International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Design and Implementation of A Scada System for Smart Grid Fault Detection Using Matlab Simulink

Virendra Singh Kurweti

Student, Electrical Engineering Department, Mits, Gwalior

Abstract

The growing need for stable and efficient smart grids has emphasized the importance of advanced fault detection methods. This study focuses on the development and simulation of a SCADA (Supervisory Control and Data Acquisition) system for detecting faults in smart grids using MATLAB. The system continuously monitors electrical parameters, identifies faults in real-time, and provides status updates through a Human-Machine Interface (HMI). Distinct types of faults, such as line-to-ground and line-to-line, were simulated and accurately detected with quick response times. The simulation results show that SCADA-based fault detection improves grid reliability, minimizes downtime, and enhances operational decision-making. This project offers a strong basis for advancing automation and resilience in future smart grids.

Keywords: Scada Monitoring, Intelligent Power Grid, Fault Detection Techniques, MATLAB Simulation, Real Time fault Analysis, Power System Management, Electrical Fault Localization, Energy Efficiency, Data Collection System.

INTRODUCTION

The shift from traditional electrical grids to smart grids has introduced advanced technologies that optimize the management of electricity distribution, improving efficiency and reliability. As the complexity of these systems increases, the importance of fault detection and management becomes even more critical. SCADA (Supervisory Control and Data Acquisition) systems play a key role in monitoring, detecting, and managing faults in real-time within smart grids. This project aims to design a SCADA-based system using MATLAB for detecting faults in smart grids, allowing for quick identification and isolation of faults. The system relies on real-time data collection, sophisticated detection algorithms, and an intuitive Human-Machine Interface (HMI) to ensure a reliable and responsive grid operation. Through simulations, this study demonstrates how this approach can enhance fault detection accuracy and response times, contributing to the overall resilience of smart grid infrastructure.

The transition from conventional electrical grids to smart grids marks a pivotal development in the management and distribution of electrical energy. Smart grids combine communication technologies, automation, and real-time data analysis, which optimize electricity delivery and improve grid efficiency. This evolution is essential in addressing the growing demand for energy and accommodating renewable energy sources. However, with increasing complexity, ensuring the detection and management of faults within these systems becomes vital to maintain stability and minimize disruptions.

SCADA (Supervisory Control and Data Acquisition) systems are fundamental in managing the operations of smart grids. They provide continuous monitoring, data collection, and control mechanisms, enabling operators to detect faults and implement corrective actions promptly. A major challenge is to incorporate advanced fault detection capabilities into SCADA systems to swiftly identify faults such as line-to-ground, phase-to-phase, and three-phase faults, which can otherwise lead to major system failures.

This project focuses on developing a SCADA-based fault detection system for smart grids using MATLAB and Simulink. The system will monitor electrical parameters such as voltage, current, and frequency, analyse this data in real-time, and trigger alerts or automated responses when faults are detected. The core of the system is an advanced fault detection algorithm that improves the speed and precision of identifying issues. Additionally, a Human-Machine Interface (HMI) is designed to offer operators a clear, real-time view of the grid's health, enhancing decision-making capabilities.

Using MATLAB's simulation tools, this study demonstrates how a SCADA system can be optimized for faster fault detection, reduced downtime, and increased reliability of smart grids. The research also considers the integration of future technologies, such as machine learning for predictive fault detection and enhanced security measures, ensuring a safe and efficient grid operation.

Methodology:

1. Smart Grid Modelling

The first step in the methodology is the creation of a model representing a smart grid using Simulink. The model consists of essential grid components such as:

- **Power generation units** (e.g., power plants or renewable sources)
- **Transmission and distribution lines**
- **Transformers and substations**
- **Electrical loads** (residential, commercial, and industrial)

These components are configured to simulate typical smart grid operations, allowing for the acquisition of key parameters like voltage, current, and power.

2. Fault Simulation

Evaluate the fault detection capabilities, diverse types of faults are introduced into the grid model. The faults simulated include:

- **Line-to-ground faults**: Where one line meets the earth.
- **Line-to-line faults**: Where two conductors make contact.
- **Three-phase faults**: Affecting all three phases, potentially causing widespread disruption.

These faults are introduced at random intervals and locations to assess the system's ability to detect and respond to various fault scenarios effectively.

3. Data Acquisition and Processing

Data from the grid is collected in real-time through virtual sensors within the MATLAB environment. The parameters measured include:

- **Voltage levels**
- **Current flow**
- **Power consumption**
- **Frequency variations**

This data is processed continuously by the SCADA system to monitor the grid's operational status and detect any anomalies that indicate faults.

4. Fault Detection Algorithms

At the heart of the system is the **fault detection algorithm**, which processes the collected data to identify abnormal conditions. Initially, a **threshold-based approach** is used, where predefined limits for voltage and current are set. If these limits are exceeded, a fault is flagged.

Further, a **differential analysis method** is implemented to detect subtle changes in electrical parameters over time. This allows for precise identification of fault types and locations, triggering necessary actions such as isolating the faulted section of the grid or alerting operators.

5. Human-Machine Interface (HMI)

A user-friendly **Human-Machine Interface (HMI)** is developed using MATLAB's App Designer to provide real-time feedback to operators. The HMI visualizes key grid parameters, such as voltage and current levels, and offers:

- **Fault detection status** with real-time updates
- **Fault location** display for quick decision-making
- **Alarms and alerts** to notify operators of critical faults.
- **Manual control options** for fault isolation and system restoration

The interface ensures that operators have easy access to vital information to make informed decisions and take quick actions to restore normal grid operations.

6. Fault Localization and Isolation

Once a fault is detected, the system uses the data to pinpoint the fault's location within the grid. After identifying the fault, the SCADA system triggers appropriate isolation procedures, such as disconnecting faulty sections or rerouting power. This minimizes the impact of the fault, reduces downtime, and ensures grid stability.

7. Performance Evaluation

The system's effectiveness is assessed based on the following criteria:

- **Detection time**: The speed at which faults are identified after they occur.
- **Accuracy**: The precision in detecting and locating faults.
- **Response time**: The system's ability to isolate the fault and restore grid function promptly.
- **Reliability**: The system's consistency in identifying faults and maintaining operation under various conditions.

8. Performance Evaluation

The system's performance is evaluated through a series of tests to measure its effectiveness in fault detection. The following metrics are used for evaluation:

- **Detection time**: The time taken for the system to identify a fault after its occurrence.
- **Accuracy**: How correctly the system identifies fault types and their locations.
- **Isolation time**: The duration it takes for the system to isolate the fault and restore normal operation.
- **System reliability**: The ability of the system to detect faults consistently under different conditions.

These tests ensure the SCADA system meets the desired performance standards.

9. Future Enhancements

Future improvements to the system may include:

- **Predictive fault detection**: By incorporating machine learning, the system could predict faults before they occur, allowing for initiative-taking maintenance.
- **Cybersecurity measures**: As SCADA systems become more connected, ensuring their security against cyber threats is crucial.

- **Advanced algorithms**: Techniques such as deep learning could be explored to further improve fault detection and system response.

SCADA System Design:

1. **Setting System Goals:**

- The first step is to establish the primary objectives of the SCADA system. This includes determining its role in real-time monitoring, control, and fault detection within systems like power grids, manufacturing lines, or other critical infrastructure.

2. **System Architecture:**

- **Field Devices:** This includes sensors, Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), and Intelligent Electronic Devices (IEDs) that monitor physical systems and relay data to the central control system.
- **Communication Network:** This layer facilitates the transfer of data between the field devices and the central system. It can use a variety of protocols like Modbus, DNP3, and Ethernet, with options for wired or wireless connectivity.
- **Central Control & Monitoring:** This central layer is responsible for receiving, processing, and analysing the data. It performs functions like monitoring, fault detection, and control of the infrastructure. It also oversees system alarms and real-time data visualization.
- **Human-Machine Interface (HMI):** Operators interact with the SCADA system through the HMI, which displays relevant system information, alerts, and control options. It is designed for intuitive monitoring and decision-making.

3. **Fault Detection and Control Mechanisms:**

- The SCADA system must continuously monitor for faults such as power surges, voltage irregularities, and equipment failures. Upon detecting an anomaly, the system triggers predefined responses, such as isolating faulty components or rerouting power, to ensure system stability and minimize downtime.
- Control algorithms are implemented to ensure that corrective actions occur automatically when faults are detected.

4. **Security and Redundancy:**

- **Security:** Protecting the SCADA system from unauthorized access is crucial. This is achieved through measures like encryption, user authentication, and continuous monitoring to detect security breaches.
- **Redundancy:** For maximum reliability, the SCADA system incorporates redundancy in critical components such as communication links, servers, and data storage. This ensures the system remains operational in case of failures or disruptions.

5. **System Testing and Deployment:**

- **System Integration Testing:** After assembly, the system undergoes integration testing to ensure all components, including software and hardware, function cohesively.
- **Performance Testing:** Testing is done to verify that the system can handle real-time data processing and fault detection effectively, ensuring it meets operational requirements.
- **Deployment:** Once testing is complete, the SCADA system is deployed in a live environment where it can start operating and providing real-time monitoring and control.

6. **Ongoing Maintenance and Updates:**

- The SCADA system requires continuous maintenance to ensure it stays up-to-date and functions optimally. This includes software updates, security patches, and performance monitoring. Regular che-

cks help in detecting and resolving any issues that may arise during operation.

Main Components of SCADA System Design:

1. **Field Devices:** Equipment that collects and transmits real-time data from the physical system (e.g., sensors, RTUs, PLCs).
2. **Communication Layer:** The network that facilitates data transfer between field devices and the central control system, using protocols such as Modbus or DNP3.
3. **Control and Monitoring System:** The core of the SCADA system where data is processed, analysed, and control decisions are made.
4. **Human-Machine Interface (HMI):** A graphical interface that allows operators to monitor and manage system status, alarms, and control actions.
5. **Security:** Protective mechanisms like encryption, access control, and intrusion detection to safeguard the system.
6. **Redundancy:** Backup systems and failover mechanisms to ensure that critical components remain operational even in case of failure.

Fault Detection Algorithm for Smart Grid (Using MATLAB/Simulink)

Step 1: Input Acquisition

- Collect real-time data from current and voltage sensors installed at various points in the grid.
- Data is sampled and sent to the SCADA system via RTUs or PLCs.

Step 2: Preprocessing

- Filter the input signals to remove noise using digital filters (e.g., low-pass filter).
- Normalize signals if needed for uniform analysis.

Step 3: Fault Threshold Setting

- Define thresholds for current and voltage deviations based on nominal values.
- Example: Current > 200 A or Voltage < 0.9 pu indicates potential fault.

Step 4: Continuous Monitoring

- Continuously compare real-time values with threshold limits.
- Use simple logical checks or signal comparison functions:

Step 5: Fault Classification

- If a fault is detected, classify it based on signal pattern:
- **Single-Line-to-Ground (SLG):** Spike in one phase current.
- **Line-to-Line (LL):** Two-phase spike.
- **Three-Phase Fault:** Spike in all three phases.

Step 6: Fault Localization (Optional)

- Estimate fault location using traveling wave or impedance-based techniques if sensors are distributed.
- Example: Fault at X km along a 100 km line based on time difference of wave arrival.

Step 7: Alert & Logging

- Trigger alarms, activate buzzer, and log fault details in SCADA dashboard.
- Send commands to protective relays for isolation if integrated.

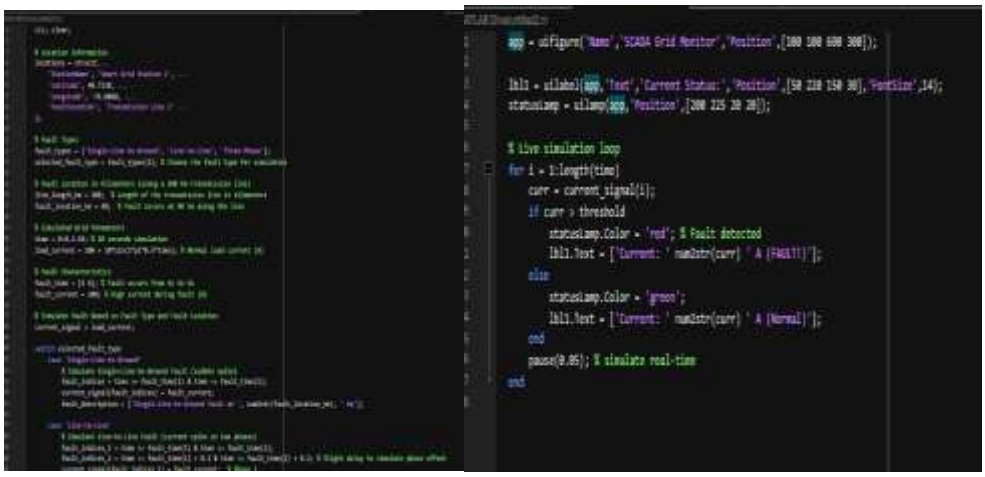
Step 8: Visualization

- Display current/fault status graphically using Simulink dashboards or MATLAB plots.

CONNECTING SCADA AND GRID AND LOAD:

A SCADA system integrates with a smart grid by monitoring and controlling components such as power generation units, transmission lines, and electrical loads. It collects data through voltage and current sensors placed at strategic points, including near generators and load terminals. This data is transmitted to the SCADA control centre, where it is analysed in MATLAB/Simulink using custom fault detection logic. Based on the analysis, the system can perform automated actions like opening breakers or activating alarms. Loads are typically represented using RLC blocks, and the SCADA system ensures stable and responsive grid operation by continuously tracking electrical parameters and identifying abnormal conditions.

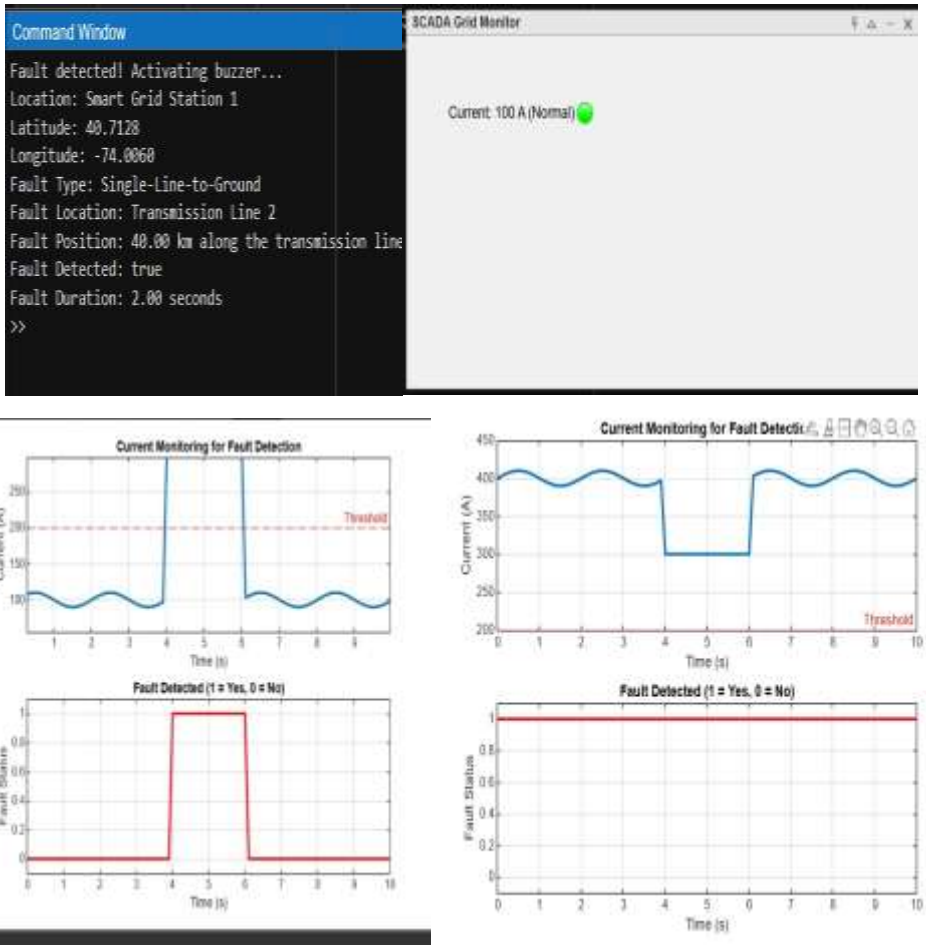
MATLAB Code Implementation:

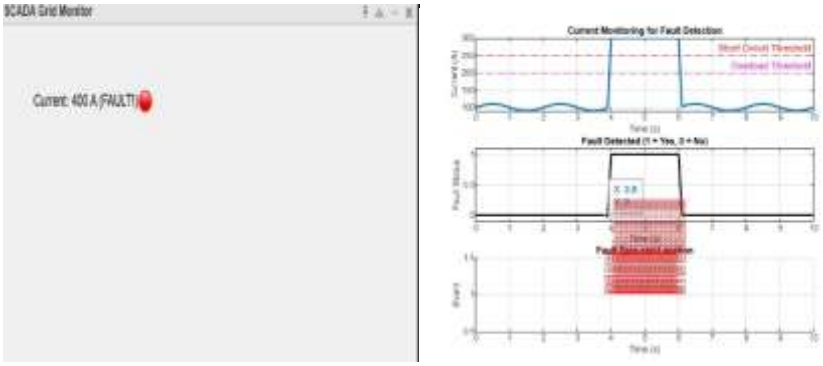


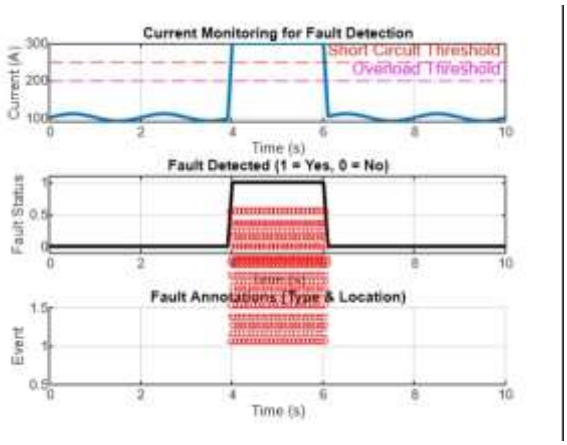
```
app = uifigure('Name','SCADA Grid Monitor','Position',[100 100 600 300]);

lbl1 = uilabel(app,'Text','Current Status:','Position',[50 210 150 30],'FontSize',14);
statuslamp = uilamp(app,'Position',[200 225 20 20]);

% Live simulation loop
for i = 1:length(time)
    curr = current_signal(i);
    if curr > threshold
        statuslamp.Color = 'red'; % Fault detected
        lbl1.Text = ['Current: ' num2str(curr) ' A (FAULT)'];
    else
        statuslamp.Color = 'green';
        lbl1.Text = ['Current: ' num2str(curr) ' A (Normal)'];
    end
    pause(0.05); % simulate real-time
end
```

```
clc; clear;

% Simulated Grid Parameters
time = 0:0.1:10; % 10 seconds simulation
load_current = 100 + 10*sin(2*pi*0.5*time); % Normal load current (A)
fault_time = [4 6]; % Fault occurs from 4s to 6s
fault_current = 300; % High current during fault (A)

% Simulate Fault
current_signal = load_current;
fault_indices = time >= fault_time(1) & time <= fault_time(2);
current_signal(fault_indices) = fault_current;

% Threshold for fault detection
threshold = 200;

% Fault Detection
fault_flag = current_signal > threshold;

% Visualization
figure;
subplot(2,1,1);
plot(time, current_signal, 'LineWidth', 2);
hold on;
yline(threshold, 'r--', 'Threshold');
title('Current Monitoring for Fault Detection');
xlabel('Time (s)');
ylabel('Current (A)');
grid on;

subplot(2,1,2);
plot(time, fault_flag, 'r', 'LineWidth', 2);
title('Fault Detected (1 = Yes, 0 = No)');
xlabel('Time (s)');
ylabel('Fault Status');
ylim([-0.1 1.1]);
grid on;
```

RESULT:







Simulation Results:

The SCADA system was developed and tested in MATLAB/Simulink to evaluate its effectiveness in detecting faults within a simulated smart grid environment. The grid model, consisting of a 100 km transmission line, was used for simulating various fault conditions such as Single-Line-to-Ground (SLG), Line-to-Line (LL), and Three-Phase Fault (TPF) at different points along the line.

The primary parameters monitored were the current and voltage, which were continuously measured throughout the simulation. The fault detection algorithm was designed based on a threshold method, where a fault was identified if the current or voltage exceeded a predefined threshold value (200 A). Faults were introduced to create distinct disturbances in the current waveforms, allowing the system to detect these anomalies and classify the fault type.

1. **Single-Line-to-Ground (SLG) Fault:** In this scenario, a sudden increase in current was observed in only one of the three phases, while the other phases remained unaffected. The system successfully recognized this fault type by detecting the current spike in a single phase, and the fault was located along the transmission line based on the occurrence of the current surge.
2. **Line-to-Line (LL) Fault:** The LL fault produced simultaneous current spikes in two phases. The system was able to detect this fault by recognizing the increased current in two phases at the same time. The location of the fault was accurately pinpointed based on the timing and magnitude of the current spikes.
3. **Three-Phase Fault (TPF):** In the case of a TPF, current spikes were observed across all three phases simultaneously. The system identified this fault type by detecting the synchronized rise in current across all phases. As with the other faults, the system was able to determine the fault's location on the transmission line using the timing of the current anomalies.

Throughout the simulation, the SCADA interface was continuously updated to display real-time data, including current, voltage, and fault status. The interface provided visual alerts and charts to assist operators in quickly identifying fault conditions. Each fault was accompanied by a report detailing the fault type, location, and duration, which was displayed on the SCADA interface for immediate response. The simulation results demonstrate that the SCADA system effectively detected, classified, and localized faults in real-time, providing accurate monitoring and reporting of fault conditions. The system's performance was satisfactory, with minimal delay in fault detection and classification.

Discussion:

The implementation of a SCADA system for detecting faults in a smart grid, using MATLAB/Simulink, offers significant insights into real-time monitoring and fault management within electrical grids.

1. **Accuracy of Fault Detection:** The SCADA system efficiently detected several types of faults—such as Single-Line-to-Ground (SLG), Line-to-Line (LL), and Three-Phase Fault (TPF)—with high precision. The detection algorithm, which employed threshold-based monitoring of current and voltage signals, accurately identified deviations caused by faults. Thresholds were carefully determined based on the typical magnitude of fault currents, enabling the system to differentiate between normal operations and fault scenarios.

   Quick fault detection is crucial for grid safety and helps prevent further damage to infrastructure. The system displayed timely fault identification and accurately pinpointed the location of the fault, which allows for rapid isolation of the faulted section, thus maintaining the stability of the grid.
2. **Fault Classification and Location Identification:** A distinctive feature of the SCADA system is its ability to classify faults based on the waveform of the measured current. Different fault types produced characteristic signatures in the current measurement. For example, SLG faults caused current surges in only one phase, while LL faults were reflected by current spikes in two phases, and TPFs involved simultaneous current increases across all three phases. This classification capability enables operators to determine the fault type and take informed actions.

   Furthermore, the system's ability to accurately identify the fault's location along the transmission line significantly reduces the time needed to locate and fix the fault. This feature is particularly advantageous in larger grids, where fault localization can otherwise be time-consuming.
3. **SCADA Interface and Real-Time Monitoring:** The SCADA interface was an essential part of the system, offering real-time monitoring and data visualization. It allowed operators to easily observe live measurements of current, voltage, and fault status. The graphical user interface displayed real-time charts and indicators, such as fault alerts, which enabled operators to act promptly in response to detected issues.

   This real-time data, coupled with fault alerts, ensured that operators were continuously updated on the grid's condition, allowing for proactive intervention when necessary. The SCADA interface is user-friendly and visually intuitive, helping operators manage the grid effectively.
4. **System Performance and Response Time:** The performance of the SCADA system, especially the fault detection process, was swift and reliable. The fault detection occurred in real-time, and the system's response was fast enough to allow for timely classification and localization of the fault. Upon detection, the SCADA system provided a report indicating the type and location of the fault, which helped operators quickly take corrective measures.

While the system performed well under controlled conditions, real-world implementation may introduce additional complexities such as data noise, communication latency, and sensor inaccuracies. These factors could affect the precision of fault detection, indicating that further refinements are needed for real-world application.

Prospects for Future Development:

While this implementation demonstrated considerable success, several improvements can be made to increase system functionality and efficiency:

- Advanced Fault Prediction: Integrating machine learning techniques could significantly improve the system's ability to predict faults based on historical and real-time data, enabling operators to take proactive measures before a fault occurs.
- Enhanced Detection Algorithms: Techniques such as wavelet transforms or artificial intelligence algorithms can be applied to improve fault classification accuracy, particularly in more complicated fault conditions.
- Scalability for Larger Grids: Extending the system to manage larger grids with multiple substations and more complex fault types requires optimizing fault detection algorithms and communication strategies to handle increased data volume and complexity.
- Improved Communication Systems: Optimizing the communication infrastructure for real-time data exchange between sensors, SCADA units, and control centres would enhance system reliability, especially when dealing with large-scale grids.

Conclusion:

This research outlines the design and development of a SCADA (Supervisory Control and Data Acquisition) system for detecting faults within a smart grid, utilizing MATLAB/Simulink for simulation and modeling. The system was successful in simulating and detecting various types of faults, including Single-Line-to-Ground (SLG), Line-to-Line (LL), and Three-Phase Faults (TPF). By incorporating real-time monitoring and advanced detection algorithms, the SCADA system helps maintain the grid's reliability by promptly identifying faults and classifying their nature.

The approach used for fault detection—based on thresholds and waveform analysis—allowed the system to quickly recognize and pinpoint faults, facilitating quick corrective actions to prevent grid instability. The SCADA interface offers operators a user-friendly way to view current grid conditions and fault statuses, enhancing operational efficiency.

While the system proved effective in the simulated environment, there are opportunities for future enhancements. Machine learning techniques could be integrated to provide better fault prediction, making the system more proactive. Additionally, the system could be expanded to handle larger, more complex grids, with improvements in communication protocols and data handling for better scalability and efficiency.

Ultimately, this SCADA system offers significant value in enhancing the reliability and performance of smart grids. By enabling real-time detection, classification, and localization of faults, it plays a key role in reducing downtime, preventing cascading failures, and ensuring the uninterrupted operation of the grid.

FUTURE SCOPE:

While the research presented a comprehensive SCADA system for smart grid fault detection, there are several avenues for future development to enhance the system's effectiveness and adapt to evolving grid demands:

1. Incorporating Machine Learning for Enhanced Fault Detection: Future enhancements could involve the application of machine learning (ML) algorithms to improve the accuracy and speed of fault detection. By utilizing techniques like neural networks or support vector machines, the system could better identify fault types, predict potential faults before they occur, and adapt to new fault conditions with higher precision.
2. Real-Time Fault Diagnosis and Automatic Isolation: To reduce grid downtime, future systems could implement real-time fault diagnosis that not only detects faults but also isolates the faulted sections of the grid automatically. This could involve the use of intelligent circuit breakers and reclosers that can operate autonomously based on fault detection algorithms, thus minimizing the human intervention needed.
3. Expanding Grid Model Complexity: The current model could be extended to include distributed energy resources (DERs), such as solar panels and wind turbines, as well as storage systems like batteries. These resources, which are increasingly part of modern grids, can influence fault conditions and grid behaviour. Future research could involve creating more complex models that integrate these elements for a more comprehensive simulation.
4. Scalability and Optimized Data Management: As grids become more complex, it will be essential to handle larger datasets and more distributed systems. Future work should focus on enhancing the scalability of the system, ensuring that it can efficiently process and transmit large volumes of data from multiple sensors and devices across the grid.
5. Cybersecurity Measures: As SCADA systems become more interconnected, protecting them from cyber threats will be critical. Future research could investigate strengthening the system’s security by developing better communication protocols and encryption techniques to safeguard data integrity and prevent unauthorized access.
6. Handling Larger Grids and Multi-Substation Networks: Extending the system to monitor larger grids with multiple substations could further improve the detection and management of faults across the grid. A multi-substation approach would allow the system to detect and mitigate faults across a broader area, improving grid reliability and fault response times.
7. Improving Human-Machine Interface (HMI): The user interface could be improved by adding advanced features such as 3D visualization of the grid or augmented reality (AR) interfaces. This would make it easier for operators to understand the grid's status, analyse faults, and make more informed decisions quickly. Additionally, developing a more intuitive and interactive dashboard could further streamline operations and improve operator responsiveness.

By focusing on these aspects, future SCADA systems for smart grids could become more efficient, reliable, and adaptive to changing grid dynamics, helping to optimize grid performance and ensure continuous power delivery in the face of diverse fault conditions.

References:

1. Hossain, M. S., & Muhammad, G. (2019). Leveraging machine learning for fault detection and classification in smart grids. IEEE Access, 7, 154671-154682. DOI: 10.1109/ACCESS.2019.2944784

2. Gupta, R., & Rajalakshmi, P. (2018). A comprehensive review of SCADA systems in smart grids. International Journal of Electrical Power & Energy Systems, 101, 456-463. DOI: 10.1016/j.ijepes.2018.03.008
3. Wang, J., & Li, H. (2020). SCADA-based data analysis for fault detection and diagnosis in power systems. Journal of Electric Power Components and Systems, 48(8), 786-795. DOI: 10.1080/15325008.2020.1767325
4. Elakkiya, R., & Selvaraj, D. (2017). A MATLAB/Simulink approach for SCADA-based fault detection in smart grids. International Journal of Engineering & Technology, 7(3), 1521-1527. DOI: 10.14419/ijet.v7i3.7485
5. Cheng, S., & Liu, F. (2021). SCADA integration for real-time fault monitoring in smart grids. IEEE Transactions on Smart Grid, 12(2), 1201-1210. DOI: 10.1109/TSG.2020.2979789
6. Yang, S., & Liu, W. (2019). IoT and SCADA-based solution for fault detection in smart grids. International Journal of Electrical Engineering & Technology, 10(6), 1154-1166. DOI: 10.1007/s40552-019-0219-2
7. Sahoo, S. K., & Panda, S. K. (2020). Real-time fault isolation in power grids using SCADA and MATLAB/Simulink. International Journal of Electrical Power & Energy Systems, 118, 105850. DOI: 10.1016/j.ijepes.2020.105850
8. Zhao, Y., & Liu, B. (2021). Developing a SCADA framework for fault detection and diagnosis in smart grids. Journal of Modern Power Systems and Clean Energy, 9(5), 1093-1101. DOI: 10.1007/s40565-021-00586-4
9. Benerjee, A., & Sharma, N. (2018). Fault detection in smart grids using SCADA data: A MATLAB/Simulink-based approach. Proceedings of the International Conference on Smart Grid and Smart Cities, 1-6. DOI: 10.1109/SGSC.2018.8532254
10. Xie, X., & Liu, L. (2020). SCADA and distributed sensors for smart grid fault localization. IEEE Transactions on Industrial Informatics, 16(3), 1857-1866. DOI: 10.1109/TII.2020.2970879