**TERM PROJECT EE 271**

**Fall 2022**

**Design of a Distance Relay**

**Objective**

The objective of this project is to design and implement a 3-step digital distance relay. The distance relay function should be designed and algorithms be written and tested. The distance relay should be demonstrated on a set of data (provided in COMTRADE format - therefore part of the project will require that you familiarize yourself with this standard and write a program to read a standard COMTRADE file).

The COMTRADE standard can be downloaded from the IEEE Explore: *IEEE Standard C37.111-2013, “Common Format for Transient Data Exchange (COMTRADE) for Power Systems*”.

A short description of the distance relay function is given below.

**Distance (Modified Impedance) Function**: The relay monitors the three currents and three voltages of the line and computes the impedance “looking” into a circuit. The impedance computation is described in the class notes, chapter 5 (you will note that there are nine elements).

The relay has the following settings:

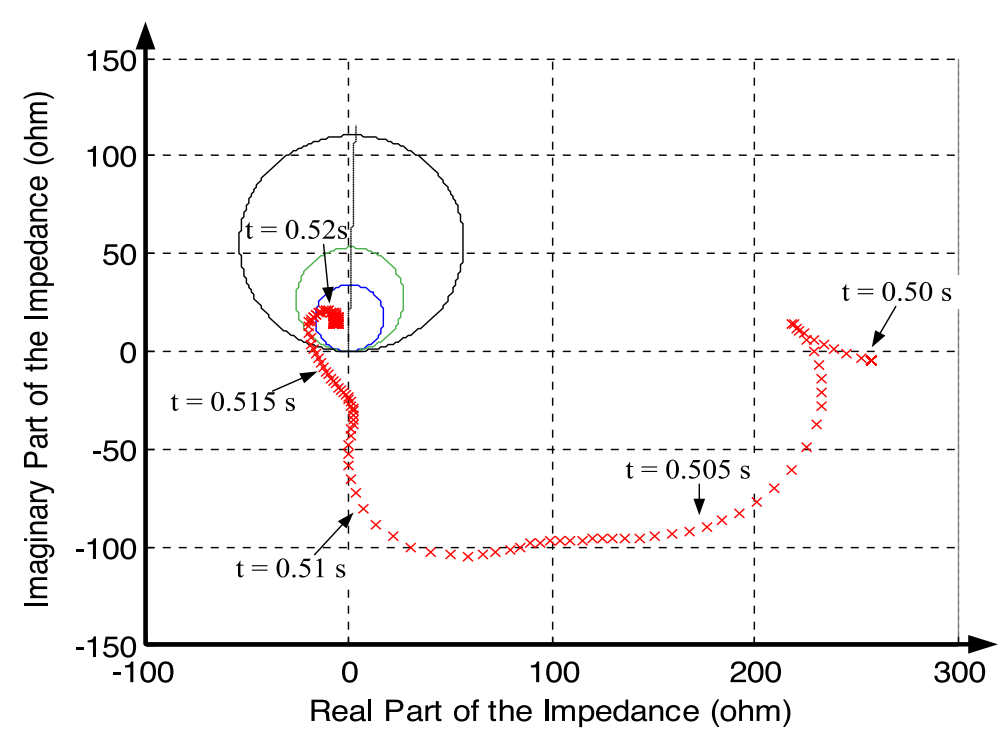
(a) zone 1, zone 2 and zone 3 impedance (including the magnitudes and the phase angles);

(b) compensation factor;

(c) zone 1, zone 2 and zone 3 time delay.

The characteristics of the relay are illustrated in Figure 1 (blue, green and black circles correspond to zone 1, zone 2 and zone 3, respectively). Once the impedance “enters” a certain zone of the relay characteristics, a timer is initiated. As long as the impedance remains in the certain zone of the relay characteristics, the timer continues. When the time delay setting has been reached, the relay trips the line under protection. If the impedance “leaves” the characteristic of the relay, the timer resets.

The impedance “seen” by the relay is also depicted in Figure 1 (the trace is shown with red crosses). Note that in this figure we only show the performance of one element, which is the one that trips the line the fastest (the other eight elements do not trip the line or trip the line slower than this element; you can show all nine of them to check which one is the “fastest” element). In Figure 1, the fault occurs at 0.5 seconds. We can observe that the trace enters zone 3, zone 2 and zone 1 of the relay, subsequently. The relay (zone 1) detects the fault at 0.519 sec and trips the line at 0.519 sec + zone 1 time delay (one of the settings).



**Figure 1 Example characteristics of a distance relay and an example trace of impedance “seen” by the relay**

Other examples of figures can be found in the following paper [1] (see figure 6, 9, 12, 15, 18, 21 of the paper):

[1] Y. Liu, S. Meliopoulos, R. Fan, L. Sun and Z. Tan, “Dynamic State Estimation Based Protection on Series Compensated Transmission Lines”, *IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp 2199-2209, Oct. 2017.

**Implementation and Demonstration**

This project can be completed in a number of ways. You can use a programming language of your choice (including Matlab, C, C++, etc). The exchange of data will be via the COMTRADE format. For your convenience, the following procedural tasks are suggested:

**Task 0**: Develop a user interface for (a) entering the relay settings and (b) reporting the relay actions.

**Task 1**: Develop a computer program to read COMTRADE data. You may also write a computer program to display this data graphically, so that you can verify that the data have been read correctly.

**Task 2:** Develop a computer program that emulates the function of the 3-step distance relay. Specifically,

- Compute a running power frequency AC current value (magnitude and phase) over a sliding 1-power frequency cycle window.

- Compute a running power frequency AC voltage value (magnitude and phase) over a sliding 1-power frequency cycle window.

**Task 3:** Issue a trip command according to the selected relay settings.

For demonstration purposes (or for testing your “relay”) two COMTRADE files (each COMTRADE file includes a “.cfg” file and a “.dat” file), corresponding to two events have been provided. The files contain normal operating conditions and fault conditions. The types of faults and the time when the faults occur in the events will be unknown to you. You will be given the power system from which the data have been obtained - transmission line sequence parameters of the target area and adjacent area. You do not need to know the model of the rest of the system. Then you should determine the settings of the relay from the power system data and your “relay” should be able to identify internal faults and take the correct action. Please summarize all your setting values in a table. The responses of your “relay” should be documented in your final report and you should be able to demonstrate the operation of your “relay” (including whether and when to operate). If the relay operates, can you identify the type of the fault?

**Discussion**

The description and instructions for designing a distance relay are purposely terse. Exercise your judgment and be creative. You are encouraged to use graphical displays of your data to gain insight into your project.

**Final Report Preparation**

The final report should have (as a minimum) the following structure:

Introduction and Scope

Problem Formulation

Solution Methodology – Description of your algorithm

Presentation of Results

Conclusions and Observations

Source codes

**Appendix. System parameters and COMTRADE files**

The transmission line of interest has the following parameters:

500 kV rated voltage (line to line)

Positive (negative) sequence impedance: *Z*1 = 8.5088 + *j* 274.86;

Zero sequence impedance: *Z*0 = 243.25 + *j* 889.83;

The sequence impedance parameters of adjacent lines are as follows

Positive (negative) sequence impedance: *Z*1 = 7.6767 + *j* 212.15;

Zero sequence impedance: *Z*0 = 212.44 + *j* 727.37;

Two COMTRADE files are attached. They correspond to two events:

COMTRADE file 1:

**EE271\_DistanceProtection\_Event1.cfg**

**EE271\_DistanceProtection\_Event1.dat**

COMTRADE file 2:

**EE271\_DistanceProtection\_Event2.cfg**

**EE271\_DistanceProtection\_Event2.dat**