Joe Stanley

ECE523 - HWK6

In [ ]:

```python
# Import Libraries
import numpy as np
import matplotlib.pyplot as plt
import electricpy as ep

# Define Additional Functions

# Define Synch. Machine Eq Calculator
def synmach_Eq(Vt_pu,Itmag,PF,Ra,Xd,Xq):
    """
    Synchronous Machine Eq Calculator

    Given specified parameter set, will calculate
    the internal voltage on the q-axis (Eq).

    .. math:: E_q=V_{t_{pu}}-\\left[R_a\\cdot I_{t_{pu}}+
       j\\cdot X_q\\cdot I_{t_{pu}}+j(X_d-X_q)\\cdot I_{ad}\\right]

    where:

    .. math:: I_{t_{pu}}=I_{t_{mag}}\\cdot e^{-j(
       \\angle{V_{t_{pu}}}-\\cos^{-1}(PF))}

    .. math:: \\theta_q=\\angle{V_{t_{pu}}-\\left(R_a
       I_{t_{pu}}+j\\cdot X_qI_{t_{pu}}\\right)

    .. math:: I_{ad}=\\left|I_{t_{pu}}\\cdot\\sin(
       -\\cos^{-1}(PF)+\\theta_q)\\right|e^{j(\\theta_q
       -90°)}

    Parameters
    ----------
    Vt_pu:      complex
                Terminal voltage in per-unit-volts
    Itmag:      float
                Terminal current magnitude in per-
                unit-amps
    PF:         float
                Machine Power Factor, (+)ive values denote
                leading power factor, (-)ive values denote
                lagging power factor
    Ra:         float
                AC resistance in per-unit-ohms
    Xd:         float
                D-axis reactance in per-unit-ohms
    Xq:         float
                Q-axis reactance in per-unit-ohms

    Returns
    -------
    Eq:         complex
                Internal Synchronous Machine Voltage
                in per-unit-volts
    """
    # Calculate Required Terms
    phi = np.arccos(PF)
    Itmag = abs(Itmag)
    It_pu = Itmag*np.exp(-1j*(np.angle(Vt_pu)+phi))
    th_q = np.angle(Vt_pu - (Ra*It_pu+1j*Xq*It_pu))
    Iad = (abs(It_pu)*np.sin(phi+th_q))*np.exp(1j*(th_q-np.pi/2))
    # Calculate Eq
    Eq = Vt_pu - (Ra*It_pu+1j*Xq*It_pu+1j*(Xd-Xq)*Iad)
    return(Eq)
```

In [ ]:
```python
# Define Synch. Machine Fault Current Calculator
def synmach_Ia(t,Eq,Xd,Xdp,Xdpp,Tdp,Tdpp):
    """
    Synch. Machine Symmetrical Fault Current Calc.

    Determines the Symmetrical Fault Current of a synchronous
    machine given the machine parameters, the internal voltage,
    and the time for which to calculate.

    .. math:: I_a(t)=\\sqrt{2}\\left|E_q\\right|\\left[
       \\frac{1}{X_d}+\\left(\\frac{1}{X'_d}-\\frac{1}{X_d}
       \\right)\\cdot e^{\\frac{-t}{T'_d}}+\\left(\\frac{1}
       {X"_d}-\\frac{1}{X'_d}\\right)\\cdot e^{\\frac{-t}{T"_d}}
       \\right]

    Parameters
    ----------
    t:          float
                Time at which to calculate the fault current
    Eq:         float
                The internal machine voltage in per-unit-volts
    Xd:         float
                The Xd (d-axis) reactance in per-unit-ohms
    Xdp:        float
                The X"d (d-axis transient) reactance in
                per-unit-ohms
    Xdpp:       float
                The X"d (d-axis subtransient) reactance in
                per-unit-ohms
    Tdp:        float
                The T'd (d-axis transient) time constant of the
                machine in seconds
    Tdpp:       float
                The T"d (d-axis subtransient) time constant of
                the machine in seconds

    Returns
    -------
    Ia:         float
                Peak symmetrical fault current in per-unit-amps
    """
    # Calculate Time-Constant Term
    t_c = 1/Xd+(1/Xdp-1/Xd)*np.exp(-t/Tdp)+(1/Xdpp-1/Xdp)*np.exp(-t/Tdpp)
    # Calculate Fault Current
    Ia = np.sqrt(2)*abs(Eq)*t_c
    return(Ia)
```

In [ ]:

```
# Define Synch. Machine Asymmetrical Current Calculator
def synmach_Iasym(t,Eq,Xdpp,Xqpp,Ta):
    """
    Synch. Machine Asymmetrical Fault Current Calc.

    Determines the asymmetrical fault current of a synchronous
    machine given the machine parameters, the internal voltage,
    and the time for which to calculate.

    .. math:: I_{asym}=\\sqrt{2}\\left|E_q\\right|\\frac{1}{2}
       \\left(\\frac{1}{X"_d}+\\frac{1}{X"_q}\\right)e^{\\frac{-t}
       {T_a}}

    Parameters
    ----------
    t:          float
                Time at which to calculate the fault current
    Eq:         float
                The internal machine voltage in per-unit-volts
    Xdpp:       float
                The X"d (d-axis subtransient) reactance in
                per-unit-ohms
    Xqpp:       float
                The X"q (q-axis subtransient) reactance in
                per-unit-ohms
    Ta:         float
                Armature short-circuit (DC) time constant in seconds

    Returns
    -------
    Iasym:      float
                Peak asymmetrical fault current in per-unit-amps
    """
    # Calculate Time Constant Term
    t_c = 1/Xdpp + 1/Xqpp
    # Calculate Asymmetrical Current
    Iasym = np.sqrt(2)*abs(Eq)*1/2*t_c*np.exp(-t/Ta)
    return(Iasym)
```

In [1]:
```python
# Define Power-Factor Voltage/Current Relation
def vipf(V=None,I=None,PF=1,find=''):
    """
    Voltage / Current / Power Factor Solver

    Given two of the three parameters, will solve for the
    third, beit voltage, current, or power factor.

    Parameters
    ----------
    V:          complex
                System voltage (in volts), default=None
    I:          complex
                System current (in amps), default=None
    PF:         float
                System power factor, (+)ive values denote
                leading power factor, (-)ive values denote
                lagging power factor; default=1
    find:       str, optional
                Control argument to specify which value
                should be returned.

    Returns
    -------
    V:          complex
                System voltage (in volts), default=None
    I:          complex
                System current (in amps), default=None
    PF:         float
                System power factor, (+)ive values denote
                leading power factor, (-)ive values denote
                lagging poer factor; default=1
    """
    # Test to find Voltage
    if isinstance(V,float) and isinstance(I,complex):
        phi = -np.sign(PF)*np.arccos(PF)
        V = V*np.exp(-1j*phi)
    # Test to find Current
    elif isinstance(V,complex) and isinstance(I,float):
        phi = np.sign(PF)*np.arccos(PF)
        I = I*np.exp(-1j*phi)
    # Test to find Power Factor
    else:
        phi = np.angle(V) - np.angle(I)
        PF = np.cos(phi)
    # Return
    find = find.upper()
    if find == 'V':
        return(V)
    elif find == 'I':
        return(I)
    elif find == 'PF':
        return(PF)
    else:
        return(V,I,PF)
```

Problem 1:

1. A cylindrical rotor, synchronous machine with the machine parameters given below is operating at rated current (1.0 pu) and 85% lagging power factor when a 3 fault occurs at the machine terminals. Compute:

(a) The steady-state voltage E_q behind the synchronous impedance. Plot a phasor diagram showing E_q, V_a (terminal voltage), and I_a

(b) The voltage E" behind the synchronous impedance

(c) The initial symmetrical fault current

(d) The peak symmetrical current after 5 cycles and 10 cycles.

(e) The maximum asymmetrical current after 5 cycles and 10 cycles.

$pu := 1$ $\quad X_d := 1.05pu$ $\quad X''_d := 0.12pu$ $\quad T'_{d0} := 5.6sec$

$X_q := 1.02pu$ $\quad X''_q := 0.15pu$ $\quad T'_d := 1.1sec$

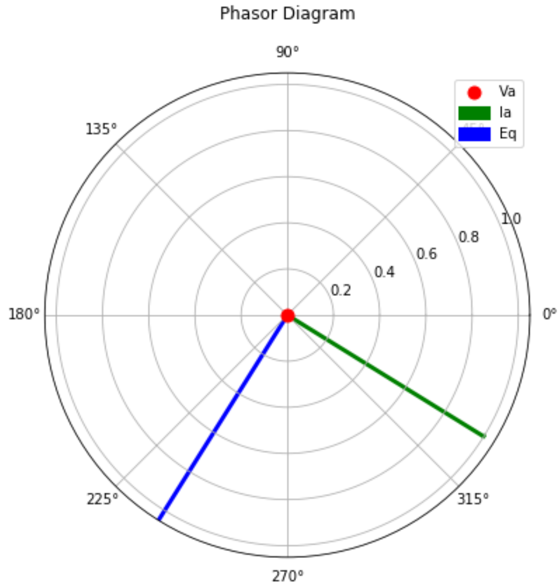
$X'_d := 0.23pu$ $\quad X_2 := 0.12pu$ $\quad T''_d := 0.035sec$

$X'_q := 0.23pu$ $\quad R_a := 0.0055pu$ $\quad T_a := 0.16sec$

In [3]:

```python
# Define Parameters - Set Va as Reference Voltage
I = 1.0
PF = 0.85
Xd = 1.05
Xq = 1.02
Xdp = 0.23
Xqp = 0.23
Xdpp = 0.12
Xqpp = 0.15
X2 = 0.12
Ra = 0.0055
Tpd0 = 5.6 # sec
Tpd = 1.1 # sec
Tppd = 0.035 # sec
Ta = 0.16 # sec

### A)

# Define Terminal Voltage According to Faulted Condition
Va = ep.phasor(0,0)
print("a)")
ep.cprint(Va,"V-pu","Terminal Voltage:")

# Calculate the angle of the Current
Ia = vipf(Va,I,PF,"I")
ep.cprint(Ia,"A-pu","Current:")

# Calculate Eq
Eq = synmach_Eq(Va,Ia,PF,Ra,Xd,Xq)
ep.cprint(Eq,"V-pu","Internal Voltage (Eq):")

# Plot Phasor Diagram
ep.phasorplot([Va,Ia,Eq],labels=["Va","Ia","Eq"],size=6,linewidth=3,colors=['red','green','blue'])

### B)

# Calculate E"q
Eqpp = synmach_Eq(Va,Ia,PF,Ra,Xdpp,Xqpp)
print("\nb)")
ep.cprint(Eqpp,"V-pu","Subtransient Internal Voltage:")

### C)

# Calculate Initial Symmetrical Fault Current
Isym0 = synmach_Ia(0,Eqpp,Xd,Xdp,Xdpp,Tpd,Tppd)
print("\nc)\nInitial Symmetrical Fault Current:",round(Isym0,3),"A-pu")

### D)

# Calculate Symmetrical Fault Current after 5 and 10 cycles
t = ep.tcycle([5,10])
Isym_5_10 = synmach_Ia(t,Eqpp,Xd,Xdp,Xdpp,Tpd,Tppd)
print("\nd)\nSymmetrical Fault Current (5-cycles):",round(Isym_5_10[0],3),"A-pu")
print("Symmetrical Fault Current (10-cycles):",round(Isym_5_10[1],3),"A-pu")

### E)

# Calculate Asymmetrical Fault Current after 5 and 10 cycles
Iasym_5_10 = synmach_Iasym(t,Eqpp,Xdpp,Xqpp,Ta)
print("\ne)\nAsymmetrical Fault Current (5-cycles):",round(Iasym_5_10[0],3),"A-pu")
print("Asymmetrical Fault Current (10-cycles):",round(Iasym_5_10[1],3),"A-pu")
```

```
a)
Terminal Voltage: 0.0 ∠ 0.0° V-pu
Current: 1.0 ∠ -31.788° A-pu
Internal Voltage (Eq): 1.05 ∠ -122.097° V-pu
```




```
b)
Subtransient Internal Voltage: 0.12 ∠ -123.888° V-pu

c)
Initial Symmetrical Fault Current: 1.416 A-pu

d)
Symmetrical Fault Current (5-cycles): 0.759 A-pu
Symmetrical Fault Current (10-cycles): 0.663 A-pu

e)
Asymmetrical Fault Current (5-cycles): 0.757 A-pu
Asymmetrical Fault Current (10-cycles): 0.45 A-pu
```

Problem 2:

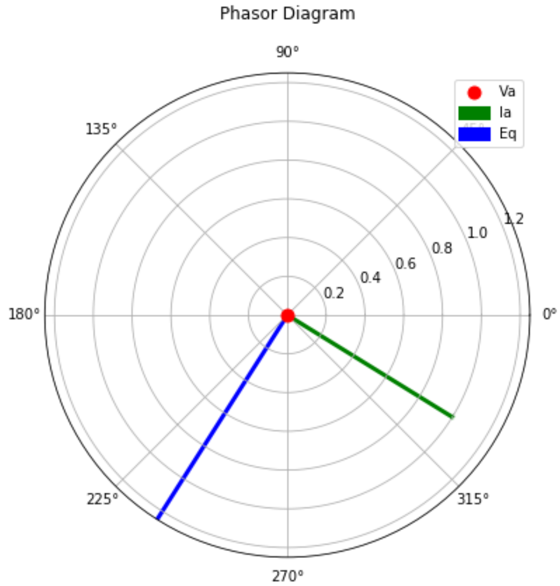
2. Repeat problem 1 using the data for the salient pole machine given below.

$X_d := 1.25\text{pu}$	$X''_d := 0.24\text{pu}$	$T'_{d0} := 5.6\text{sec}$
$X_q := 0.75\text{pu}$	$X''_q := 0.34\text{pu}$	$T'_d := 1.8\text{sec}$
$X'_d := 0.37\text{pu}$	$X_2 := 0.24\text{pu}$	$T''_d := 0.035\text{sec}$
$X'_q := 0.75\text{pu}$	$R_a := 0.009\text{pu}$	$T_a := 0.15\text{sec}$

In [4]:

```python
# Define Machine Parameters
Xd = 1.25
Xq = 0.75
Xdp = 0.37
Xqp = 0.75
Xdpp = 0.24
Xqpp = 0.34
X2 = 0.24
Ra = 0.009
Tdp0 = 5.6
Tdp = 1.8
Tdpp = 0.035
Ta = 0.15

### A)

# Define Terminal Voltage According to Faulted Condition
Va = ep.phasor(0,0)
print("a)")
ep.cprint(Va,"V-pu","Terminal Voltage:")

# Calculate the angle of the Current
Ia = vipf(Va,I,PF,"I")
ep.cprint(Ia,"A-pu","Current:")

# Calculate Eq
Eq = synmach_Eq(Va,Ia,PF,Ra,Xd,Xq)
ep.cprint(Eq,"V-pu","Internal Voltage (Eq):")

# Plot Phasor Diagram
ep.phasorplot([Va,Ia,Eq],labels=["Va","Ia","Eq"],size=6,linewidth=3,colors=['red','green','blue'])

### B)

# Calculate E"q
Eqpp = synmach_Eq(Va,Ia,PF,Ra,Xdpp,Xqpp)
print("\nb)")
ep.cprint(Eqpp,"V-pu","Subtransient Internal Voltage:")

### C)

# Calculate Initial Symmetrical Fault Current
Isym0 = synmach_Ia(0,Eqpp,Xd,Xdp,Xdpp,Tdp,Tdpp)
print("\nc)\nInitial Symmetrical Fault Current:",round(Isym0,3),"A-pu")

### D)

# Calculate Symmetrical Fault Current after 5 and 10 cycles
t = ep.tcycle([5,10])
Isym_5_10 = synmach_Ia(t,Eqpp,Xd,Xdp,Xdpp,Tdp,Tdpp)
print("\nd)\nSymmetrical Fault Current (5-cycles):",round(Isym_5_10[0],3),"A-pu")
print("Symmetrical Fault Current (10-cycles):",round(Isym_5_10[1],3),"A-pu")

### E)

# Calculate Asymmetrical Fault Current after 5 and 10 cycles
Iasym_5_10 = synmach_Iasym(t,Eqpp,Xdpp,Xqpp,Ta)
print("\ne)\nAsymmetrical Fault Current (5-cycles):",round(Iasym_5_10[0],3),"A-pu")
print("Asymmetrical Fault Current (10-cycles):",round(Iasym_5_10[1],3),"A-pu")
```

```
a)
Terminal Voltage: 0.0 ∠ 0.0° V-pu
Current: 1.0 ∠ -31.788° A-pu
Internal Voltage (Eq): 1.25 ∠ -122.476° V-pu
```




```
b)
Subtransient Internal Voltage: 0.24 ∠ -123.305° V-pu

c)
Initial Symmetrical Fault Current: 1.415 A-pu

d)
Symmetrical Fault Current (5-cycles): 0.935 A-pu
Symmetrical Fault Current (10-cycles): 0.865 A-pu

e)
Asymmetrical Fault Current (5-cycles): 0.693 A-pu
Asymmetrical Fault Current (10-cycles): 0.397 A-pu
```

Problem 3:

3. A 2000 HP, 4160V, induction motor operates at a slip of 2%, with 93% efficiency at rated load. The machine parameters are given below. Assume a power factor of 0.85 lagging at rated conditions. Do the following:

(a) Sketch the positive and negative sequence equivalent circuits using the machines ratings as a base.

(b) Convert the equivalent circuits of part A to a 4160V, 100MVA base

(c) Compute the initial fault current provided by the machine to a 3 phase fault at the motor terminals (rated prefault voltage at the terminals and rated load).

(d) Repeat part C for a LL fault

$R_s := 0.02\text{pu}$ $X_s := 0.075\text{pu}$ $R_r := 0.02\text{pu}$ $X_r := 0.075\text{pu}$ $X_m := 3.0\text{pu}$

In [ ]:
```
1
```

Problem 4:

4. Given a three winding autotransformer whose H, X, and Y windings are rated at 200kV, 100kV and 10kV respectively and with short circuit test impedances of:

$MVA := MW$

$V_h := 200kV$ $\quad V_x := 100kV$ $\quad V_y := 10kV$

$Z_{hx} := 10\%$ $\quad S_{bhx} := 30MVA$

$Z_{xy} := 9\%$ $\quad S_{bxy} := 10MVA$

$Z_{hy} := 15\%$ $\quad S_{bhy} := 10MVA$

Compute the following in pu on a 50 MVA base.
Assume the H and X are connected Y-grounded, and the Y is delta.

(a) Equivalent circuit impedances Z_h, Z_x, and Z_y. Sketch positive, negative and zero sequence diagrams

(b) Autotransformer equivalent circuit impedances Z_c, Z_t, and Z_s

In [ ]:
```
1
```