

# Analysis of Different Algorithm of Numerical Relay Using Python

Arpit Sanjay Thakre<sup>1</sup>, Atharv Moon<sup>2</sup>, Arpit Mendhe<sup>3</sup>, Pavan Harle<sup>4</sup>, Om Bobde<sup>5</sup>,

Prof. Rajendra Bhombe<sup>6</sup>, Prof. Priyanka Rajput<sup>7</sup>

Students, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>1</sup>

Students, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>2</sup>

Students, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>3</sup>

Students, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>4</sup>

Students, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>5</sup>

Associate Professor, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>6</sup>

Teacher Assistant, Electrical Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, India<sup>7</sup>

**Abstract:** Numerical relays play a vital role in modern power system protection by providing fast, reliable, and accurate fault detection. Unlike conventional electromechanical and static relays, numerical relays use microprocessors and digital signal processing techniques to analyze electrical parameters and generate trip signals during abnormal conditions. This paper presents a comparative analysis of different algorithms used in numerical relays, implemented using Python. The study focuses on commonly used protection algorithms such as overcurrent, distance, and differential protection techniques. Each algorithm is modeled and tested under various fault conditions including line-to-ground, line-to-line, and three-phase faults.

Key performance parameters such as response time, detection accuracy, sensitivity, and reliability are evaluated. Simulation results demonstrate that digital implementation of relay algorithms improves fault detection speed and accuracy while reducing computational complexity. Among the analyzed techniques, differential protection shows superior performance in terms of fast response and precision, while distance protection provides effective zonebased fault identification. The use of Python enables flexible modeling, easy simulation, and efficient comparison of relay characteristics. The findings of this study contribute to better understanding and practical implementation of numerical relay algorithms in modern smart grid and power system protection applications.

**Keywords:** Numerical Relay, Overcurrent Protection, Distance Protection, Differential Protection, Python, Power System Protection.

## I. INTRODUCTION

Power system protection is an essential part of electrical power networks to ensure safe, reliable, and continuous operation. Faults such as short circuits, line-to-ground faults, and equipment failures can cause severe damage to transmission lines, transformers, and generators if not detected quickly. Protective relays are therefore used to sense abnormal conditions and isolate the faulty section from the rest of the system. Traditionally, electromechanical and static relays were used for protection purposes; however, these relays have limitations in terms of speed, accuracy, flexibility, and maintenance requirements.

With advancements in microprocessor technology and digital signal processing, numerical relays have replaced conventional relays in modern power systems. Numerical relays operate by sampling voltage and current signals, converting them into digital form, and applying mathematical algorithms to detect faults. They offer higher accuracy, faster response time, multifunction capability, and improved reliability. Additionally, numerical relays allow easy modification of protection settings and integration with communication systems in smart grids.

## II. OBJECTIVE

► Implement a set of representative numerical relay algorithms in Python (e.g., impedance-based, differential, directional, adaptive threshold and phasor-based methods).

Validate each algorithm on synthetic and recorded power-system disturbance signals (faults, inrush, transient, noise) to measure correctness and robustness.

► Quantitatively compare algorithms using performance metrics: detection accuracy, false-trip rate, decision time (latency), computational cost, and sensitivity to measurement noise and CT/PT errors.

► Propose and test simple enhancements (filtering, adaptive thresholds, decision fusion) to improve performance where weaknesses are identified.

## III. METHODOLOGY

### 1. Overview

The objective of this project is to implement, compare, and analyze several numerical relay algorithms using Python for fault detection and classification in power systems. The main goals include

- Implement classical and modern numerical relay algorithms such as DFT/Phasor, RMS, Kalman filter, Wavelet transform, and a simple machine learning classifier.
- Generate simulated fault and normal waveforms with controlled noise, harmonics, and varying sampling rates. Evaluate the algorithms based on detection speed, accuracy, false-trip rate, and robustness to disturbances.

### 2. Tools and Environment

- Programming language: Python 3.10+
- Libraries: numpy, scipy, pandas, matplotlib, pywt (Py Wavelets), scikit-learn, numba (optional for speed), jupyter for experiments.
- Optional: Simulink/PSCAD for realistic signal simulation, otherwise synthetic signal generation in Python.

### 3. Data Generation and Preprocessing

- Signal Generation
- Three-phase voltage and current signals (50/60 Hz) will be generated with adjustable parameters:
- Fault types: Single line-to-ground (SLG), Line-to-line (LL), Double line-to-ground (DLG), Three-phase (LLL)
- Fault parameters: Inception angle, fault resistance, fault location.
- Sampling rates: 1 kHz, 4 kHz, and 10 kHz.
- Disturbances: Gaussian noise (SNR: 40dB, 20dB, 0dB), harmonics (5th, 7th), DC offset, CT saturation.
- Each time-series window will be labeled as Normal, SLG, LL, DLG, or LLL

### 4. Algorithms to Implement

- DFT/Phasor Estimation
- Fundamental frequency phasor calculation using Discrete Fourier Transform
- Positive, negative, and zero sequence components will be used for fault classification
- Fault detection based on threshold ratios of sequence components.

Pseudocode: for each window:

```
X=fft(window)
VI=extract_bin(X, fundamental_bin)
V2=compute_negative_sequence(phasors)  VO=compute_zero_sequence(phasors)  if  IV2/V1>thresh2  or
[V0/[VI]>thresh:
flag_fault()
classify_by_sequence_ratios()
RMS and Moving-Window RMS
RMS computed over sliding windows faults detected when step changes exceed thresholds.
```

### 5. Experimental Design and Testing Matrix

- Scenarios
- Normal (no fault)

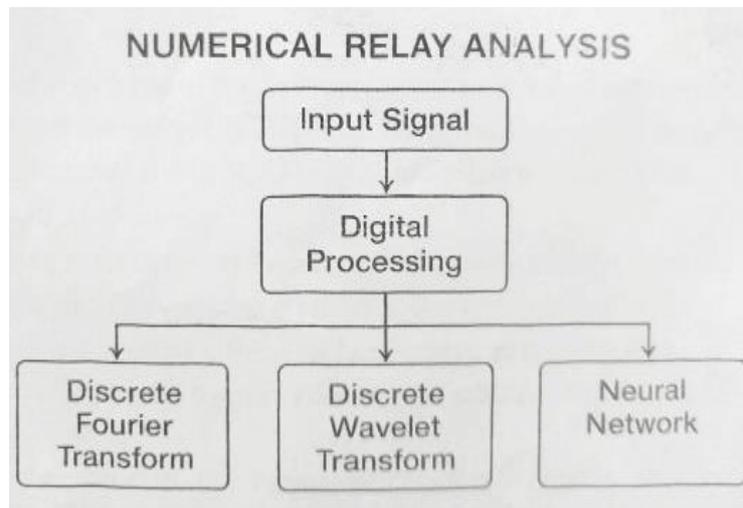
- Faults: SLG, LL, DLG, LLL at multiple inception angles and resistances
- SNR levels: 40, 20, 10, 0 dB.

**6. Evaluation and Analysis**

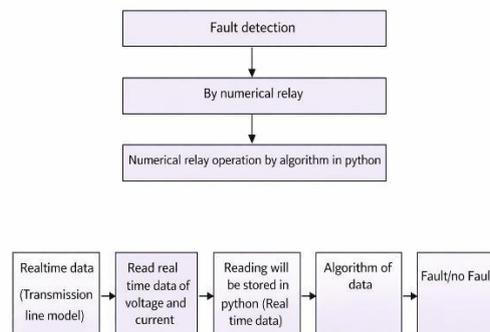
- Results presented via tables, line plots, bar charts, and confusion matrices
- Trade-offs: speed vs accuracy, robustness vs computational cost
- Optional Statistical tests (eg, paired t-test) to compare algorithms.

**7. Deliverables**

- Python code modules
- signal generator.py, preprocess.py, dtj\_relay.py, kalmars\_relay.py, wavelet\_relay.py, sml\_relay.py, evaluate.py
- Synthetic dataset with labels
- Report including methodology, results, discussion, and conclusions
- Presentation slides summarizing findings

**BLOCK DIAGRAM**

Flowchart for python-based implementation of Mann &amp; Morrison Method

**IV. CONCLUSION**

This study analyzed different numerical relay algorithms using Python. The results show that numerical relays provide better speed, accuracy, and flexibility compared to conventional relays. Among the analyzed methods, differential protection performed the best, while distance protection was effective for transmission lines. Python-based implementation proved to be simple and efficient for analyzing relay performance.

**REFERENCES**

- [1]. T. Bujanovic, P. Ghosh, "Adaptive Algorithm for Microprocessor Based Distance Relays in Smart Grid may 2016 IEEE International Conference
- [2]. H. panahi, R. samani and M. sanaye-pasand, "Advances in transmission network fault location in modern power systems: review, outlook and future works", December, 2021 Jaccess
- [3]. M. Ojaghi and R. Ghahremani, "Piece-wise Linear Characteristic for Coordinating Numerical Overcurrent Relays", 0885-8977 (c) IEEE 2016
- [4]. M. N. Alam, "Adaptive Protection Coordination Scheme using Numerical Directional Overcurrent Relays", IEEE Trans
- [5]. D. S. Kumar, D. Srinivasan, "A Numerical Protection Strategy for Medium-Voltage Distribution Systems", IEEE 2018
- [6]. Books Reference: P. Kundur, Power System Stability and Control, McGraw-Hill, 1994.
- [7]. G. I. Rogers, Power System Protection and Switchgear, 3rd Edition, CRC Press, 2012