Distance Relay Simulation using Discrete Fourier Transform

Supriyo Roy

May 6, 2025

Abstract

This report presents a detailed simulation of a numerical distance protection relay implementing three-zone mho characteristics with Discrete Fourier Transform (DFT) based phasor estimation. The project models a complete transmission line protection system including signal generation, fault simulation, phasor estimation, impedance calculation, and trip logic. The implementation uses Python with NumPy for numerical computations and Matplotlib for visualization of waveforms and impedance trajectories. The report includes comprehensive system analysis, block diagrams of the simulation architecture, and detailed discussion of the algorithms employed.

1 Introduction

Modern power systems require reliable protection schemes to maintain stability during faults. Distance relays provide selective protection by measuring the impedance between the relay location and fault point. This project implements a complete numerical distance relay simulation with the following advanced features:

- Three-zone stepped distance protection
- Mho characteristic implementation
- DFT-based phasor estimation
- Support for multiple fault types (LG, LL)
- Dynamic impedance trajectory visualization

2 System Architecture

The simulation architecture consists of several interconnected modules as shown in Figure 1.

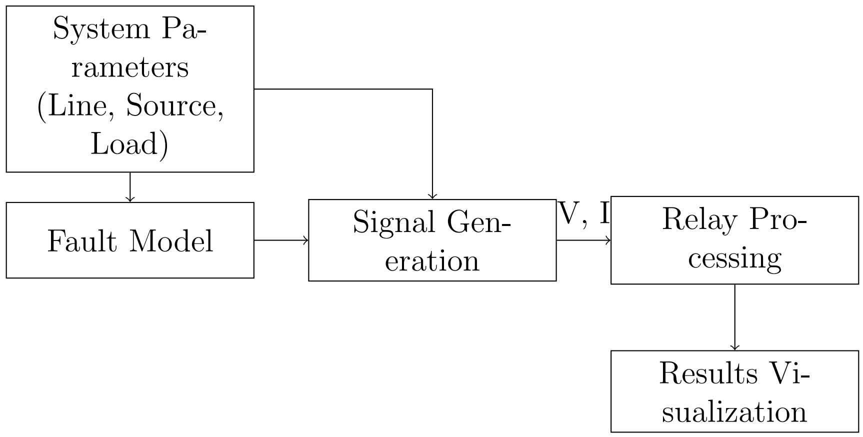

Figure 1: System block diagram

3 Detailed System Modeling

3.1 Power System Model

The simulated system consists of:

$$\begin{cases} \text{Source Voltage} & V_s = 220/\sqrt{3}\,\text{kV} \\ \text{Line Impedance} & Z_{line} = (0.02 + j0.4)\,\Omega/\text{km} \\ \text{Line Length} & L = 100\,\text{km} \\ \text{Source Impedance} & Z_s = (1 + j10)\,\Omega \\ \text{Load Impedance} & Z_{load} = (200 + j150)\,\Omega \end{cases} \tag{1}$$

3.2 Relay Algorithm Flow

The relay processing algorithm follows the sequence shown in Figure 2.

4 Implementation Details

4.1 Signal Generation Module

The signal generation creates three-phase voltages and calculates corresponding currents based on system conditions:

$$v_a(t) = V_m \cos(2\pi f t) \tag{2}$$

$$v_b(t) = V_m \cos(2\pi f t - 120°) \tag{3}$$

$$v_c(t) = V_m \cos(2\pi f t + 120°) \tag{4}$$

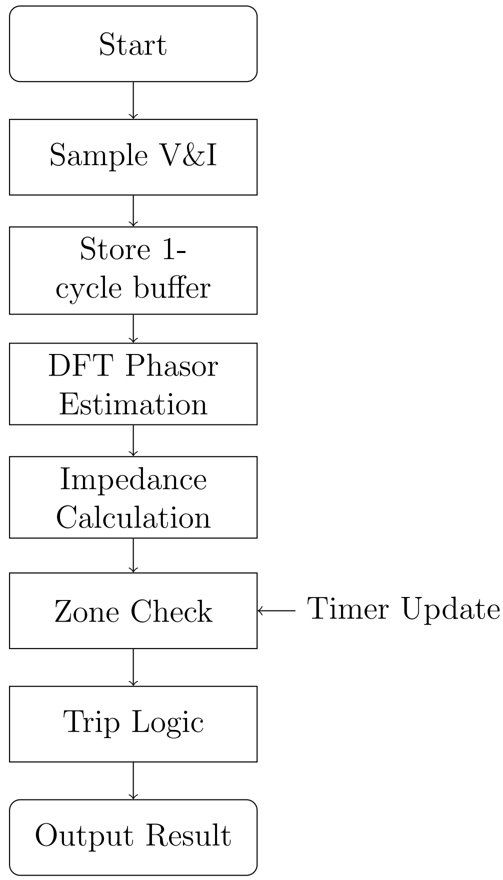

Figure 2: Relay algorithm flowchart

4.2 Fault Modeling

The fault model implements two fault types with configurable parameters:

4.2.1 Phase-to-Ground Fault

$$I_{fault} = \frac{V_{th}}{Z_{th} + Z_f} \tag{5}$$

where:

$$Z_{th} = Z_s + d \times Z_{line} \tag{6}$$

4.2.2 Phase-to-Phase Fault

$$I_{ab} = \frac{V_a - V_b}{2Z_s + 2d \times Z_{line} + Z_f} \tag{7}$$

4.3 Phasor Estimation

The DFT implementation uses a one-cycle moving window:

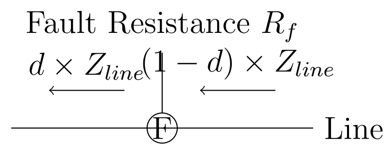

Figure 3: Fault location modeling

```
def calculate_phasor_dft(signal_window, N_cycle):
    k = 1  # Fundamental frequency component
    dft_val = np.sum(signal_window *
              np.exp(-1j * 2 * np.pi * k *
              np.arange(N_cycle) / N_cycle))
    phasor = (np.sqrt(2) / N_cycle) * dft_val.conjugate()
    return phasor
```

Key aspects:

- Sampling at 1 kHz (20 samples/cycle at 50 Hz)
- Cosine reference phasor
- RMS value calculation

4.4 Impedance Calculation

The apparent impedance calculation varies by fault type:

Table 1: Impedance Calculation Methods

Fault Type	Impedance Calculation
Phase-Ground	$Z = V_a/I_a$
Phase-Phase	$Z = (V_a - V_b)/(I_a - I_b)$
Three-Phase	$Z = V_a/I_a$ (all phases similar)

4.5 Mho Characteristic Implementation

The mho characteristic is implemented as:

$$|Z - Z_{center}| \leq |Z_{radius}| \tag{8}$$

Zone settings:

- Zone 1: $Z_1 = 0.8Z_{line}$, instantaneous
- Zone 2: $Z_2 = 1.2Z_{line}$, 0.3s delay
- Zone 3: $Z_3 = 1.5Z_{line}$, 1.0s delay

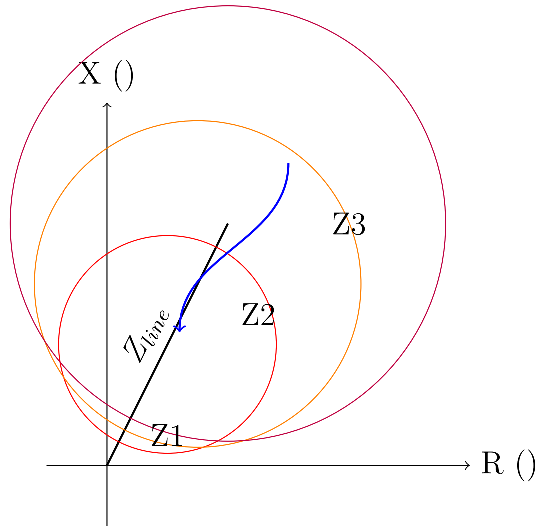

Figure 4: Mho characteristics and impedance trajectory

5 Results and Analysis

5.1 Simulation Cases

Three test cases were simulated:

Table 2: Test Cases

Case	Fault Type	Location	R_f
1	Phase-Ground	60%	1
2	Phase-Phase	30%	5
3	Phase-Ground	110%	10

5.2 Waveform Analysis

Figure 5 shows voltage and current during a phase-ground fault:

Key observations:

- Voltage depression during fault
- Current increase by 5-10x normal load current
- Phase angle shift between voltage and current

5.3 Impedance Trajectory

The R-X diagram (Figure 6) shows the impedance trajectory:

- Pre-fault: Load impedance (high R, moderate X)

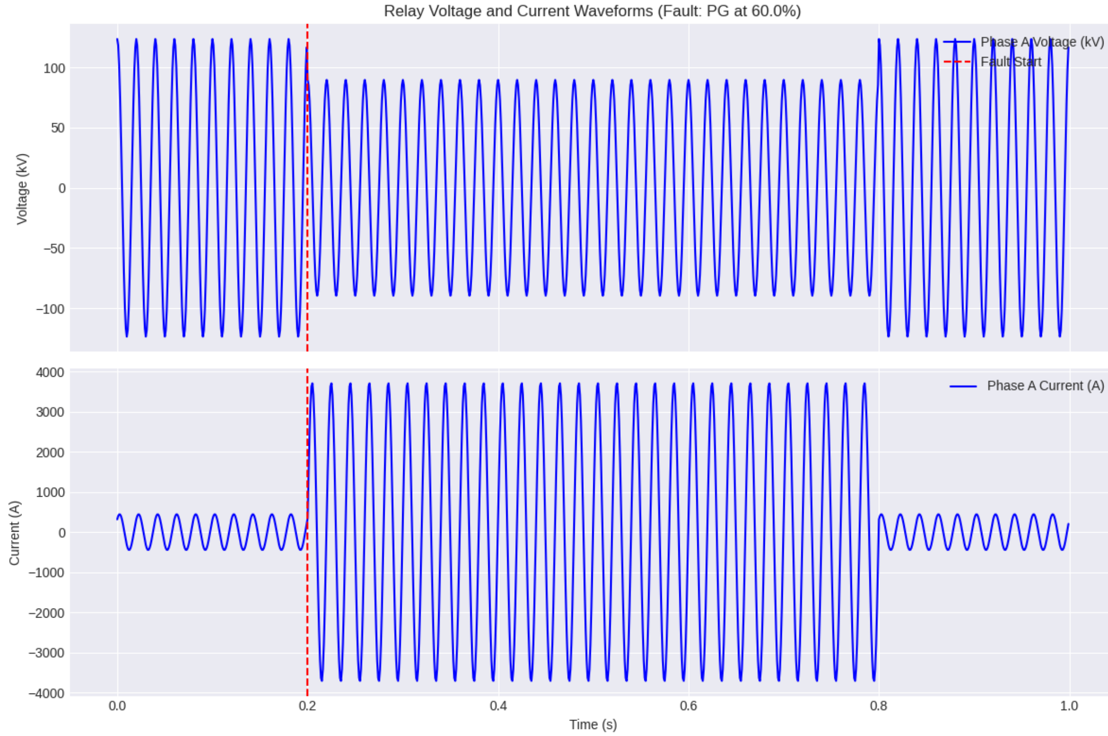

Figure 5: Voltage and current waveforms during fault

- Fault: Immediate shift toward line impedance
- Trip occurs when impedance enters Zone 1 circle

6 Conclusion

The simulation successfully demonstrates:

- Accurate phasor estimation using DFT with 1-cycle window
- Proper impedance calculation for different fault types
- Correct operation of three-zone mho characteristic relay
- Visualization of dynamic system behavior during faults

Recommendations

Future enhancements could include:

- Implementation of full-cycle and half-cycle DFT algorithms
- Adding transient detection logic

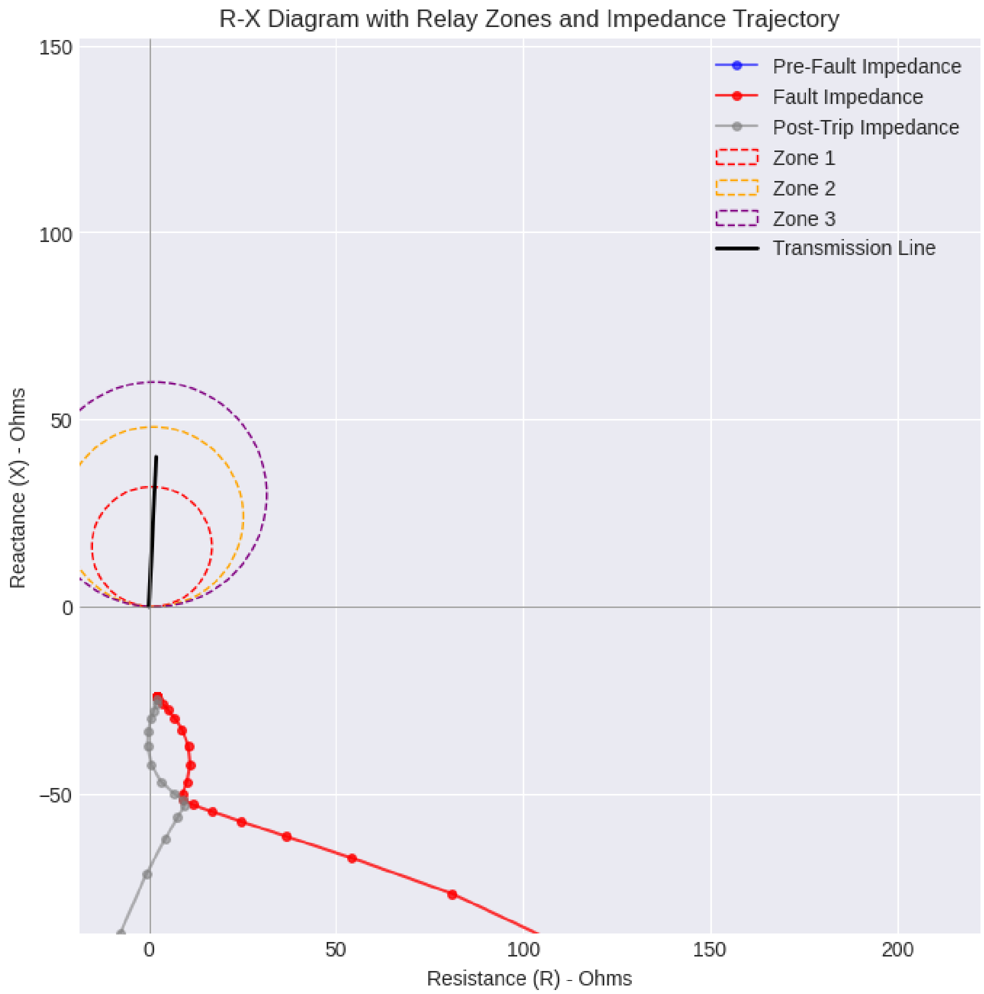

Figure 6: R-X diagram with impedance trajectory

- Implementing communication-assisted protection schemes
- Adding more fault types (three-phase, evolving faults)

Appendix: Complete Code Structure

```
distance_relay_simulation/

 main.py                # Main simulation script
 modules/
    system_model.py     # Power system modeling
    fault_model.py      # Fault calculations
    relay_algorithm.py  # Protection logic
    visualization.py    # Plot generation
 tests/                 # Unit tests
```

```
results/                # Output figures and data
```

References

[1] IEEE Std C37.113-2015, *IEEE Guide for Protective Relay Applications to Transmission Lines*. IEEE, 2015.

[2] P. M. Anderson, *Power System Protection*. Wiley-IEEE Press, 1999.

[3] A. G. Phadke, J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*. Springer, 2008.