Multiagent System-Based Adaptive Numerical Relay Design and Development: Part I - Firmware

Abdulfetah Abdela Shobole[1], Motuma Abafogi[1,*], Hammad Khalid[2], Yahia Amireh[1], Abdurrahman Zaim[1]
[1]*Department of Electrical and Electronics Engineering, Istanbul Sabahattin Zaim University, Halkali Mahallesi, Istanbul, Turkey*
[2]*Department of Computer Engineering, Koc University, Sariyer, Istanbul, Turkey*
*abdulfetah.shobole@izu.edu.tr; *motuma.abafogi@izu.edu.tr ; hkhalid22@ku.edu.tr; amireh.yahia@std.izu.edu.tr; ahmed.a@std.izu.edu.tr*

***Abstract*—Protection relays that incorporate advanced microprocessors are vital to electrical grids, providing fast and reliable responses to faulty conditions using efficient communication protocols. Instant detection and response to various faults are essential to minimise the risk of damage. Numerical relays can identify faulty conditions and trigger circuit breakers to open, thus preventing further damage to the system. Due to the lack of autonomous decision-making capabilities, existing numerical relays require manual reconfiguration in situations such as a change in network configuration and protection settings. These relays also do not have the ability to coordinate fault clearance when multiple sources supply power to the grid. A comprehensive overview of the research aimed at developing a multiagent system (MAS)-based adaptive protection relay will be provided by dividing it into separate articles such as firmware and hardware. This first part delves into the firmware aspects of this innovative relay, highlighting its adaptive capabilities and key considerations in its development. The second part will provide detailed design descriptions for the hardware features of the relay. An STM32MPU multicore advanced microprocessor is utilised to design and develop the adaptive numerical relay firmware. It incorporates several protection relay ANSI codes, communication protocols, and MAS-based adaptive protection schemes as part of the firmware.**

***Index Terms*—Adaptive protection; IEC 61850; Multiagent system; Power system protection.**

I. INTRODUCTION

Numerical relays are primarily tasked with monitoring and handling undesirable and dangerous faulty situations that might occur in electrical substations. These microprocessor-based relays possessing various features and capabilities can effectively detect and handle faults in the power system for various protection features such as overcurrent, overvoltage, undervoltage, distance protection, differential protection, and more. [1] To facilitate efficient data exchange among protection devices and remote monitoring systems within electrical substations, several communication protocols have been developed, such as Modbus, DNP3, IEC 60870, and the IEC 61850 standard [2].

The development of multifunction numerical relays using advanced processors has been provided in [3], [4]. A field-programmable gate array (FPGA) based multifunctional protection relay is implemented in [5], [6], featuring versatile capabilities. The design and implementation of two protection relays capable of handling fault conditions, such as overcurrent, is presented in [7], utilising both FPGA and microcontroller platforms. In [8], a multifunction digital protection relay that incorporates various protection schemes, including overcurrent, overvoltage, undervoltage, overfrequency, and underfrequency is introduced.

The protection scheme introduced in [9] aims to redefine the overcurrent protection relay settings by incorporating an adaptive time multiplier setting (TMS) that is determined by multiplying the acceleration factor (AF) and the tuning factor (TF). The effectiveness of this approach has been demonstrated in optimising the response of overcurrent relays and minimising the overall operating time. The research presented in [10] proposes a hybrid optimisation algorithm, combining the Firefly Algorithm and Linear Programming, to address the complex and nonlinear problem of coordination and settings of directional overcurrent relays in power system networks. The proposed method effectively reduces the total relay operating time by relaxing the search space and attaining optimal solutions, as demonstrated through testing on IEEE 8, 15, and 30-bus test systems. A current and voltage-based protection algorithm, presented in [11], exhibits accurate fault current detection capabilities within solar in-feed distribution systems, and its effectiveness has been verified using the IEEE 30-bus reference distribution system. A fault detection and classification scheme for electrical networks based on estimated Euclidean distances of successive samples of the actuating signal using magnitudes of three-phase current phasors computed by a discrete Fourier transform is presented in [12].

The research work presented in [13] aims to establish a unified platform that facilitates communication between devices operating on different data transfer standards, while the authors in [14], [15] primarily examine the interoperability of IEC 61850-based devices produced by different manufacturers. In [16], a demonstrative analysis of

Manuscript received 2 May, 2023; accepted 29 September, 2023.
This work is supported by the Scientific and Technological Research Council of Turkiye (TUBITAK) within the scope of TUBITAK 3501 - Career Development Program under Grant No. 3501-121E069.

the application of the IEC 61850 generic object-oriented substation event (GOOSE) publisher and subscriber to transmit time-critical data between intelligent electronic devices (IEDs) within a local area network (LAN) is provided. The findings in [17] show a wireless LAN implementation of the IEC 61850 manufacturing message specification (MMS) protocol to facilitate data exchange among IEDs in a given electrical substation. The authors in [18] explore the implementation of the line differential protection function using IEC 61850 sampled value (SV) and GOOSE communication protocols. In addition, in [19], data acquisition cards and IEC 61850-based real-time simulators for protection relays are compared.

Adaptive protection has emerged as the leading and evolving method for power system protection in recent times [20]–[24]. A relay and generation agent-based multiagent system protection technique for coordinating overcurrent relays when there are multiple power sources is provided in [25]. The analysis in [26] discusses the use of multiagent systems to increase the transient stability of the electrical grid, while the work in [27] proposes a dual relay and configuration agent-based fault analysis and management for smart distribution networks. On the other hand, the protection scheme proposed in [28] discusses a multicategorical approach to fault elimination in distribution networks. A MAS architecture is explored for smart distribution networks in [29]–[31], while, in [32], a decentralised adaptive protection system design incorporating intelligent protection relays is introduced. A multiagent system is proposed in [33], offering a comprehensive solution for coordination and protection of the power grid. This adaptive protection scheme is further demonstrated in [34], providing evidence of its effectiveness and functionality.

In existing numerical relays, changes in network configuration or protection settings require manual reconfiguration of each relay. These relays also lack the ability to coordinate fault identification and isolation in the presence of multiple power sources. There are no significant resources in the literature regarding developing a fully functional adaptive protection relay based on multiagent systems. In addition to providing a thorough description of the fundamental design and development features of the adaptive protection relay, this article concerning the firmware aspect of the adaptive protection relay and the subsequent second article on the hardware parts will attempt to fill this void in the literature.

This article is structured as follows. The next section provides a comprehensive description of the adaptive multiagent system-based coordination processes during network configuration changes and fault occurrences. Subsequently, a detailed overview of the firmware structure of the designed numerical relay is presented. Finally, a practical demonstration section shows the effective interaction among these agents in the adaptive protection scheme.

II. ADAPTIVE MULTIAGENT SYSTEM-BASED COORDINATION

The adaptive protection scheme based on a Multiagent System (MAS) involves the collaboration of multiple agents working together to make crucial decisions that aim to improve the overall protection of the power system. The research work presented in [33] describes an adaptive protection approach that involves multiple agents with specific roles. The Setting Calculation Agent (SCA), Short-Circuit Current Calculation Agent (SCCA), Load Flow Calculation Agent (LFCA), and Network Configuration and Mapping Agent (NCMA) are tasked with managing network configuration changes. The Logic and Operating Agent (LOA) and Coordination and Optimisation Agent (COA) handle both faults and network configuration changes, while the Device Agent (DA) and Communication Agent (CA) serve as interfaces connecting the system to the external world. The interaction between these agents during changes in network configuration and fault occurrences is shown in Fig. 1 [33], [34].

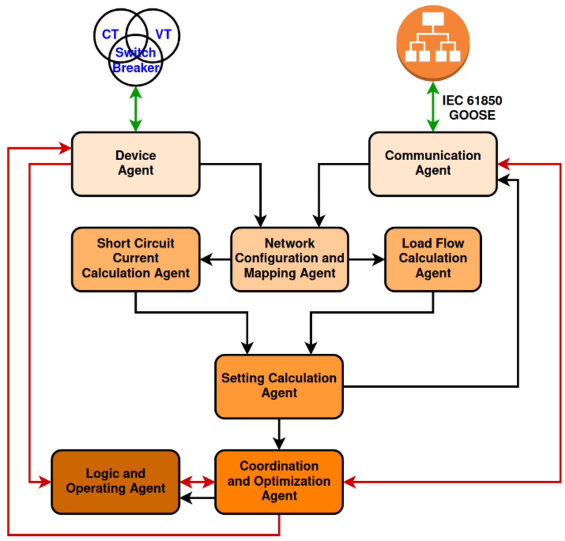

Fig. 1. MAS-based coordination.

In Fig. 1, the black lines illustrate the collaboration of multiple agents during changes in network configuration, such as additions or removals of loads and sources. Agents calculate the required adjustments in operating times and protection parameters, which are then communicated to higher level relays using protocols such as IEC 61850 GOOSE. The CA provides an interface to other numerical relays with such adaptive features. At the same time, the DA monitors network configuration change signals from switches or circuit breakers (CBs). The main task of NCMA is to monitor the network configuration for any changes via CA and DA. It will relay this information to the SCCA and LFCA whenever any change occurs. SCCA performs short-circuit analysis, while LFCA conducts load flow analysis. Subsequently, the SCA computes new protection settings based on newly acquired information, which will be relayed to COA and CA. COA will synchronise these changes with LOA while the CA transmits this information to other relays within the network. In addition, LOA implements appropriate logic for operating the relays and CBs, thereby guaranteeing a coordinated and timely response to network configuration changes.

The red lines in Fig. 1 illustrate the coordination among multiple agents during fault occurrence. The analogue inputs acquired via CTs and VTs are continuously monitored for the occurrence of any fault. If a faulty condition occurs, the DA will immediately inform the LOA. Moreover, the COA may receive blocking or follow-me signals via CA. LOA adjusts its logic based on COA signals, devising a trip logic for relay and CB operation. The tripping procedure is coordinated and accomplished by the COA. In a situation where the relay cannot execute the trip operation successfully, DA will automatically notify the LOA agent. As a coordinating agent, the COA will take alternative measures and command the CA agent to alert backup relays, guaranteeing complete fault monitoring.

III. FIRMWARE STRUCTURE OF THE NUMERICAL RELAY

The firmware structure comprises four main sections: Data Acquisition (DAQ) and protection algorithms, Local Human-Machine Interface (HMI), Embedded Database and OS, and Communication protocols, as shown in Fig. 2.

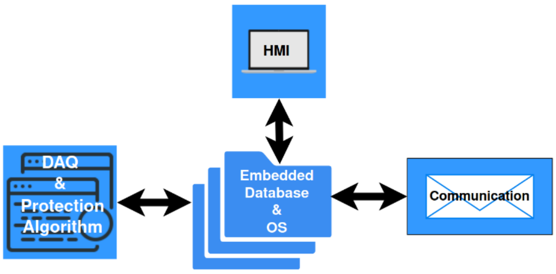

Fig. 2. Numerical relay firmware.

An STM32MPU multicore microprocessor is used to develop the firmware for the numerical relay. In subsequent sections, details of the firmware structure involved in developing the numerical relay are provided.

A. Embedded Database and OS

The STM32MPU microprocessor is equipped with a powerful dual arm cortex-A7 and a cortex-M4, operating at frequencies of up to 800 MHz and 209 MHz, respectively. The cortex-M4 section is the primary processor responsible for capturing real-time data from current and voltage transformers. At the same time, the dual cortex-A7 handles various other tasks, such as HMI operations and communication with other relays or supervisory control and data acquisition (SCADA) systems, and more. An embedded Linux OS has been deployed and is accessible through the cortex A7 core. The MCU SRAM internal memory is used to realise interprocess communication between cortex A7 and M4.

A flat file database is utilised to store essential parameters such as protection settings and communication parameters. It is a type of storage system that is used to store data in a simple and direct manner. A flat-file database can be a practical choice due to its simplicity and less complex implementation. The binary format is often employed to store these parameters efficiently, as it allows for compact representation and faster read/write operations. A file-lock mechanism is implemented to prevent concurrent processes that access database files from corrupting them. Corruption can happen either because a file being read by one process is written by another, or vice versa. The file-lock system ensures that if such a scenario occurs, the process desiring access to a file that is already being accessed by another process will have to wait until it is done. The standardised POSIX record lock (fcntl) provided in UNIX systems is utilised to implement a file-lock. When trying to obtain a lock for a file that is already locked by another process, fcntl is set so that it waits until the other process releases the lock. The flock structure is set to be a read lock for read operations, where the file is accessible by other reading processes but not writing ones. For write operations, the flock structure is set to be a write lock, where the file is locked from both the writing and reading processes.

B. Data Acquisition and Protection Algorithm

The protection algorithm function is a crucial component implemented within the firmware of the numerical relay. Protection schemes can be either nondirectional, where the power flow direction does not matter, or directional, where the relays are configured to respond to faults in specified directions. Figure 3 shows a basic description of the steps involved in the protection algorithms.

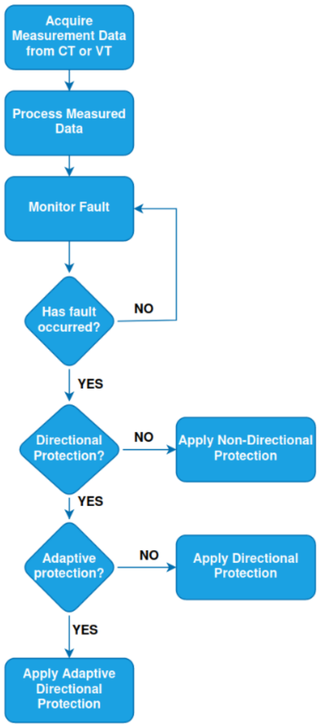

Fig. 3. DAQ and protection algorithm.

Eight analogue inputs are given to the STM32MPU processor, which comprises four current transformer (CT) inputs and four voltage transformer (VT) inputs. The inputs are then sampled and converted using the processor's analogue-digital converter (ADC). Direct memory access (DMA) is used to transfer the converted samples into the memory for efficient processor usage. After the real-time values have been acquired, a fast Fourier transform is applied to extract the magnitudes and phases from the measured data. In this work, 32 data samples per cycle are utilised, resulting in the effective extraction of 15 harmonics. Samples per cycle are achieved by configuring one of the processor's timers to send a trigger signal to the ADC to set off the conversion. The timer can be set to count for the time interval between each sample and sends the trigger signal each time its counter is filled. The three protection schemes implemented in this work are presented as follows.

– For nondirectional protection, the measured magnitude values are compared with preset thresholds. Values beyond the threshold values will initiate a trip sequence.

– For directional protection, if the measured magnitude exceeds the preset threshold values, the next step involves determining the fault direction to distinguish between the forward and reverse directions. The system will initiate a trip sequence if the directionality requirement is satisfied.

– The adaptive version of the directional protection involves the interaction among multiple agents described in Section II to provide a coordinated response to faults and changes in network configuration.

Equation (1) represents the commonly used formula for the inverse time characteristics. The operating time for the numerical relay can be determined by setting the values of α and β according to the chosen characteristic curve

$$t = \frac{TMS \times \beta}{\left(\frac{I}{I_p}\right)^{\alpha} - 1}, \quad (1)$$

where TMS is the time multiplier setting, t is the operating time, I is the measured current value, I_p is the pickup current value, α is the curve type constant, and β is the curve type constant.

This work has implemented several ANSI standards as part of the protection algorithm. Protection functions implemented include phase overcurrent protection (ANSI 50/51), residual overcurrent protection (ANSI 50N/51N), overvoltage protection (ANSI 59), and undervoltage protection (ANSI 27). A concise description of some of these protection functions is provided as follows.

Excessive currents can cause massive damage to the electrical grid. To isolate the faulty zone from the rest of the grid, the phase overcurrent protection (ANSI 50/51) function is implemented. This protection function is intended to compare measured current values with some predefined threshold values. Whenever this threshold value is exceeded, the trip procedure will be initiated.

Undervoltage protection (ANSI 27) is intended to prevent damage to electrical grids that may arise from low-voltage operations. This protection function helps to determine whether the measured voltage has fallen below a preset percentage of the rated voltage value. A trip procedure is initiated if a significant decrease in the measured voltage value is detected.

On the contrary, overvoltage protection (ANSI 59) protects the grid from an overvoltage event that can arise from power sources that generate voltages higher than intended due to a malfunction. The numerical relay monitors the voltage across the electrical grid by comparing it with the threshold value, and upon overvoltage detection, it initiates the trip sequence.

To implement directional protection, the direction of the flow of energy must be determined. Directional protection adds a new layer of decision-making by taking into account the direction of the fault in addition to the magnitude, thereby introducing more selectivity to the protection system. This method is especially useful in particular power system configurations such as parallel feeder and multiple generator configurations. The direction of the energy flow can be determined by utilising various methods. Among these methods, the cross-polarisation approach is utilised in this work. In this approach, the phase angle of the fault vector, which is the faulted phase current/s, is compared with the polarising vector, which is the nonfaulted line-to-line voltage/s. The polarising vector is shifted to 90 degrees minus a preset characteristic angle, and after that the difference in the phase angle between it and the fault vector is extracted. If the resultant value falls within 90 degree range of the polarising vector, the fault is regarded to be in the forward region. Otherwise, it is considered to be in the reverse region. This method is depicted in Fig. 4.

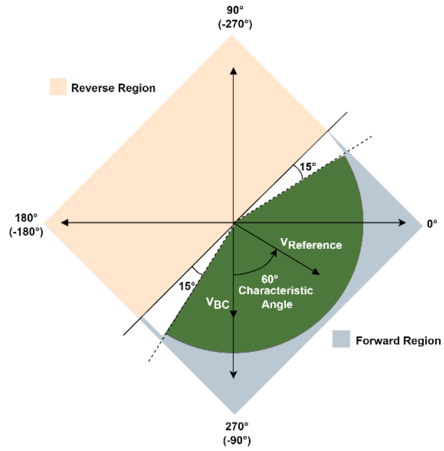

Fig. 4. Determination of directionality using a reference voltage with a characteristic angle of 60 degrees.

C. Communication Protocols

Communication protocols such as IEC 61850, Modbus, DNP3, and IEC 60870 are commonly used in protection relays to ensure efficient and reliable data exchange between devices. Multithreading is a central processing unit (CPU) feature that enables the simultaneous execution of multiple instruction sequences. Figure 5 illustrates the multithreading implementation of communication protocols IEC 61850, Modbus TCP, and IEC 60870-5-104 utilised in this work.

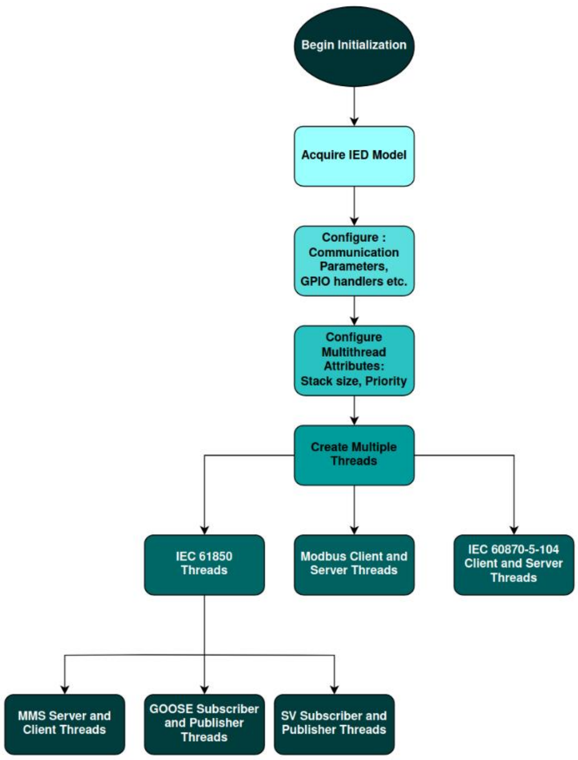

Fig. 5. Implementation of multithreaded communication protocols.

Preemptive multithreading prioritises higher priority threads for execution, allowing them to proceed without waiting for lower priority threads to finish. To implement the communication protocols employed in this work, the initial steps involve importing the IED model, declaring variables, and setting POSIX thread attributes, as shown in Fig. 5.

The multithreaded structure shown in the illustration is implemented using C code and is designed to prioritise time-critical data communication protocols, mainly GOOSE. When multithreading is employed, the system can handle concurrent tasks and effectively manage the demands of various communication protocols.

D. Local HMI

The human-machine interface (HMI) of the numerical relay is utilised to visually present data, monitor various parameters and measurements, configure settings, and perform other related tasks.

A C code for the local HMI is developed to run in the cortex A7 section to manage various tasks such as monitoring and acquiring keypad status, display data, controlling LEDs, etc. The basic structure involved in the initialisation of HMI is shown in Fig. 6.

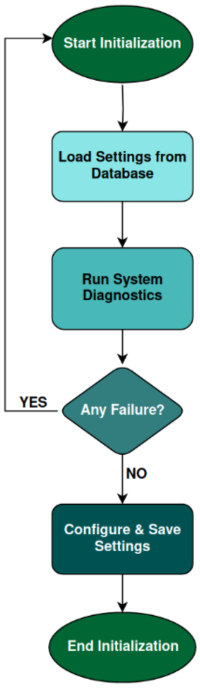

Fig. 6. Local HMI initialisation.

As indicated in Fig. 6, after loading the predefined settings from the database, the relay will run some tests to check whether the relay is functioning correctly. The user can then configure parameters such as protection settings, IP addresses, rated current selection (1A/5A), etc. The code includes an I2C communication protocol functionality via MCP23017 IC to monitor the keypad status. A 128x64 liquid-crystal display (LCD) is programmed to display relay features, such as measured values, protection settings, and more.

IV. Implementation of Adaptive Protection Coordination

Multiple STM32MPU boards were used to showcase coordination among various numerical relays. The adaptive protection schemes in [33] and [34] were utilised to simulate multiagent system-based protection coordination. A modified version of the IEC 61850 GOOSE message structure provided in [34] used in this work is shown in Table I.

TABLE I. GOOSE MESSAGE STRUCTURE BETWEEN LOWER AND HIGHER LEVEL RELAYS.

GOOSE MESSAGE STRUCTURE (Numerical relays along the same direction of energy flow)
* Operating Time
* Coordination Time Interval (CTI)
* NetConfig
* Fault Current
* Blocking Signal

The GOOSE message structure comprises five parameters, as described in Table I. These parameters include the NetConfig status, which indicates the network configuration status, the relay's calculated operating time, the CTI, the fault current (necessary to calculate the TMS in higher level relays) and the blocking signal indicator.

For a given operating time (t_i) received from a downstream relay, the upstream relay uses (2) to determine its time multiplier setting (TMS_{i+1}) [33]

$$TMS_{i+1} = \frac{\frac{t_i + CTI}{\beta}}{\frac{I_i}{\left(I_{pi}+1\right)} - 1}, \tag{2}$$

where CTI is the coordination time interval, I_i is the fault current for the downstream relay, I_{pi+1} is the pickup current for the upstream relay, and α and β are the curve type constants.

The upstream relay uses (3) to update its operating time (t_{i+1}). Similarly, all higher level relays will update their protection settings based on the new settings of the downstream relay acquired via the GOOSE communication protocol

$$t_{i+1} = \frac{TMS_{i+1} \times \beta}{\left(\frac{I_{i+1}}{I_{pi+1}}\right) - 1}, \tag{3}$$

where I_{i+1} is the fault current for the upstream relay.

The implementation involves three STM32MPU boards simulating numerical relays A, B, and C, with Relay A as the primary relay and Relays B and C as lower and higher level backups, respectively. To detect changes in the status of the switch or the breaker within the network, four digital input pins on Board A are configured to represent circuit breakers CB1, CB2, CB3, and CB4.

Following the methodology discussed in [34], when only CB2 is closed, concatenating the CB status results in a Netconfig value of 4. For a 0.2 second operating time value, 0.12 seconds CTI value, 1.3 A fault current used in the initial TMS calculation, and 0 blocking signal value (indicating no fault), the GOOSE message published by the primary relay can be observed in the captured network traffic data using Wireshark. The transmission of relevant data can be observed in Fig. 7.

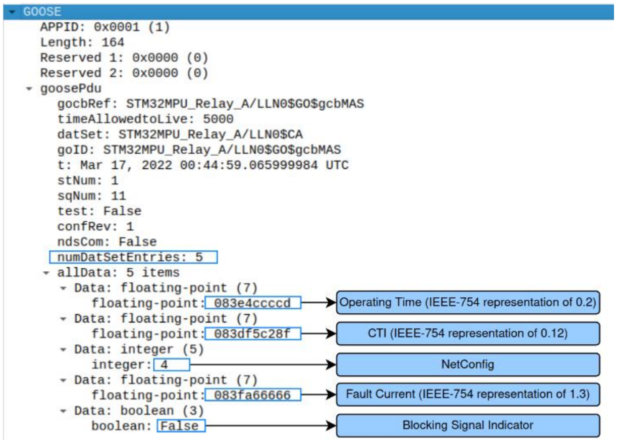

Fig. 7. GOOSE message published by Relay A (primary).

Since Relay B serves as a backup for Relay A, it will receive the GOOSE message published by primary Relay A. Based on the content of the message, Relay B will calculate its new protection settings. The captured network traffic data illustrated in Fig. 8 shows that the updated operating time for Relay B is 0.32 seconds. It will send this information to Relay C, which serves as a backup for Relay B. Relay C will then determine its new protection settings based on the received message. The new operating time for Relay C is 0.44 seconds, as depicted in Fig. 9.

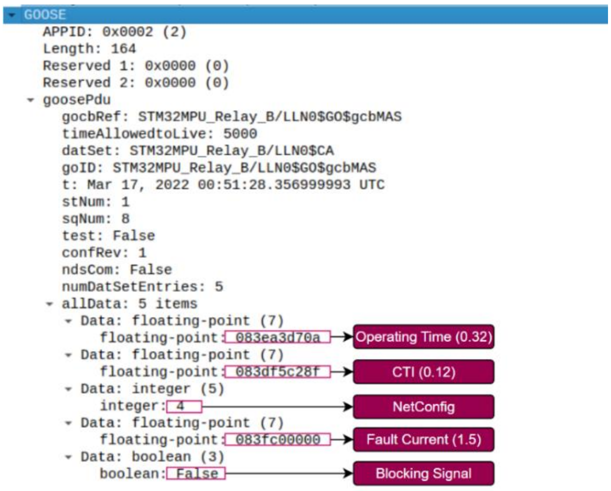

Fig. 8. GOOSE message published by Relay B (backup 1).

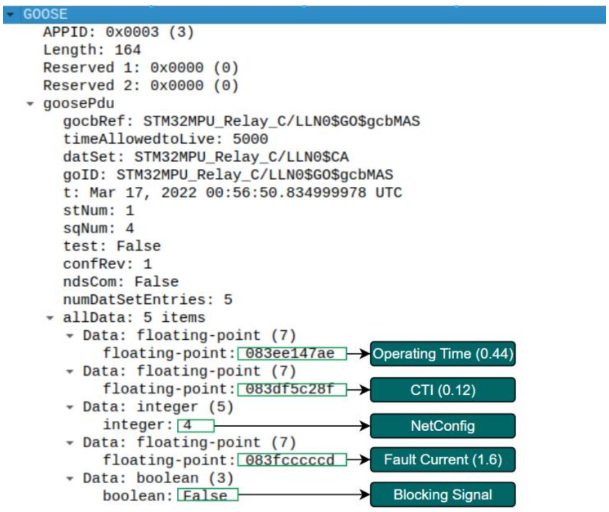

Fig. 9. GOOSE message published by Relay C (backup 2).

The graph in Fig. 10 illustrates the coordination among multiple numerical relays during network configuration changes. As depicted in this figure, when a network configuration changes (i.e., for NetConfig values of 4 to 7), the primary relay (Relay A) updates its protection settings first and then informs its immediate higher level backup relay (Relay B) to adjust its settings accordingly. After adjusting its protection settings, Relay B passes this information on to its immediate higher level relay (Relay C). By doing so, all primary and backup relays will have up-to-date protection settings whenever any network configuration change occurs in the power grid.

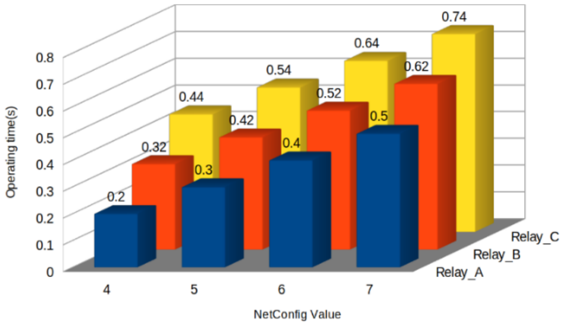

Fig. 10. Coordination of operating time for changes in network configuration.

Similarly, in this work, a GOOSE message structure was implemented for relays located on opposite sides of the power flow. The message comprises the operating time and follow-me signal described in [34].

V. Conclusions

Numerical relays play a crucial role in ensuring the efficient operation of electrical grids. Modern versions of these relays come equipped with various protection functionalities, and utilise multiple communication protocols to exchange data between IEDs or SCADA systems. This work presents the firmware structure used in the design of an adaptive numerical relay. It provides details of the protection schemes implemented, communication protocols, local HMI, and embedded systems. STM32MPU boards are utilised to implement all these features. Intelligent grid management can be enhanced by utilising adaptive multiagent systems (MAS). This approach improves the control, supervision, and decision-making processes. The practical implementation of a MAS-based protection approach demonstrates its effectiveness in analysing real-time data and formulating coordinated decisions. Using IEC 61850 GOOSE for real-time information exchange can aid in rapid fault detection and isolation. This work also includes comprehensive flow charts that depict the overall relay operation. The hardware implementation of the presented research, which includes all the hardware modules, will be covered in the second part of this work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Ramarao, S. K. Telagamsetti, and V. S. Kale, "Design of microcontroller based multi-functional relay for automated protective system", in *Proc. of 2014 Students Conference on Engineering and Systems*, 2014, pp. 1–6. DOI: 10.1109/SCES.2014.6880058.

[2] M. A. Aftab, S. M. S. Hussain, I. Ali, and T. S. Ustun, "IEC 61850 based substation automation system: A survey", *Int. J. Electr. Power Energy Syst.*, vol. 120, art. 106008, 2020. DOI: 10.1016/j.ijepes.2020.106008.

[3] M. Abdel-Salam, R. Kamel, K. Sayed, and M. Khalaf, "Design and implementation of a multifunction DSP-based-numerical relay", *Electr. Power Syst. Res.*, vol. 143, pp. 32–43, 2017. DOI: 10.1016/j.epsr.2016.10.033.

[4] P. R. Lopes and R. B. Junior, "Development of a low-cost relay prototype for real-time power protection functions", *Learning & Nonlinear Models (LNLM) - J. Brazilian Soc. Comput. Intell.*, vol. 21, no. 1, pp. 90–109, 2023. DOI: 10.21528/lnlm-vol21-no1-art7.

[5] V. Maheshwari, B. Das Devulapalli, and A. K. Saxena, "FPGA-based digital overcurrent relay with concurrent sense-process-communicate cycles", *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 66–73, 2014. DOI: 10.1016/j.ijepes.2013.08.032.

[6] S. Varshney, S. Jain, S. Gupta, Sakshi, and V. Maheshwari, "Design and development of multifunction voltage relay on FPGA", in *Proc. of 2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 2017, pp. 1–5. DOI: 10.1109/TEL-NET.2017.8343538.

[7] K. Shehata, A. Bahaa, K. Morad, and A. Sharaf, "Design and implementation of FPGA based and microcontroller based current relay", in *Proc. of the 16th International Conference on Microelectronics, 2004 (ICM 2004*), 2004, pp. 783–786. DOI: 10.1109/ICM.2004.1434783.

[8] J. Rahebi and M. M. S. Al-Shalah, "Design, modeling and implementation of multi-function protective relay with digital logic algorithm", *Eur. J. Sci. Technol.*, no. 19, pp. 549–565, 2020. DOI: 10.31590/ejosat.738337.

[9] F. Wadie, E. Abdulgawad, and A. Mousa Abd-Allah, "Adaptive real-time protection scheme for distribution networks with high penetration levels of renewable energy resources", *Ain Shams Eng. J.*, vol. 14, no. 7, art. 102005, 2023. DOI: 10.1016/j.asej.2022.102005.

[10] S. P. Ramli, H. Mokhlis, W. R. Wong, M. A. Muhammad, and N. N. Mansor, "Optimal coordination of directional overcurrent relay based on combination of Firefly Algorithm and Linear Programming", *Ain Shams Eng. J.*, vol. 13, no. 6, art. 101777, 2022. DOI: 10.1016/j.asej.2022.101777.

[11] M. V Tejeswini, I. J. Raglend, T. Yuvaraja, and B. N. Radha, "An advanced protection coordination technique for solar in-feed distribution systems", *Ain Shams Eng. J.*, vol. 10, no. 2, pp. 379–388, 2019. DOI: 10.1016/j.asej.2019.04.003.

[12] Ch. D. Prasad and P. K. Nayak, "A DFT-ED based approach for detection and classification of faults in electric power transmission networks", *Ain Shams Eng. J.*, vol. 10, no. 1, pp. 171–178, 2019. DOI: 10.1016/j.asej.2018.02.004.

[13] J.-S. Kim *et al.*, "Microgrids platform: A design and implementation of common platform for seamless microgrids operation", *Electr. Power Syst. Res.*, vol. 167, pp. 21–38, 2019. DOI: 10.1016/j.epsr.2018.10.019.

[14] M. El Hariri, T. A. Youssef, and O. A. Mohammed, "On the implementation of the IEC 61850 standard: Will different manufacturer devices behave similarly under identical conditions?", *Electron.*, vol. 5, no. 4, p. 85, 2016. DOI: 10.3390/electronics5040085.

[15] T. S. Ustun, "Interoperability and interchangeability for microgrid protection systems using IEC 61850 standard", in *Proc. of 2016 IEEE International Conference on Power and Energy (PECon)*, 2016, pp. 7–12. DOI: 10.1109/PECON.2016.7951463.

[16] T. Bhattacharjee and M. Jamil, "GOOSE publishing and receiving operations of IEC 61850 enabled IEDs", in *Proc. of 2019 IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP)*, 2019, pp. 1–6. DOI: 10.1109/ICESIP46348.2019.8938272.

[17] S. Hwang, "Investigation of wireless IEC 61850 MMS using Raspberry Pi", vol. 12, no. 13, pp. 5770–5778, 2021.

[18] P. S. Pereira Jr, R. C. Bernardino, G. S. Salge, C. M. Martins, P. S. Pereira, and G. E. Lourenço, "Performance assessment of a line protection implemented with process bus and GOOSE through transient closed loop tests", *Electr. Power Syst. Res.*, vol. 197, art. 107221, 2021. DOI: 10.1016/j.epsr.2021.107221.

[19] M. Krakowski, K. Kurek, and Ł. Nogal, "Comparative analysis of the DAQ cards-based and the IEC 61850-based real time simulations in the matlab/simulink environment for power system protections", *Electr. Power Syst. Res.*, vol. 192, art. 107000, 2021. DOI: 10.1016/j.epsr.2020.107000.

[20] H. Khalid and A. Shobole, "Existing developments in adaptive smart grid protection: A review", *Electr. Power Syst. Res.*, vol. 191, art. 106901, 2021. DOI: 10.1016/j.epsr.2020.106901.

[21] J. R. Pesente, J. G. Rolim, and M. Moreto, "MultiAgent Systems in power system protection: Review, classification and perspectives", *IEEE Lat. Am. Trans.*, vol. 14, no. 7, pp. 3285–3290, 2016. DOI: 10.1109/TLA.2016.7587632.

[22] A. Cagnano, E. De Tuglie, and P. Mancarella, "Microgrids: Overview and guidelines for practical implementations and operation", *Appl. Energy*, vol. 258, art. 114039, 2020. DOI: 10.1016/j.apenergy.2019.114039.

[23] S. S. Puhan, R. Sharma, and S. R. Lenka, "A report on Multi-Agent System application in power system", in *Soft Computing Applications in Modern Power and Energy Systems. Lecture Notes in Electrical Engineering*, vol. 975. Springer, Singapore, 2023, pp. 133–143. DOI: 10.1007/978-981-19-8353-5_9.

[24] A. Shobole, M. Baysal, M. Wadi, and M. R. Tur, "An adaptive protection technique for smart distribution network", *Elektron. ir Elektrotechnika*, vol. 26, no. 4, pp. 46–56, 2020. DOI: 10.5755/j01.eie.26.4.25778.

[25] E. Abbaspour, B. Fani, and E. Heydarian-Forushani, "A bi-level multi agent based protection scheme for distribution networks with distributed generation", *Int. J. Electr. Power Energy Syst.*, vol. 112, pp. 209–220, 2019. DOI: 10.1016/j.ijepes.2019.05.001.

[26] M. S. Rahman, M. A. Mahmud, H. R. Pota, and M. J. Hossain, "A multi-agent approach for enhancing transient stability of smart grids", *Int. J. Electr. Power Energy Syst.*, vol. 67, pp. 488–500, 2015. DOI: 10.1016/j.ijepes.2014.12.038.

[27] M. S. Rahman, N. Isherwood, and A. M. T. Oo, "Multi-agent based coordinated protection systems for distribution feeder fault diagnosis and reconfiguration", *Int. J. Electr. Power Energy Syst.*, vol. 97, pp. 106–119, 2018. DOI: 10.1016/j.ijepes.2017.10.031.

[28] A. Kiani, B. Fani, and G. Shahgholian, "A multi-agent solution to multi-thread protection of DG-dominated distribution networks", *Int. J. Electr. Power Energy Syst.*, vol. 130, art. 106921, 2021. DOI: 10.1016/j.ijepes.2021.106921.

[29] A. A. Shobole and M. Wadi, "Multiagent systems application for the smart grid protection", *Renew. Sustain. Energy Rev.*, vol. 149, art. 111352, 2021. DOI: 10.1016/j.rser.2021.111352.

[30] Z. Liu and H. K. Høidalen, "A simple multi agent system based adaptive relay setting strategy for distribution system with wind generation integration", in *Proc. of 13th International Conference on Development in Power System Protection 2016 (DPSP)*, 2016, pp. 1–6. DOI: 10.1049/cp.2016.0025.

[31] J. Treviño, A. Conde, E. Fernández, and O. Arreola, "Tuning interconnection relays using a multi-agent system to detect weak infeed conditions", *Electr. Power Syst. Res.*, vol. 169, pp. 139–149, 2019. DOI: 10.1016/j.epsr.2019.01.002.

[32] M. J. Daryani and A. E. Karkevandi, "Decentralized cooperative protection strategy for smart distribution grid using Multi-Agent System", in *Proc. of 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, 2018, pp. 134–138. DOI: 10.1109/SGCF.2018.8408958.

[33] A. A. Shobole, "Multi-Agent Systems based adaptive protection for smart distribution network", *Electr. Power Components Syst.*, vol. 49, nos. 18–19, pp. 1432–1444, 2021. DOI: 10.1080/15325008.2022.2131937.

[34] A. A. Shobole and M. Abafogi, "Adaptive protection in smart distribution networks: Coordination demonstration of Multi-Agent Systems", in *Proc. of 2023 5th Global Power, Energy and Communication Conference (GPECOM)*, 2023, pp. 483–488. DOI: 10.1109/GPECOM58364.2023.10175704.