Collin Griffin
Dr. Kerestes
ECE 2774
28 April 2024

Milestone 4

Repository

Link: https://github.com/griffincj/Power-System-Analysis

Overview

This milestone adds the ability to conduct a fault study in addition to the power flow study that was implemented for milestone 3. A command line text interface is presented to the user upon startup that allows selection between the two study types. This report is expanded from the first part of milestone 4 (i.e. simply allowing the user to request a fault study *or* power flow study, and building the sequence networks). This now contains the algorithm to calculate faulted voltages and currents for all fault types.

Main Module

The output required for this module can be displayed by running the main method of the main.py module. The main module now offers a selection between running a fault study and a power flow study. Upon selecting the fault study option, the user will also be prompted to enter the bus to fault and the pre-fault voltage, as well as whether the fault is symmetric or asymmetric, and what specific type of asymmetric fault it is, if applicable.
The fault impedance is defaulted to zero, and can be modified in the method signature. A fault impedance of zero represents a bolted fault/worst case, and thus is the most useful in calculating the sizing of elements impacted by a fault.
After the study is run, the user will be prompted to prompted to select another study to be performed and will continue to be prompted until the program is exited. This is for ease of use for the user.

PowerSystem

The power system class has been extended to add several new methods for solving the power flow problem.

Methods

- **Calc_z_bus(self)**
  - Starts by calculating the y-bus and then adding the positive sequence reactance for each generator to the y-bus. The y-bus is then inverted, which returns the **z-bus**

- Negative sequence matrix is virtually the same, but with negative sequence reactance in place of positive sequence reactance.
- Zero sequence has additional of accounting for grounding. If a circuit element is ungrounded, a negative one is passed, and the code conditionally sets that particular element in the matrix to 0 (i.e. 1/infinity)

- **Calc_balanced_fault(self, selected_bus, pre_fault_v)**
  - The fault current is calculated from the pre-fault voltage phasor and the z-bus value at the diagonal for the faulted bus. The i_vector is all zeroes, except for the faulted bus
  - The voltage vector is calculated by multiplying the z-bus and the i_vector. For the final voltage vector, the calculated voltage is subtracted from the pre-fault voltage.
- **Calc_unbalanced_fault(self, selected_bus, pre_fault_v, fault_type, fault_impedance: 0):**
  - Calculates unbalanced faults: line to line, single line to ground, and double line to ground
  - Uses all three sequence networks in algorithm
  - First calculates the zero, positive, and negative sequence current using equations from class. These differ based on the type of fault selected
  - Then calculate sequence voltages by taking a vector [0, Vf, 0] and subtracting the product of a 3x3 matrix with the k bus's zero, positive, and negative impedances on the diagonal and zeros elsewhere and a vector of the zero, positive, and negative current for the faulted (N) bus.
  - The phase voltages are then calculated by multiplying the sequence voltage matrix times the "A" matrix
  - The phase currents for the faulted bus are calculated by taking the product of the "A" matrix and the sequence current vector