Joe Stanley

ECE524 - HWK4

In [1]:

```
# Import Libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve
import eepower as eep
from eepower import u,m,k,M
```

In [86]:

```python
# Define Function to Calculate TRV
def pktransrecvolt(C,L,R=0,VLL=None,VLN=None,freq=60):
    """
    pktransrecvolt Function

    Peak Transient Recovery Voltage calculation
    function, evaluates the peak transient
    recovery voltage (restriking voltage) and
    the Rate-of-Rise-Recovery Voltage.

    Parameters
    ----------
    C:          float
                Capacitance Value in Farads.
    L:          float
                Inductance in Henries.
    R:          float, optional
                The resistance of the system used for
                calculation, default=0.
    VLL:        float, exclusive
                Line-to-Line voltage, exclusive
                optional argument.
    VLN:        float, exclusive
                Line-to-Neutral voltage, exclusive
                optional argument.
    freq:       float, optional
                System frequency in Hz.

    Returns
    -------
    Vcpk:       float
                Peak Transient Recovery Voltage in volts.
    RRRV:       float
                The RRRV (Rate-of-Rise-Recovery Voltage)
                calculated given the parameters in volts
                per second.
    """
    # Evaluate alpha, omega-n, and fn
    alpha = R/(2*L)
    wn = 1/np.sqrt(L*C) - alpha
    fn = wn/(2*np.pi)
    # Evaluate Vm
    if VLL!=None:
        Vm = np.sqrt(2/3)*VLL
    elif VLN!=None:
        Vm = np.sqrt(2)*VLN
    else:
        raise ValueError("One voltage must be specified.")
    # Evaluate Vcpk (worst case)
    Vcpk = wn**2/(wn**2-2*np.pi*freq)*Vm*2
    # Evaluate RRRV
    RRRV = 2*Vm*fn/0.5
    return(Vcpk,RRRV)

# Define TRV Reduction Resistor Function
def trvresistor(C,L,reduction):
```

```
    """
    trvresistor Function

    Function to find the resistor value that
    will reduce the TRV by a specified
    percentage.

    Parameters
    ----------
    C:          float
                Capacitance Value in Farads.
    L:          float
                Inductance in Henries.
    reduction:  float
                The percentage that the TRV
                should be reduced by.

    Returns
    -------
    Rd:         float
                Damping resistor value, in ohms.
    wd:         float
                Omega-d
    tpk:        float
                Time of peak voltage.
    """
    # Evaluate omega-n
    wn = 1/np.sqrt(L*C)
    # Generate Constant Factor
    fctr = (1-reduction)*2 - 1
    # Define Function Set
    def equations(data):
        Rd, wd, tpk = data
        X = np.sqrt(wn**2-(1/(2*Rd*C))**2) - wd
        Y = np.exp(-tpk/(2*Rd*C))-fctr
        Z = wd*tpk - np.pi
        return(X,Y,Z)
    Rd, wd, tpk = fsolve(equations, (500,260*k,10*u))
    return(Rd, wd, tpk)

# Define Natural Frequency/Resonant Frequency Calculator
def natfreq(C,L,Hz=True):
    """
    natfreq Function

    Evaluates the natural frequency (resonant frequency)
    of a circuit given the circuit's C and L values. Defaults
    to returning values in Hz, but may also return in rad/sec.

    Parameters
    ----------
    C:          float
                Capacitance Value in Farads.
    L:          float
                Inductance in Henries.
    Hz:         bool, optional
                Control argument to set return value in either
```

```
                Hz or rad/sec; default=True.

    Returns
    -------
    freq:       float
                Natural (Resonant) frequency, will be in Hz if
                argument *Hz* is set True (default), or rad/sec
                if argument is set False.
    """
    # Evaluate Natural Frequency in rad/sec
    freq = 1/np.sqrt(L*C)
    # Convert to Hz as requested
    if Hz:
        freq = freq / (2*np.pi)
    return(freq)
```

Problem 1:

Problem 8.6 in the text book. Assume that it is a 24 kV system. Simulate the circuit and find the peak TRV and RRRV. Also simulate a case where a TRV reduction resistance of 500 ohms is utilized and is in the circuit for 1 cycle and provide the peak TRV and new RRRV. Measure or calculte the energy dissipated in the resistor in your transients program. Extra credit: solve analytically as well.

In [91]:

```
# Define Givens
VLL = 24*k
L = 1.5*m
C = 0.005*u
Ifault = 20*k

# Calculate TRV and RRRV
TRVpeak, RRRVpeak = pktransrecvolt(C,L,VLL=VLL)
print("Peak TRV:",TRVpeak/k,"kV")
print("Peak RRRV:",RRRVpeak/k*u,"kV/uS")

# Calculate TRV and RRRV
TRVpeak, RRRVpeak = pktransrecvolt(C,L,R=500,VLL=VLL)
print("Peak TRV:",TRVpeak/k,"kV")
print("Peak RRRV:",RRRVpeak/k*u,"kV/uS")
```

```
Peak TRV: 39.1918359953 kV
Peak RRRV: 4.55528027787 kV/uS
Peak TRV: 39.1918362596 kV
Peak RRRV: 2.47608880791 kV/uS
```

Problem 2:

A 10 MVAR capacitor bank is connected on the secondary of a 25 MVA, 230:13.2kV delta-wye grounded transformer with a per unit reactance of 6%. The short circuit rating of the connection back to the 230kV source is 18.5kA. Do the following:

(a) Calculate the resonant frequency for this circuit if the capacitor bank is connected in delta (assume balanced 3 phase connection for now). Verify with a simulation model (it is easiest to do the simulation with dc sources that represent instantaneous values of a 3 phase set).

(b) Repeat part (a) if it connected in wye. Verify with a simulation model.

(c) Assuming that the capacitor bank is wye connected and grounded: suppose the system is operating in steady-state. Estimate the voltages trapped on each of the capacitors if the breakers connecting the capacitor to the bus open. Verify with a simulation model.

(d) Repeat part (c) if the capacitor bank is connected in delta. Verify with a simulation model.

(e) Repeat part (c) if the capacitor bank is Y connected with a neutral to ground capacitance of 100 nF. Again, verify with simulations.

In [51]:

```python
# Define Givens
Q_cbank = 10*M #VAR
Vls = 13.2*k # Line-to-Line
xfm_rat = 25*M #VA
xfm_PS = 230/13.2
xfm_SP = 1/xfm_PS
xfm_x = 6/100 #PU
SC_rat = 18.5*k

# a) Resonant Frequency Delta-Connected
# Evaluate Inductance
Lx_xfm_d = xfm_x * eep.zpu(xfm_rat,VLL=eep.phaseline(VLL=Vls))
L_xfm_d = eep.reactance(Lx_xfm_d)
print("Inductance:",L_xfm_d/m,"mH")
# Evaluate Capacitance
C_bnk_d = eep.farads(Q_cbank,Vls)
print("Capacitance:",C_bnk_d/u,"uF")
# Evaluate Resonant Frequency
res_freq = natfreq(C_bnk_d,L_xfm_d)
print("a) Resonant Frequency:",res_freq,"Hz\tT-cycle:",1/res_freq/m,"milisec

# b) Resonant Frequency Wye-Connected
# Evaluate Inductance
Lx_xfm_y = xfm_x * eep.zpu(xfm_rat,VLL=eep.phaseline(VLL=Vls))
L_xfm_y = eep.reactance(Lx_xfm_y)
print("Inductance:",L_xfm/m,"mH")
# Evaluate Capacitance
C_bnk_y = eep.farads(Q_cbank,eep.phaseline(VLL=Vls))
print("Capacitance:",C_bnk_y/u,"uF")
# Evaluate Resonant Frequency
res_freq = natfreq(C_bnk_y,L_xfm_y)
print("b) Resonant Frequency:",res_freq,"Hz\tT-cycle:",1/res_freq/m,"milisec

# c) Estimate Trapped Voltages
TRV_y = pktransrecvolt(C_bnk_y,L_xfm_y,VLL=Vls)[0]
print("c) Trapped Voltage in Cap (TRV):",TRV_y/k,"kV")

# d) Estimate Trapped Voltages
TRV_d = pktransrecvolt(C_bnk_d,L_xfm_d,VLL=Vls)[0]
print("d) Trapped Voltage in Cap (TRV):",TRV_d/k,"kV")

# e) Estimate Trapped Voltages
Ccombined = C_bnk_y + 100*eep.n*3
TRV_y_groundcap = pktransrecvolt(Ccombined,L_xfm_y,VLL=Vls)[0]
print("e) Trapped Voltage in Cap (TRV):",TRV_y_groundcap/k,"kV")
```

```
Inductance: 0.36974876379109134 mH
Capacitance: 152.23728104137527 uF
a) Resonant Frequency: 670.82039325 Hz  T-cycle: 1.490711985 milisec
Inductance: 0.36974876379109134 mH
Capacitance: 456.71184312412566 uF
b) Resonant Frequency: 387.298334621 Hz T-cycle: 2.58198889747 milisec
c) Trapped Voltage in Cap (TRV): 21.5568820902 kV
d) Trapped Voltage in Cap (TRV): 21.5559671683 kV
e) Trapped Voltage in Cap (TRV): 21.5568829917 kV
```

Problem 3:

A 138kV winding on a 20MVA transformer has a leakage reactance of 0.25H and a phase to ground capacitance of 4000pF. The winding is connected Y-grounded. The transformer draws a magnetizing current of 3A RMS at a power factor (0.1 lagging) from the 138kV source when the low voltage winding is open. Assume an ideal source on the power system side.

I. Calculate the peak line to ground voltage across the winding and across the breaker contacts when the HV breaker is opened with the secondary already open if (a) the breaker clears at a natural current zero and (b) if the breaker chops an instantaneous current of 1.5 A. Simulate and compare results.

II. Using simulation only, repeat with a source inductance of 150mH and a shunt capacitance of 200pF on the source side of the breaker and comment on the differences.

In [84]:

```
# Define Givens
Vwinding = 138*k
Srating = 20*M
L_leakage = 0.25 #Henries
C_phsgrnd = 4000*eep.p #Farads
Imagrms = 3
Ipf = 0.1

# Calculate Current
Inl_rms = eep.phasor(Imagrms,np.degrees(np.arccos(Ipf)))
eep.cprint(Inl_rms,"A","RMS No-Load Current:")
# Calculate Inductance
XL = Vwinding/abs(Inl_rms.imag)
Lm = eep.reactance(XL)
print("Magnetizing Inductance:",Lm/m,"mH")
Ltot = Lm + L_leakage

# I.a Natural Current Zero
Vbreaker = pktransrecvolt(C_phsgrnd,Lm,VLN=Vwinding)[0]
print("I.a) Voltage Across Winding:",0,"V")
print("     Voltage Across Breaker:",Vbreaker/k,"kV")

# I.b Inst. Current Chop of 1.5A
Ichop = 1.5
Zo = np.sqrt(Lm/C_phsgrnd)
print("Zo:",Zo/k,"kΩ")
eep.cprint(XL*Ichop/k,"kV","I.b) Voltage Across Winding:")
print("     Voltage Across Breaker:",Ichop*Zo/k,"kV")
```

```
RMS No-Load Current: 3.0 ∠ 84.261° A
Magnetizing Inductance: 122633.49782446676 mH
I.a) Voltage Across Winding: 0 V
     Voltage Across Breaker: 390.3951378 kV
Zo: 175.095329624 kΩ
I.b) Voltage Across Winding: 69.348 ∠ 0.0° kV
     Voltage Across Breaker: 262.642994436 kV
```

II)

The addition of the inductor and shunt capacitor seemed to leave the results largely unafected. The circuits seemed to respond almost identically.

In [89]:
```
x,y=pktransrecvolt(C=12.7*eep.n,L=3.157*m,VLL=34.5*k)
y/k*u
```

Out[89]: 2.8321381691232625

In [ ]:
```

```