Spring 2024

46705

Power Grid Analysis

Assignment 3

Subject: Symmetrical Components and Fault Analysis

Part A issued: **02.04.2024**
Part B issued: **09.04.2024**
Part C issued: **16.04.2024**

Submission date: **30.04.2024**

Introduction

The purpose of this assignment is to practice on the Learning Objectives of the Module on "Symmetrical Components and Fault Analysis" of the 46705 "Power Grid Analysis" course.

- **Describe** the Fortescue (symmetrical component) transformation for grid analysis, and **determine** sequence networks of loads, series impedances, transmission lines, rotating machines, and transformers.
- **Apply** the transformation between the phase and sequence domains in small grids and **analyze** unsymmetrical (single line-to-ground, line-to-line and double line-to-ground) faults.
- **Calculate** unsymmetrical fault currents in large grids using sequence bus impedance matrices.

A student who has fulfilled the Intended Learning Outcomes for this assignment should be able to:

- **Describe** the Fortescue (symmetrical component) transformation for grid analysis, and **explain** how the zero-, positive- and negative-sequence components lay the foundations in power grid analysis.
- **Apply** the symmetrical component method, and **determine** sequence networks of loads, series impedances, transmission lines, rotating machines, and transformers.
- **Determine** the delivery of power using sequence networks.
- **Describe** how a three-phase power system can be represented by its sequence networks and **determine** their Thévenin equivalents.

- **Describe** the conditions for single line-to-ground, line-to-line, and double line-to-ground faults, and **apply** the transformation between the phase and sequence domains to **analyze** unsymmetrical faults in small grids.
- **Determine** the bus impedance matrices for sequence networks and **compute** unsymmetrical fault currents and voltages in large grids.

Note: It is a good idea to practice these points in your conclusion and keep them in mind as they set the framework for the evaluation and examination of the course.

You will work in groups and submit one report per group. Please see the guidelines at the end of this document.

Textbook: Glover, Overbye, and Sarma, Power System Analysis & Design, 6th edition.

If you used other material in obtaining a solution (e.g., other books and papers) you should reference your source.

Part A of this assignment requests that you summarize your understanding on the symmetrical components.

Part B of this assignment requests that you summarize your understanding on the symmetrical components and fault analysis for small grids.

Part C of this assignment requests that you write a Python program that carries out fault analysis of a given network and validate the results obtained from Part B.

Note: The Python fault analysis program created in this assignment is based on the Python power flow program that was developed in an earlier Assignment.

Part A

Question A.1

According to:

> *"G. T. Heydt, S. S. Venkata, N. Balijepalli, "High Impact Papers in Power Engineering, 1900-1999," in Proc. 2000 North American Power Symposium (NAPS), vol. 1, October 2000, Waterloo, Ontario."*

the following paper:

> *"C. L. Fortescue, "Method of Symmetrical Co-Ordinates Applied to the Solution of Polyphase Networks," AIEE Transactions, vol. 37, part II, pages 1027-1140 (1918)."*

was judged to be the most important power engineering paper in the 20th century. Do you agree? Please justify your answer (in your own words). [Maximum Length of your answer: 100 words]

Question A.2

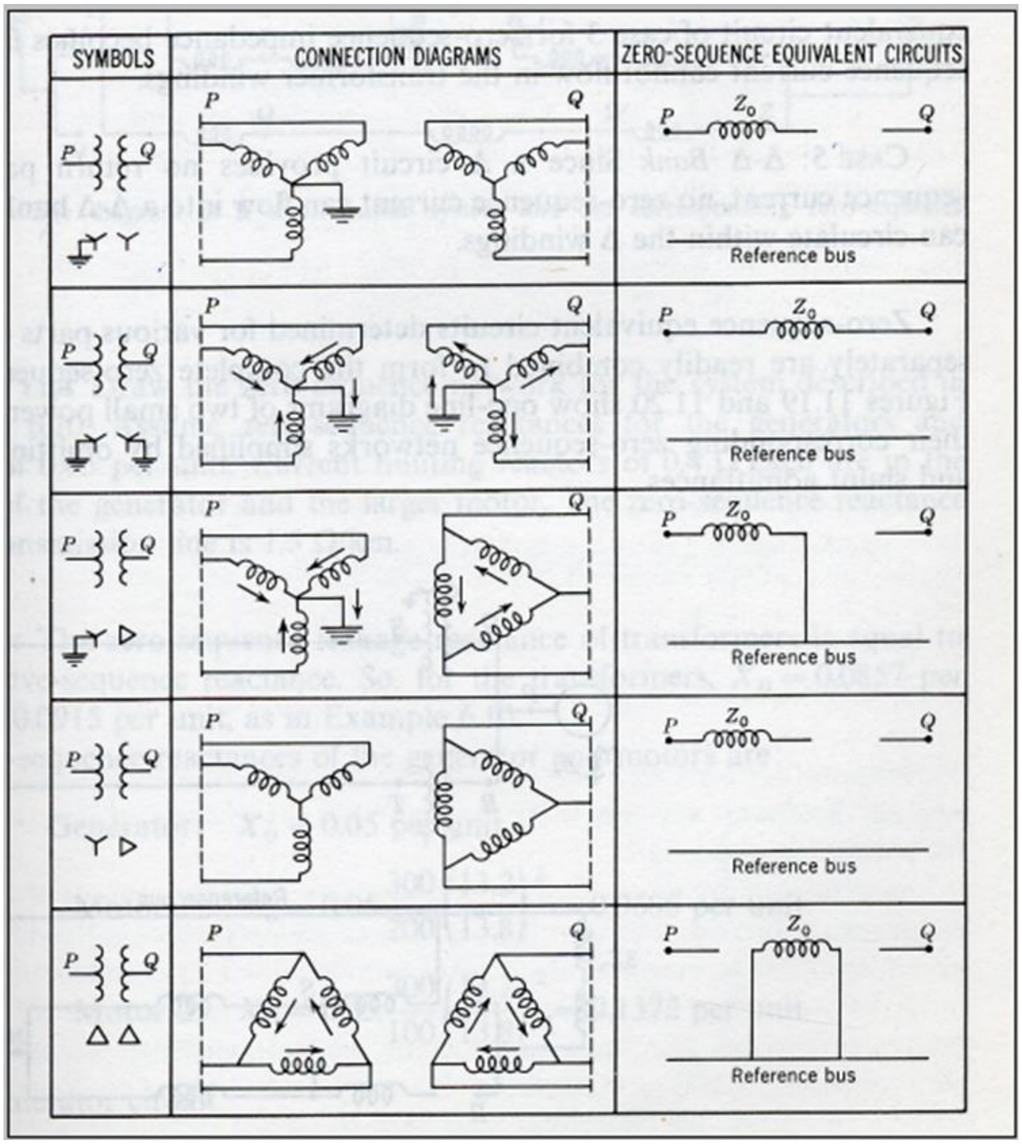
Figure 8.19 of your textbook provides the per-unit sequence networks of practical Y - Y, Y - Δ, and Δ - Δ transformers. In addition, your textbook lists certain assumptions, for drawing sequence networks (including transformers).

Table A.1 provides zero-sequence equivalent circuits for certain types of transformers.

Please explain the derivations of the zero-sequence equivalent circuits for at least 3 (of your selection) of the 5 types. (**Question A.2.1**)

How would the circuits change if the Y-connected side was grounded with an impedance Z_N? (**Question A.2.2**)

Table A.1. Zero-sequence equivalent circuits of transformers.



Question A.3

Using the voltages of Problem 8.6(a) of your textbook and the currents of Problem 8.5 of your textbook, compute the complex power dissipated based on:

- Phase components. (**Question A.3.1**)
- Symmetrical components. (**Question A.3.2**)

Part B

Question B.1

The single-line diagram of a three-phase power system is shown in the following Figure.

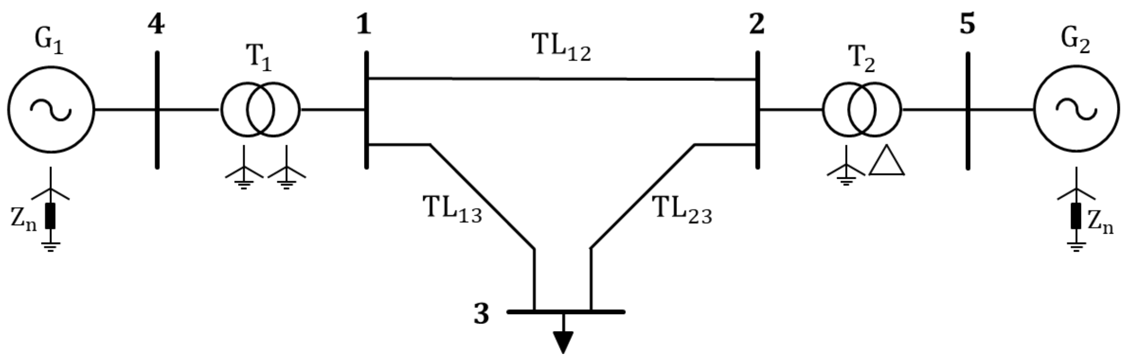


Figure B.1. Single-line diagram of three-phase power system.

Equipment ratings and per-unit reactances are given as follows:

Synchronous Generators:

G_1	100 MVA	25kV	$X_1 = X_2 = 0.2$	$X_0 = 0.05$	$X_n = 0.03$
G_2	100 MVA	13.8kV	$X_1 = X_2 = 0.2$	$X_0 = 0.05$	$X_n = 0.03$

Transformers:

T_1	100 MVA	25/230kV	$X_1 = X_2 = 0.05$	$X_0 = 0.05$
T_2	100 MVA	13.8/230kV	$X_1 = X_2 = 0.05$	$X_0 = 0.05$

Transmission Lines:

TL_{12}	100 MVA	230kV	$X_1 = X_2 = 0.1$	$X_0 = 0.3$
TL_{13}	100 MVA	230kV	$X_1 = X_2 = 0.1$	$X_0 = 0.3$
TL_{23}	100 MVA	230kV	$X_1 = X_2 = 0.1$	$X_0 = 0.3$

Pre-fault voltage: 1.0 per unit.

Using a 100-MVA, 230-kV base for the transmission lines:

- Draw the per-unit sequence networks. (**Question B.1.1**)
- Reduce the sequence networks to their Thévenin equivalents "looking in" at bus 3, using inspection of the sequence networks. (**Question B.1.2**)

Neglect Δ – Y phase shifts.

<u>Question B.2</u>

Compute the fault currents and voltages at the fault for the following faults at **bus 3**:

- A bolted three-phase fault. (**Question B.2.1**)
- A bolted single line-to-ground fault. (**Question B.2.2**)
- A bolted line-to-line fault. (**Question B.2.3**)
- A bolted double line-to-ground fault. (**Question B.2.4**)

Please include a figure with the connections of the sequence networks for faults in Questions B.2.2, B.2.3, and B.2.4.

Discuss the properties of each fault based on your findings. (**Question B.2.5**)

<u>Question B.3</u>

Compute the fault currents and voltages at the fault for the following faults at **bus 1**:

- A bolted three-phase fault. (**Question B.3.1**)
- A bolted single line-to-ground fault. (**Question B.3.2**)
- A bolted line-to-line fault. (**Question B.3.3**)
- A bolted double line-to-ground fault. (**Question B.3.4**)

You do not need to include figures.

Discuss the properties of each fault based on your findings. (**Question B.3.5**)

<u>Question B.4</u>

Compute the fault currents and voltages at the fault for the following faults at **bus 2**:

- A bolted three-phase fault. (**Question B.4.1**)
- A bolted single line-to-ground fault. (**Question B.4.2**)
- A bolted line-to-line fault. (**Question B.4.3**)
- A bolted double line-to-ground fault. (**Question B.4.4**)

You do not need to include figures.

Discuss the properties of each fault based on your findings. (**Question B.4.5**)

Part C

The aim of Part C of this assignment is to get familiar with the main steps associated with the process of fault analysis. You will write a small program in Python that calculates short-circuit currents and voltages, for a given electrical network, for the following types of faults:

- 3-Phase Balanced faults;
- Single Line-to-Ground faults (at phase a);
- Line-to-Line faults (between phases b and c);
- Double Line-to-Ground faults (between phases b and c).

The main script for the fault analysis program is provided in Table C.1. You will find this script in DTU-Learn as `main4FA.py` under Assignment 3. The lines in the script, which are colored **RED** denote functions and *.py files that you should write to complete the assignment. In BOLD, you will find the information on the given network (**`filename`**), location of the fault (**`FaultBus`**), type of the fault (**`FaultType`**), impedance of the fault (**`FaultImpedance`**), and the prefault voltage (**`PrefaultVoltage`**).

Table C.1: A Python script (`main4FA.py`) that carries out fault analysis.

```
import FaultAnalysis_46705 as fa    # import Fault Analysis functions
import LoadNetworkData4FA as lnd4fa # load the network data to global variables
filename = "./TestSystem4FA.txt"
lnd4fa.LoadNetworkData4FA(filename) # makes Zbus0 available as lndfa.Zbus0 etc.
# Carry out the fault analysis ...
FaultBus = 3
# FaultType: 0 = 3-phase balanced fault; 1 = Single Line-to-Ground fault;
#            2 = Line-to-Line fault;     3 = Double Line-to-Ground fault.
FaultType = 1
FaultImpedance = 0 # (in pu)
PrefaultVoltage = 1.000 # (in pu)
# Iph: phase current array (0: phase a; 1: phase b; 2: phase c).
# Vph_mat: phase line-to-ground voltages (rows: busses; columns: phases a, b, c).
Iph,Vph_mat = fa.FaultAnalysis(lnd4fa.Zbus0,lnd4fa.Zbus1,lnd4fa.Zbus2,lnd4fa.bus_to_ind,
                               FaultBus,FaultType,FaultImpedance,PrefaultVoltage)
# Display results
fa.DisplayFaultAnalysisResults(Iph,Vph_mat,FaultBus,FaultType,FaultImpedance,PrefaultVoltage)
print('**********End of Fault Analysis**********')
```

The functions that you should write are:

- The function **`LoadNetworkData4FA`** (stored in **`LoadNetworkData4FA.py`**), which reads in all the necessary information about the network (busses, generators, lines, and transformers). The function creates the matrices and arrays needed to carry out fault analysis.
- The function **`FaultAnalysis()`** (stored in **`FaultAnalysis_46705.py`**), which is the essential part of the program. This function carries out the fault analysis calculations.
- The function **`DisplayFaultAnalysisResults()`** (stored in **`FaultAnalysis_46705.py`**), which displays the results from the fault analysis.

In what follows, you will find guidelines for the construction of each of the above functions.

Task 1 - Construction of the **`FaultAnalysis()`** function

On DTU-LEARN, under the link for Assignment 3 you find the file `FaultAnalysis_46705.py`, which contains empty definitions of functions needed for the fault analysis computations (with hints). Your task in the following is to complete all the functions in that file.

A Python skeleton for the `FaultAnalysis()` function is provided in Table C.2.

The input to the function `FaultAnalysis()` is the N×N bus impedance matrices of the zero-, positive-, and negative-sequence networks, i.e., `Zbus0`, `Zbus1`, and `Zbus2`, respectively, a mapping from bus numbers to the corresponding indices in the bus matrices and arrays `bus_to_ind`, and the fault characteristics, i.e., the location of the fault (`fault_bus`), type of the fault (`fault_type`), impedance of the fault (`Zf`), and the prefault voltage (`Vf`).

As output, the function returns the following:

- the phase (fault) currents, `Iph`, an array that stores phase a, b, and c currents, i.e., element `Iph[0]` contains the phase a current, `Iph[1]` contains the phase b current, etc.), and
- the phase (fault) line-to-ground voltages, `Vph_mat`, an Nx3 matrix that stores the voltages at each bus in rows and phase a, b, c in columns, i.e., element `Vph_mat[0][0]` contains voltage at bus 1 (first row), phase a (first column), element `Vph_mat[1][1]` contains voltage at bus 2 (second row), phase b (second column), element `Vph_mat[1][2]` contains voltage at bus 2 (second row), phase c (third column), etc.

The `FaultAnalysis()` function presented in Table C.2, includes 4 sub-functions colored **RED**, which you are asked to write.

Table C.2: Skeleton for the function `FaultAnalysis()`.

```python
def FaultAnalysis(Zbus0,Zbus1,Zbus2,bus_to_ind,fault_bus,fault_type,Zf,Vf):
    # calculate sequence fault currents
    Iseq = Calculate_Sequence_Fault_Currents(Zbus0,Zbus1,Zbus2,bus_to_ind,fault_bus,fault_type,Zf,Vf)
    # calculate sequence fault voltages
    Vseq_mat = Calculate_Sequence_Fault_Voltages(Zbus0,Zbus1,Zbus2,bus_to_ind,fault_bus,Vf,Iseq)
    # convert sequence currents to phase (fault) currents
    Iph = Convert_Sequence2Phase_Currents(Iseq)
    # convert sequence voltages to phase line-to-ground (fault) voltages
    Vph_mat = Convert_Sequence2Phase_Voltages(Vseq_mat)
    return Iph, Vph_mat
```

Question C.1

- Write the function `Calculate_Sequence_Fault_Currents()`, which calculates the sequence fault currents, and stores them in the array `Iseq`, where element `Iseq[0]` contains the zero-sequence current, `Iph[1]` contains the positive-sequence current, etc. In the report, you are expected to describe how you came up with your answer, e.g., which equation you used and implemented in Python, etc. (**Question C.1.1**)

- Write the function `Calculate_Sequence_Fault_Voltages()`, which calculates the sequence (fault) voltages, and stores them in the Nx3 matrix `Vseq_mat`, where element `Vseq_mat[0][0]` contains the bus 1 (first row) zero-sequence (first column) voltage, `Vseq[1][1]` contains the bus 2 (second row), positive-sequence (second column) voltage, `Vseq[1][2]` contains the bus 2 (second row), negative-sequence (third column) voltage, etc. In the report, you are expected to describe how you came up with your answer, e.g., which equation you used, etc. (**Question C.1.2**)
- Write the function `Convert_Sequence2Phase_Currents()`, which converts the sequence current array `Iseq` to the phase current array `Iph`. In the report, you are expected to describe how you came up with your answer, e.g., which equation you used, etc. (**Question C.1.3**)
- Write the function `Convert_Sequence2Phase_Voltages()`, which converts the sequence voltage matrix `Vseq_mat` to the phase line-to-ground voltage matrix `Vph_mat`. In the report, you are expected to describe how you came up with your answer, e.g., which equation you used, etc. (**Question C.1.4**)

Task 2 - Construction of the **`LoadNetworkData4FA()`** function

On DTU-LEARN, under the link for Assignment 3 you find the file `LoadNetworkData4FA.py` with the first lines of the function `LoadNetworkData4FA()`, which contains the relevant model data required to carry out fault analysis calculations.

The model data is stored in a text file that should be read into Python. On DTU-LEARN, under the link for Assignment 3, you will find a text file, `TestSystem4FA.txt`, which includes the model data for the system of Part B.

The Python file `ReadNetworkData4FA.py`, which is also uploaded on DTU-LEARN, under the link for Assignment 3, provides function `read_network_data_4fa_from_file()` that is used to get the relevant information from the text file `TestSystem4FA.txt` and store the information in several objects. The variables returned by `read_network_data_4fa_from_file()` are based on the respective function of the Python power flow program of your earlier Assignment. The additional information required for the fault analysis calculations is given below in **BOLD**:

- `gen_data`: contain `[bus_nr,MVA_size,P_Gen,`**`X,X2,X0,Xn,grnd`**`]` (**`X`: positive sequence, `X2`: negative sequence, `X0`: zero sequence, and `Xn`: neutral in pu, `grnd`: 1=grounded; 0=ungrounded**).
- `line_data`: contain `[fr_bus,to_bus,ID_label,R,X,B_half,`**`X2,X0`**`]` (pu system base, `X`**: positive sequence, `X2`: negative sequence, `X0`: zero sequence**).
- `tran_data`: contain `[fr_bus,to_bus,ID_label,R,X,n_pu,ang_deg,`**`fr_co,to_co,X2,X0`**`]` (`X`**: positive sequence** in pu, **`fr_co`: from connection, `to_co`: to connection, where transformer connection: 1=Y, 2=Yg, 3=D, `X2`: negative sequence, and `X0`: zero sequence in pu**).

In Table C.3, you will find a skeleton for the `LoadNetworkData4FA.py` file and the `LoadNetworkData4FA()` function.

Table C.3: Example of what the first and last few lines of `LoadNetworkData4FA.py` could look like.

```
import numpy as np
import ReadNetworkData4FA as rd4fa

def LoadNetworkData4FA(filename):
    global Ybus,Sbus,V0,buscode,pq_index,pv_index,ref,Y_fr,Y_to,br_f,br_t,br_Y,S_LD,\
    ind_to_bus,bus_to_ind,MVA_base,bus_labels,Ybus0,Ybus2,Zbus0,Zbus1,Zbus2
# read in the data from the file...
    bus_data,load_data,gen_data,line_data,tran_data,mva_base,bus_to_ind,ind_to_bus = \
    rd4fa.read_network_data_4fa_from_file(filename)

###############################################################################################
# Construct the Ybus (positive-sequence), Ybus0 (zero-sequence), and Ybus2 (negative-sequence)
# matrices from elements in the line_data and tran_data
# Keep/modify code from the Python power flow program as needed
###############################################################################################
    N = len(bus_data) # Number of busses
    Ybus = np.zeros((N,N),dtype=complex)
    Ybus0 = np.zeros((N,N),dtype=complex)
    Ybus2 = np.zeros((N,N),dtype=complex)
# Continue with your code here...
# ...
# End your code by deriving the Zbus matrices (Zbus0, Zbus1, Zbus2)
    Zbus0 = np.linalg.inv(Ybus0)
    Zbus1 = np.linalg.inv(Ybus)
    Zbus2 = np.linalg.inv(Ybus2)

    return
```

Question C.2

Complete the `LoadNetworkData4FA()` function, which creates automatically the `Ybus` (positive-sequence), `Ybus0` (zero-sequence), and `Ybus2` (negative-sequence) matrices for the sequence networks of a given electrical network, and then derives the sequence bus impedance matrices `Zbus0`, `Zbus1`, and `Zbus2`. In your implementation, you should consider the assumptions made for the fault analysis (e.g., ignore shunts, Δ-Y phase shifts etc.). It is recommended to modify the respective function that you wrote for the Python power flow program in your earlier Assignment. In the report, you are expected to describe how you came up with your answer, e.g., which equation you used, etc. Emphasis should be put on the transformer connections (your code should work for any transformer connection).

Task 3 - Construction of the **`DisplayFaultAnalysisResults()`** function

When the fault analysis is completed, the next step in the main script in Table C.1 is to display the results. The function **`DisplayFaultAnalysisResults()`**, stored in **`FaultAnalysis_46705.py`**, displays phase fault currents and phase line-to-ground voltages (at all busses), for a given fault. An example of how such printout of results should look like is provided in Table C.4.

Table C.4: Example of the output of **DisplayFaultAnalysisResults()**.

```
===============================================================
|                   Fault Analysis Results                    |
===============================================================
| Single Line-to-Ground Fault at Bus 3, phase a.              |
| Prefault Voltage: Vf = 1.000  (pu)                          |
| Fault Impedance:  Zf = 0.000  (pu)                          |
===============================================================
| Phase Currents ---------------------------------------------|
| --------------                                              |
|     ---- Phase a ----| ---- Phase b ----| ---- Phase c ----|
|     -----------------|------------------|------------------|
|      Mag(pu)  Ang(deg)|  Mag(pu)  Ang(deg)|  Mag(pu)  Ang(deg)|
|       5.466    -90.00 |   0.000   -90.00  |   0.000   -90.00  |
===============================================================
| Phase Line-to-Ground Voltages ------------------------------|
| -----------------------------                               |
|    | ---- Phase a ----| ---- Phase b ----| ---- Phase c ----|
|Bus|------------------|------------------|------------------|
|    |  Mag(pu)  Ang(deg)|  Mag(pu)  Ang(deg)|  Mag(pu)  Ang(deg)|
|---| --------  ------- | --------  ------- | --------  ------- |
| 1 |   0.427     0.00 |  0.950  -114.26  |   0.950   114.26  |
| 2 |   0.484     0.00 |  0.928  -110.99  |   0.928   110.99  |
| 3 |   0.000   180.00 |  1.022  -122.11  |   1.022   122.11  |
| 4 |   0.549     0.00 |  0.956  -115.05  |   0.956   115.05  |
| 5 |   0.636     0.00 |  0.922  -110.15  |   0.922   110.15  |
===============================================================
```

Question C.3

Write the function **DisplayFaultAnalysisResults()**, stored in **FaultAnalysis_46705.py**, which displays the results of the fault analysis, according to Table C.4. The function is expected to work for any type of fault (as described in the beginning of Part C of this assignment).

Question C.4

Use the Fault Analysis program you created to validate the results of Part B.

- For each fault type in Question B.2, use prefault voltage Vf=1.05, and provide the results in 4 separate Tables similar to Table C.4. (**Question C.4.1**)
- Repeat for each fault type in Question B.3, using prefault voltage Vf=1.05 and fault impedance Zf = j0.05. (**Question C.4.2**).

Guidelines for the Writing of the Report

When writing the report for the assignment, the below listed points should be followed:

- The report must start with a self-evaluation section, where the authors state whether the report represents an even contribution from all group members or whether some participants were not effectively contributing to the report.
- The front page of the report should contain your names and student numbers.
- The report shall contain the answer to each of the questions. The answers shall be clearly separated from each other and come in the same order as the questions in the assignment. It is not necessary to repeat the text from the assignment in your report.
- When answering the questions, both the results as well as an explanation on how you came up with your results should be included in the answer.
- Always when asked to provide plots of something (time responses, curves etc.) you must provide your interpretation of the figure.
- Make sure that the assignment's intended learning outcomes are reflected in your report.
- The report should be ended by a conclusion explaining what you have learned by working on the assignment.
- When your reports are evaluated, points are awarded for the general setup of the report. We are looking for a professional look of the report, where among others the following is considered:
    - Readability (appropriate font sizes, margins etc.) and consistence in styles applied (same look for body text, headings, captions etc. throughout the report).
    - Presentation of equations should be of an appropriate quality (do not copy/paste equations as screenshot pictures from the lecture handouts).
    - Plots and figures should be of good quality (no fuzzy looking screenshots in the report).
    - Figures and tables must have a label and caption (e.g. Figure 1: plot of active power in respect to ....).
    - If you are citing external material, you need to include a properly set-up reference list (you can use any citation style you prefer).
    - Does the report contain a conclusion section?
- You will also be awarded points for the general formulation of the report where focus is on:
    - The flow in the text when explaining how you came up with your answers and correct use of relevant terminology.
    - Interpretation and reflection of results when appropriate.
    - Grammar and spelling.

Guidelines for the Upload of the Report and Python Code

The report submission is carried out on DTU-LEARN where the following shall be uploaded:

- PDF version of your report, with your Python code in appendix.

- Your Python code used for solving the assignment (the *.py). The code shall include your comments where you explain the meaning behind the code.