc,title,plot=True):
    # Define Initial Arrays
    V0 = np.array([])
    V1 = np.array([])
    V2 = np.array([])
    Va = np.array([])
    wvi1 = np.array([])
    wvi2 = np.array([])
    wvi0 = np.array([])
    wi20 = np.array([])
    for L in LineLen:
        z0 = (L/100)*ZL0
        z1 = (L/100)*ZL1+ZT1
        z2 = z1
        Zth = np.array([z0,z1,z2]) # Sequence Impedances
        i = faultfunc(Vth,Zth)
        v = ep.abc_to_seq([Vth,Vth*ep.phs(-120),Vth*ep.phs(-240)]) - Zbus * i
        #ep.cprint(ep.seq_to_phs(v))
        va,vb,vc = np.abs(ep.seq_to_phs(v))
        # Calculate Voltage Magnitudes
        Va = np.append(Va,va)
        v0, v1, v2 = np.abs(v)
        V0 = np.append(V0,v0)
        V1 = np.append(V1,v1)
        V2 = np.append(V2,v2)
        # Calculate Angle Differences
        wvi1 = np.append(wvi1,np.angle(v1,True)-np.angle(i[1],True))
        wvi2 = np.append(wvi2,np.angle(v2,True)-np.angle(i[2],True))
        wvi0 = np.append(wvi0,np.angle(v0,True)-np.angle(i[0],True))
        wi20 = np.append(wi20,np.angle(i[2],True)-np.angle(i[0],True))
    # Plot Voltage Magnitudes
    plt.figure()
    plt.plot(LineLen,V0,label='Zero-Sequence')
    plt.plot(LineLen,V1,label='Pos-Sequence')
    plt.plot(LineLen,V2,label='Neg-Sequence')
    plt.plot(LineLen,Va,label="Va")
    plt.title(title)
    plt.xlabel("Percent of Line Length")
    plt.ylabel("Per-Unit Voltage")
    plt.legend()
    plt.savefig(title+".png")
    if plot==True:
        plt.show()
    else:
        plt.close()
    # Plot Angle Differences
    plt.figure()
    plt.plot(LineLen,wvi1,label="$\\theta_{v1}-\\theta_{i1}$",linestyle='-')
    plt.plot(LineLen,wvi2,label="$\\theta_{v2}-\\theta_{i2}$",linestyle='--')
    plt.plot(LineLen,wvi0,label="$\\theta_{v0}-\\theta_{i0}$",linestyle='-.')
    plt.plot(LineLen,wi20,label="$\\theta_{i2}-\\theta_{i0}$",linestyle=':')
    plt.title(title+" Angle Differences")
    plt.xlabel("Percent of Line Length")
    plt.ylabel("Angle-Difference (Degrees)")
    plt.legend()
    plt.savefig(title+" Angle Differences.png")
```

```python
    if plot==True:
        plt.show()
    else:
        plt.close()

# Generate Plots for Each Fault Type
VoverLinePlot(ep.fault.phs1g,"Single-Line-to-Ground",False)
VoverLinePlot(ep.fault.phs2g,"Double-Line-to-Ground",plot=False)
VoverLinePlot(ep.fault.phs2,"Line-to-Line",plot=False)
```

In [ ]:

```python
```

In [ ]:

```python
```

Problem 1 (25 pts) Consider a case where an equivalent load of 80 MW and 20 MVAR lagging is to be connected BUS 3 in the system below.

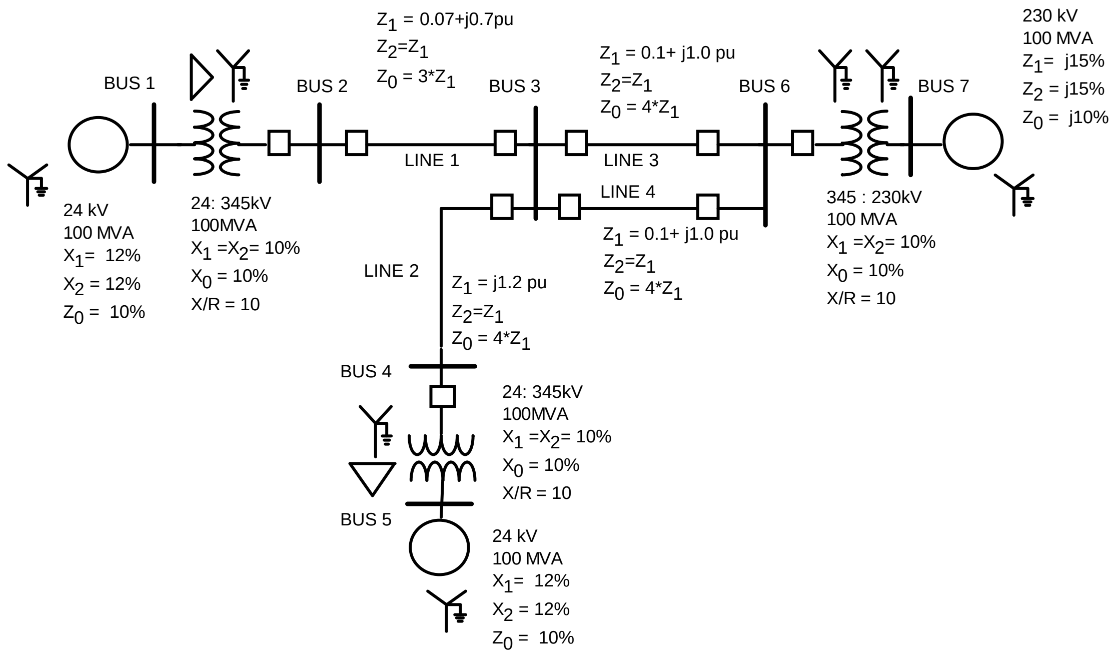
a). If the load is represented as a connection to a neighboring power system, and is treated as balanced positive sequence, how would model it for fault studies in the sequence domain, and how would you include the effect of the power flow in your fault studies? What other information do you need, if any?

b). How would the answer from part A change if you instead treated the load as a constant impedance load?

c). How would the answer from part A change if you instead treated the load as a constant current load?

d). How would the answer from part A change if you instead treated the load as a constant power load?

e). How would your answer from part A change if the load was 40% constant power, 40% constant current and 20% constant impedance?



Problem 2 (30 pts) The owner of the industrial co-gen facility connected to Bus 1 (point of common coupling or PCC) of the system below needs system equivalent impedance information in order to set their local protection equipment. You can use a fault program if you wish to do so.

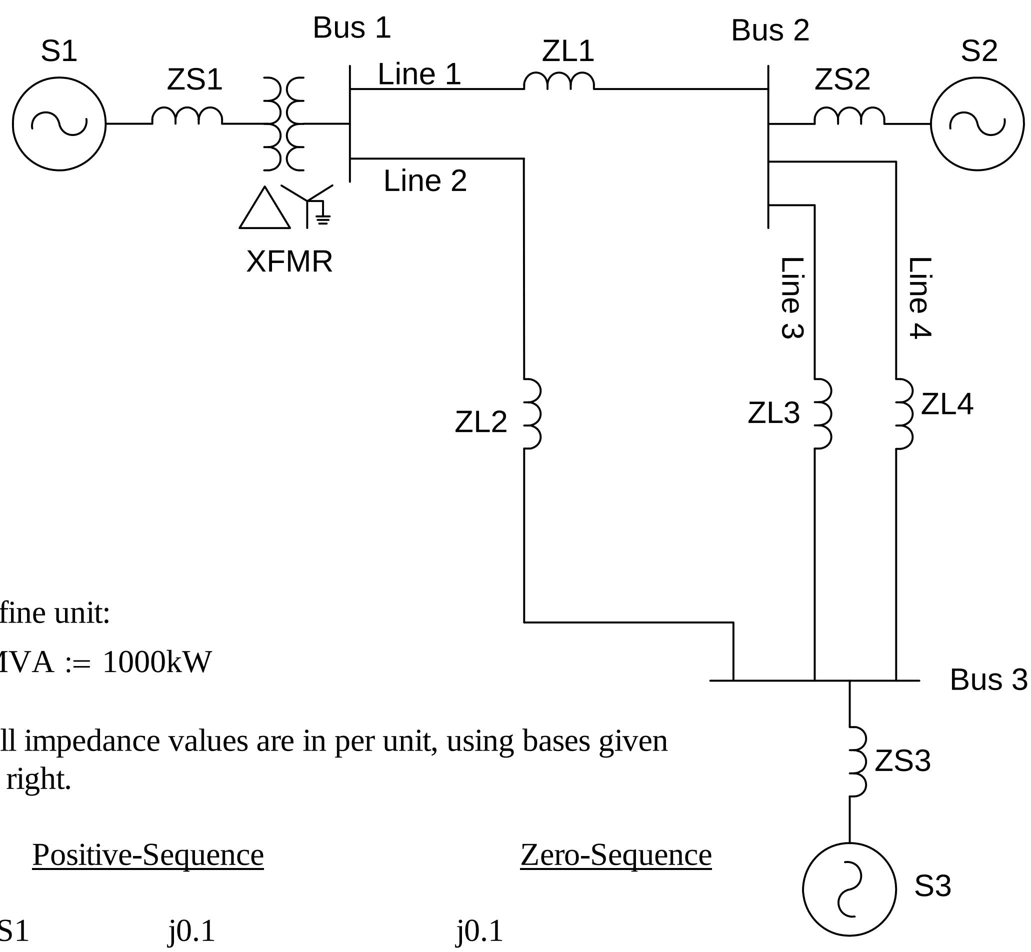
I. Calculate the minimum and maximum short circut MVA (both 3 phase and single line to ground fault cases -- hint: MVAsc was discussed in one of the early lectures) based on the following combinations:

A. Everything in the system on-line.

B. Any one of the lines out of service, but S2 and S3 both in service

C. Either S2 or S3 out of service, but all lines in service.

II. Comment on whether your results willchange if the point of common coupling were on the low side of the transformer.



Define unit:

MVA := 1000kW

All impedance values are in per unit, using bases given at right.

	Positive-Sequence	Zero-Sequence
ZS1	j0.1	j0.1
XFMR	j0.2	j0.2
ZS2	j0.2	j0.35
ZS3	j0.2	j0.7
ZL1	0.01 + j0.15	0.06 + j0.5
ZL2	0.05 + j0.5	0.20 + j2.0
ZL3	0.02 + j0.2	0.07 + j0.7
ZL4	0.02 + j0.2	0.07 + j0.7

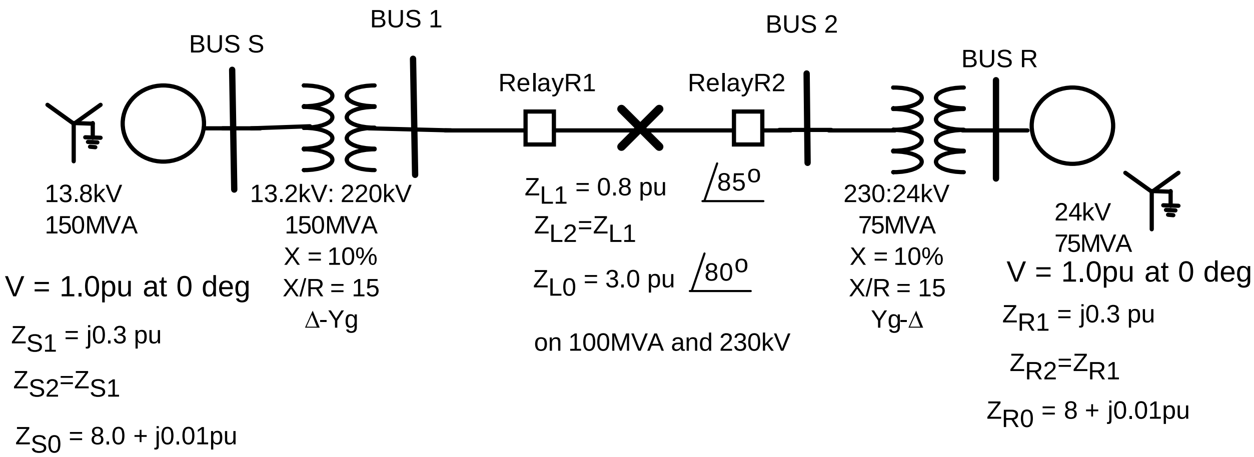
S_B := 100MVA

V_{S1} := 24kV

Transmission system: 230kV

Transformer: 24kV:230 kV, 100MVA

Problem 3. (40 pts) Given the network described below:



A. Calculate the sequence domain voltages and currents seen at BUS1 (RelayR1) for SLG, LL, and DLG faults with R_f = 0 at the following locations (you can use a fault program if you wish to do so).

(1) Bus 1
(2) 33% of the way down the line starting from BUS1 and going toward BUS2
(3) 67% of the way down the line starting from BUS 1.
(3) A fault at BUS2

B. Plot the sequence voltage magnitudes seen at BUS1 (RelayR1) versus fault location between BUS1 and BUS2 if R_f = 0 for the cases from part **A**.

C. For quantities measured at RelayR1, plot phase angle relationships between (1) the positive sequence voltage and positive sequence current, (2) the negative sequence voltage and negative sequence current, (3) the zero sequence voltage and zero sequence current, and (4) between the negative sequence current and zero sequence current for the fault cases from part **A.** versus fault location between BUS1 and BUS2.

D. Comment on your results