Employing Graph Traversal Techniques to Simplify Three and Multi-Terminal Line Applications

Naveen Ganta, P.E., M.S.
SynchroGrid
Bryan, TX USA
naveen.ganta@synchrogrid.com

Luke Hankins, P.E.
SynchroGrid
Bryan, TX USA
lukeh@synchrogrid.com

Thomas Goodwin
SynchroGrid
Bryan, TX USA
thomas.goodwin@synchrogrid.com

***Abstract*—Developing relay settings for three-terminal applications can be challenging, as protection engineers need to account for different system variables that can add complexity to their calculations. These variables could be considerations for strong and weak sources, mutual coupling, sequential tripping, tap lines, forks of different lengths, looped topology, etc. The application can get even more complex when the application becomes a multi-terminal system, meaning three plus terminals, as in the case of parallel three-terminal lines. Adding to the complexity of the system, the setting engineer must consider all events under an N-1 contingency expanding the amount of work that needs to be done to achieve proper settings.**

In this paper, we present the implementation of graph traversal algorithms to intelligently graph the grid to properly automate line setting calculations for three- and multi-terminal line applications. By expressing the grid and related calculations as a graph data structure, the complexity of the application can be greatly simplified by providing engineers with an automation template engine. A template-based system allows each engineer to specify all setting calculation requirements and system conditions, allowing the engineer to run the template on any multi-terminal system to calculate settings with minimal effort. Solutions, such as the one being presented in this paper, are imperative for engineers in order to keep up with the high demand of output they are currently facing due to the fast-growing grid and NERC compliance requirements.

***Index Terms*—Directional Time Overcurrent, Microprocessor Relays, Time Dials, Wide Area Coordination, Distance Elements**

I. Introduction

The electric grid is experiencing ever-growing complexities on all fronts due to factors like integration of distributed energy resources, grid digitization, grid modernizing initiatives, demand management efforts, and interconnections, among other technological advancements. These complexities necessitate a collective effort from utilities, regulators, technology providers, and consumers to ensure system reliability and coordination. With the electric grid getting more sophisticated on a yearly basis, the processes for developing relay settings and evaluating the system scenarios also need to become more meticulous to maintain the reliability of the grid and avoid needless power system operations.

In Section II we will discuss misoperation data of the last seven years and what challenges of system protection contribute to the high percentage of human error caused misoperations.

We will then give a brief overview of graph data structures and algorithms and how it can be applied to power systems in Section III.

We will briefly discuss the definition of a three terminal lines and a few common challenges specific to three and multi terminal lines in Section IV.

In Section V, we will present the advantages of representing a three-terminal line as a graph alongside an automated system created for heightened efficiency and a streamlined process.

We conclude in Section VI.

II. Grid Complexity

A. Misoperation Data

This section delves into the statistics of misoperations gathered by NERC since 2016 to give the reader an idea of the root causes of these unwelcome incidents and to what proportions of them are directly or indirectly due to organizational or individual practices.

Based on the misoperation data gathered from the NERC State Of Reliability reports [1]–[4] for the past seven years (2016-2022), we have compiled the annual misoperation rates across North America separately for each regional entity as seen in Figure 1.

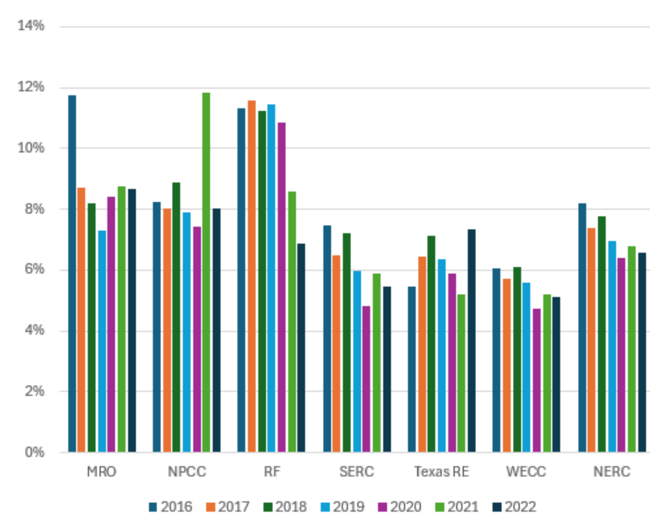

Fig. 1. Misoperation Rate by Regional Entity

There has been a gradual downward trend in the misoperation rate of overall NERC entities over the years, except for 2018 and 2021 where we have seen a spike in these misoperations. While Reliability First has seen a significant decrease in its misoperation rate since 2019, Texas Reliable Entity has seen highest number of misoperations as recent as 2022.

It has been determined that the primary causes of these misoperations over the data gathering period (2016-2022) have consistently been Incorrect Settings and Relay Failures/Malfunctions as seen in Figure 2.

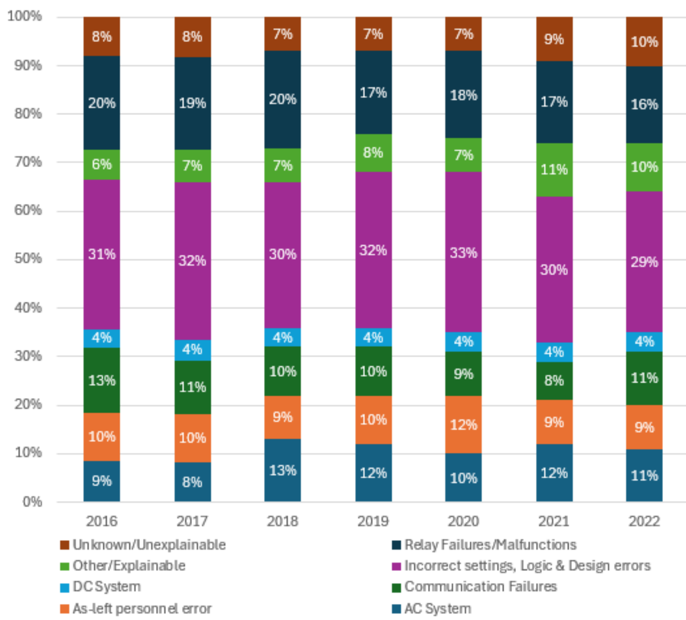

Fig. 2. Misoperations by Cause Code

While the percentage of misoperations due to incorrect settings over the past years remains close to 30%, the percentage of the misoperations due to relay failures and malfunctions averages about 18%. These two reasons have accounted for around 50% of all the misoperations ranging from a high of 52% in 2017 to the lowest value of 45% observed in 2022. The relative frequency of these top contributors has continued to decrease in the recent past. It should also be observed that there has been a slow increase in the number of misoperations coded as unknown/unexplainable in the recent past. This emphasizes the growing complexity of the electric grid and the need to improve processes to reduce these avoidable scenarios.

Let us now focus on human error as one of the potential causes of protection system misoperations. ‘Human Error’ as a cause of transmission outages is defined in the data reporting instructions of Transmission Availability data System (TADS) as a relative human factor performance that include any incorrect action traceable to employees and/or contractors to companies operating, maintaining, and/or assisting the Transmission Owner. The Misoperation Information Data Analysis System (MIDAS) has reported several different causes of human error misoperations which account for roughly 40% of misoperations over the last seven years. These include as-left personnel errors, incorrect settings, logic errors and design errors. Development of incorrect settings due to known or unknown reasons is determined to be the most prominent cause of human error misoperations.

NERC stated that, while many of the major events labeled as being caused by individual human error initially, it has been later revealed that the majority of such errors stemmed from inherent organizational errors. Such revelations hint at opportunities and the existing need for the industry – as well as the individuals – to improve reliability through increased focus in the areas of management, organization performance, and engineering design.

This paper attempts to address incorrect settings as the outstanding reason for human error misoperations by coming up with creative automated solutions that reduce the human intervention in relay setting development.

B. Challenges

With increased complexity of the electric grid, there is a heightened need for improved relay setting processes that include accurate calculations and inclusion of all possible scenarios that the system can experience as well as a need for robust peer review processes. Manual calculations and simulations involved in the relay setting development process for current power systems pose a great deal of difficulties and an increased probability of errors. Let us look at some of these possibilities.

Manual relay setting development processes for a given system comprise a lot of time consuming tasks like taking the same set of contingencies multiple times and simulating sweeping faults to evaluate the settings. The same set of fault scenarios and contingencies need to be simulated again to calibrate and accept the calculated setting before sending for a review process. If the system under study includes complex scenarios involving loops, parallel lines and three or multi terminal lines that are mutually coupled, the set of simulations needed to be evaluated and contingencies to consider will not only increase the efforts and time needed, but the probability of human error as well. Even if the relay setting development processes are recognized to be technically prudent considering all possible scenarios, they are repetitive and sometimes extremely long drawn given the complexities of the system.

We would like to identify some of the common errors that relay setting engineers may commit unknowingly which would result in incorrect settings paving the way for misoperations in the system. While considering N-1 contingencies is vital in establishing a reliable and coordinated system, forgetting to put a line or generator back in service during manual simulation in the short circuit programs is a very common oversight. While some short circuit programs have resetting features to avoid these kind of unforeseen human errors, those features only increase the time needed to develop the settings manually. When a transmission line is modeled as multiple line sections to account for its conductor profiling, some short circuit programs need each of these line sections to be individually taken as a contingency and grounded when evaluating ground elements to study the impact of mutually coupled lines. While failure to perform this action results in erroneous analysis, manual evaluation demands additional time. Other forms of common human errors during manual relay setting development include forgetting to reconfigure a bus terminal to its original state after splitting it for evaluating the system scenarios, copy-paste errors, calculation, and rounding-off errors.

All of the aforementioned difficulties and actions that may lead to possible human errors call for sophisticated automated solutions that are both customizable to individual or organizational needs and also improve the turn-around time for relay setting development. This paper demonstrates one such effort in developing a platform that provides a relay setting engineer the ability to automate the relay setting processes while giving them full control of the protection philosophy intended to be implemented.

III. Graph Overview

A. Graph Data Structure

In the field of computer science, a graph is a non-linear data structure consisting of vertices and edges. A vertex represents an entity or object – a data point in the graph. An edge represents a connection between two vertices – that there is some association or interaction between the two vertices it connects. In this fashion, a graph can be used to model translatable real-world problems across several different domains.

As an example, in social networks a graph data structure can be implemented wherein vertices represent users and edges represent connections between the users (i.e., ”friend” status). This approach is commonly used in social media platforms.

Another common example is seen in transportation networks. We will now introduce a new concept: weighted edges. A weighted edge has some numerical property associated with it. While in the social network example an edge was seen purely as a connection with no detailed properties, a transportation network may have an edge that has a weight based on distance, as seen in the classic traveling salesman problem (i.e., TSP). In this problem, the shortest path between each city is calculated (see example shown in Figure 3). The ”weight” of each edge of the graph represents a variable distance between two vertices and is not merely a connection. Through applied graph algorithms (e.g., as discussed in Section III-B) a solution to the traveling salesman problem may be found.

An important point to note is that both of the aforementioned algorithms are applied to undirected graphs, that is, there is no implication of one vertex being a child or derivative of another, they are represented as connections with no hierarchy or directionality. In the eventual application to power system models, we will see an undirected weighted graph, similar to the transportation network.

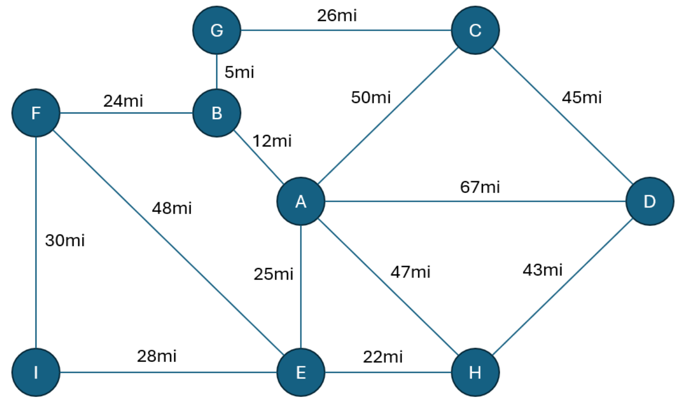

Fig. 3. The Classic Traveling Salesman Problem: Undirected Weighted Graph

B. Graph Data Algorithms

There are two primary algorithmic approaches commonly used to traverse graph data structures: Breadth-First Search (BFS) and Depth-First Search (DFS) [5].

A BFS algorithm is implemented when a vertex-based approach is necessitated. It is implemented with a queue data structure (FIFO - First In, First Out) in order to iteratively search through vertices one tier at a time. It would thus be implemented when it is important to only search a certain number of tiers out from a starting point.

A DFS algorithm is implemented when an edge-based approach is necessitated, i.e., where all possible routes out must be found and it is not relevant how far out the traversal must go. It is implemented with a stack data structure (FILO - First In, Last Out / LIFO - Last In, First Out). It may also be implemented more simply with recursion. A recursive function can be defined as a function that calls itself and has some terminating condition. Recursion is convenient because the program creates and keeps track of its own call stack so that the programmer does not have to manually implement and keep track of their own stack data structure.

Both DFS and BFS can be modified to record data as they traverse the graph. This property enables their use in a wide variety of applications (e.g. relay settings development).

C. Application to Power System Model

Graph data structures are a natural fit for representing a oneline model. Buses can be modeled as vertices and equipment connecting those buses can be modeled as edges. Each vertex can contain any amount of information necessary to represent buses, like voltage level, name, and whether or not the bus is tapped. Similarly, each edge can contain any relevant information for the equipment it represents. For example, any modeled lines can have impedance values and length. When the electrical grid is modeled as a set of vertices and edges, graph algorithms can be used to assist in grid analysis.

One example of a problem wherein modeling the grid as a graph can be advantageous is using a DFS to identify line terminals automatically for a multi-segment line. Starting from a line segment connected to the local terminal, the DFS algorithm searches down each possible path until encountering a bus that is not marked as tap. Once such a bus is encountered, the modified DFS algorithm records the current path and moves to the next. This will prevent the algorithm from traversing the whole grid. Once the algorithm is finished, all paths to the remote terminals of the line will have been found. It can also record any tap lines by finding any paths that end on a tap bus. In the case of a two-terminal line, DFS can be modified to find the remote end and the path to it, but in the case of a multi-terminal line, it can be used to find all ends and the paths to those ends. These paths can then be used to automatically calculate impedance data and to specify fault placements.

A template-based system can use these algorithms to greatly simplify the process of protective relay settings development. The system can be implemented to automatically find the primary line terminals, remote lines, source lines, transformers, and any other relevant equipment for any given line. Such a system can also be used to retrieve the impedance values of all equipment. Using this information, the proposed system can perform any kind of fault analysis that an engineer might need (e.g. faults on any surrounding equipment or many kinds of alternate grid conditions). For example, the template can specify that a simulated fault needs to be performed at the remote bus of the shortest remote line under n-1 contingencies, choosing the one with the lowest apparent impedance, or the highest current, etc. The system can go through all of the remote lines found during graph traversal, identify the shortest one, and then select all of the other remote lines as contingencies to take out service. It can then simulate all of these faults to identify the one that has the minimum impedance.

Another example of applying DFS to real-world grids modelled as graphs is identification of the operational reach of a particular relay element. The approach begins by selecting the primary line. The algorithm then does sweeping faults on the line to determine where the particular element stops operating. If the element operates at the end of the primary line, then the algorithm will do the same sweeping faults on a remote line to determine if the relay operates on that line. It will continue traversing the grid until it finds the point where the relay no longer operates. Whenever it finds an operation point, it can also record the reach down that particular line. Once it finishes, all of the farthest reach points of that element will be recorded. This approach could be very useful to test potential settings. For example, the engineer can verify that the new Zone 2 setting for a relay does not overreach a specific percentage of the remote lines.

By modelling the grid as a graph, a three-terminal line can be represented in a way that makes it much simpler to analyze. By applying the line terminal finding algorithm described above, a separate path to each remote end can be identified. In this case, a path is the set of edges joining a sequence of vertices from one line terminal to another. The impedance values of these paths can be calculated separately and faults can be automatically simulated on both paths. This approach will greatly assist engineers in the process of developing settings for these lines, as discussed in Section V.

IV. Three and Multi Terminal Lines

A. Three-terminal Line Considerations

A three-terminal line is characterized by a line with a fork in the middle that leads to three separate buses with their own respective sources and loads (i.e., not simply a tap line), with no protective elements at the fork. Three sets of settings need to be calculated, each from the perspective of each end of the fork.

When calculating settings for a three-terminal line, there are many factors for a protection engineer to consider. Both paths out from each bus need to be considered, and thus, have their respective impedance values calculated separately. Since the paths are protected by the same relay, the settings must be generalized to adequately protect both. Depending on the case, it can be impossible to have a perfect solution, so compromises must be made, necessitating a heavy reliance on communication schemes. A simple example is shown in Figure 4. The relay must never trip instantaneously for external faults, so Zone 1 must always be limited by the shortest path and some margin (even in cases were the shortest path is much shorter, leaving a significant portion of the longest path uncovered). Additionally, consider scenarios wherein the other side of the shortest path is a short remote line, the distance of which still is not as far as the longest path of the three-terminal line. For a Zone 2 setting to be able to cover the longest path, it will end up reaching beyond the remote line on the shortest path and thus must be adequately time delayed.

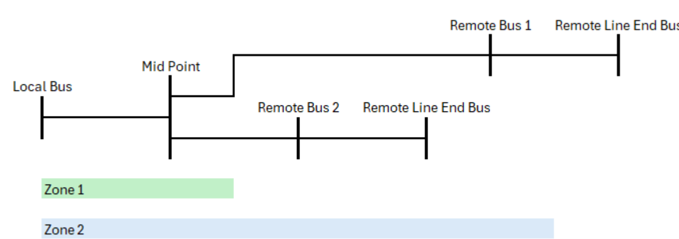

Fig. 4. Zone Considerations

Zone reaches are one of the simplest of many examples of multi-terminal specific considerations.

B. Case Study

In addition to other three-terminal-specific considerations, there is the fact that there are three sets of settings to calculate, in contrast with two sets of settings for two terminal lines (that end up being very similar). For the case study shown in Figure 5, there are many tap buses. This requires the retrieval of many R and X values to calculate the full impedance of each path of the line. In the aforementioned case study, there are also two tap lines and two tapped distribution transformers. Line settings will need to be developed such that they operate for faults on each main path of the line, each tap line, and *not*

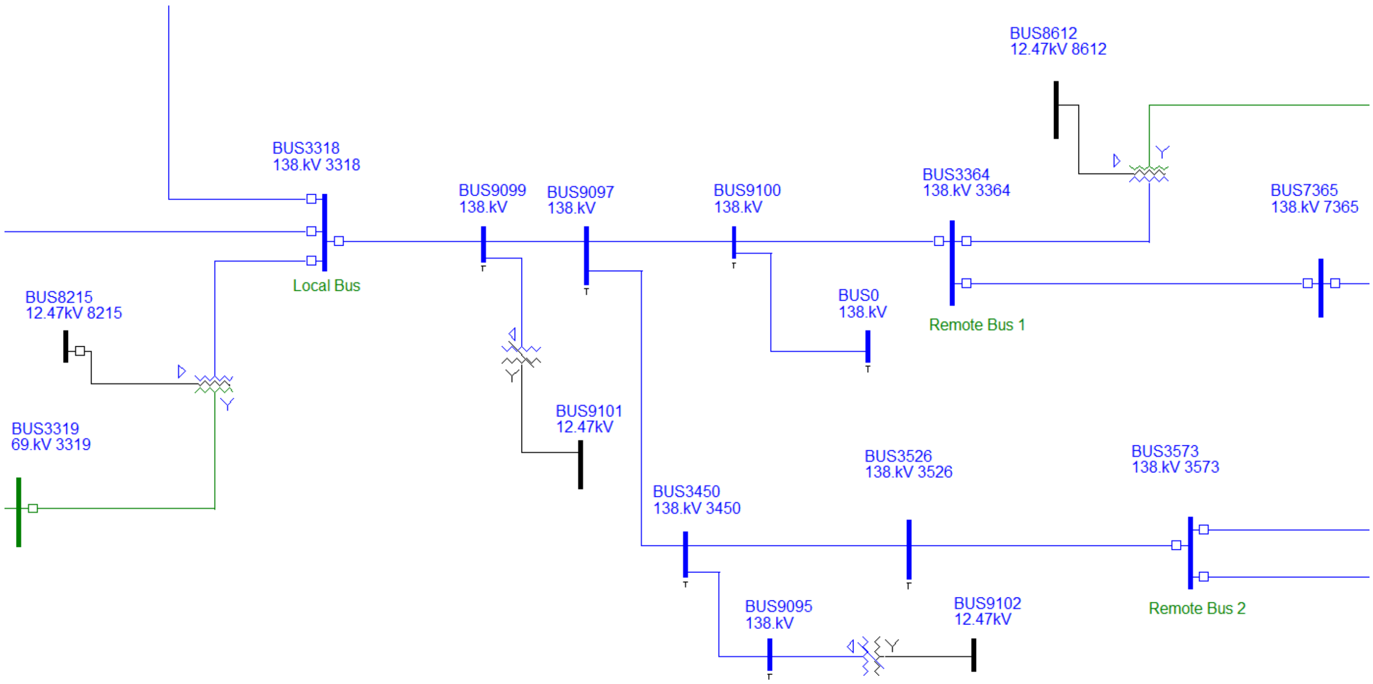

Fig. 5. Case 1: Three Terminal Line

on the other sides of the transformers. The longest tap line and shortest distribution transformer from the perspective of one relay might not be the same from the perspective of the other relays.

Many lines and transformers are connected to each end of the three-terminal line. Line settings should be made to protect appropriate lengths down remote lines while avoiding pitfalls such as those outlined in Section IV-A. The proper contingencies must be taken from this set of equipment as well as any mutually-coupled equipment to ensure the settings will operate correctly in alternate grid conditions.

The scenario given above is a very high-level view of examples of considerations when setting three-terminal lines. It would otherwise necessitate a very long checklist, but modelling multi-terminal lines as a graph opens the door for a very automated and streamlined process, which – once implemented – goes a long way in guaranteeing that all cases are accounted for and nothing gets missed.

V. Application

By modelling a three-terminal line as a graph and implementing the various grid traversal algorithms discussed in Section III-C, relay settings projects for three-terminal lines become far more manageable. This approach facilitates automated data retrieval, fault and contingency simulation, and allows the development of methods to automate calculation and testing of relay settings.

All relevant equipment is quickly identified – all source lines, both paths of the three-terminal line, tap lines and tapped distribution transformers, remote lines, and mutually coupled lines, each with their respective impedance values. There is no need for the engineer to worry if they have missed a line, and furthermore, there is no worry that an impedance value will be transcribed incorrectly. For example, a manual effort of calculating total line impedances would have the engineer copy and paste R and X values of each segment of each line, one avenue of undesirable contribution to the misoperation statistics outlined in Section II-A. Meanwhile, the graph traversal approach automates the retrieval of all impedance values along the way, combines them, and presents

PRIMARY LINE	
3573 BUS3573 (LONGEST SEGMENT)	
Positive Sequence Magnitude (pu)	0.13719
Positive Sequence Angle (°)	77.20941
Zero Sequence Magnitude (pu)	0.45084
Zero Sequence Angle (°)	77.88022
Length (mi)	34.0089
3364 BUS3364 (SHORTEST SEGMENT)	
Positive Sequence Magnitude (pu)	0.08419
Positive Sequence Angle (°)	67.58063
Zero Sequence Magnitude (pu)	0.25558
Zero Sequence Angle (°)	73.94344
Length (mi)	18.703

Fig. 6. Primary Line Impedance Values

the data in a readable form, in either pu or secondary ohms (i.e., Ω). In Figure 6, both primary paths of the three-terminal line in Figure 5 are represented in pu magnitudes with which is longest and shortest clearly labeled.

A. Shortest Primary Line Segment and Longest Primary Line Segment

All values from Figure 6 (and many more values covering the full range of analyzed lines and equipment) are displayed as well as stored in variables available to the engineer to use in any future calculation. For example, the variable for the Shortest Primary Line Segment positive sequence impedance magnitude is $SPLS_{Z1Mag}$, while the Longest Primary Line Segment counterpart is $LPLS_{Z1Mag}$. With these two variables at the engineer's disposal, they can now use them to calculate settings, such as Zone 1 and Zone 2. As discussed in Section IV-A, the engineer will likely want to use $SPLS_{Z1Mag}$ in the Zone 1 calculation to ensure underreaching the nearest remote bus and $LPLS_{Z1Mag}$ (and maybe some others such as SRL_{Z1Mag} which represents the shortest remote line of all remote lines found) in the Zone 2 calculation in order to ensure overreaching the furthest remote bus (and with a likelihood of needing to increase the Zone 2 Time Delay higher than the standard 20 cycles). Examples of these calculations are shown in Figure 7.

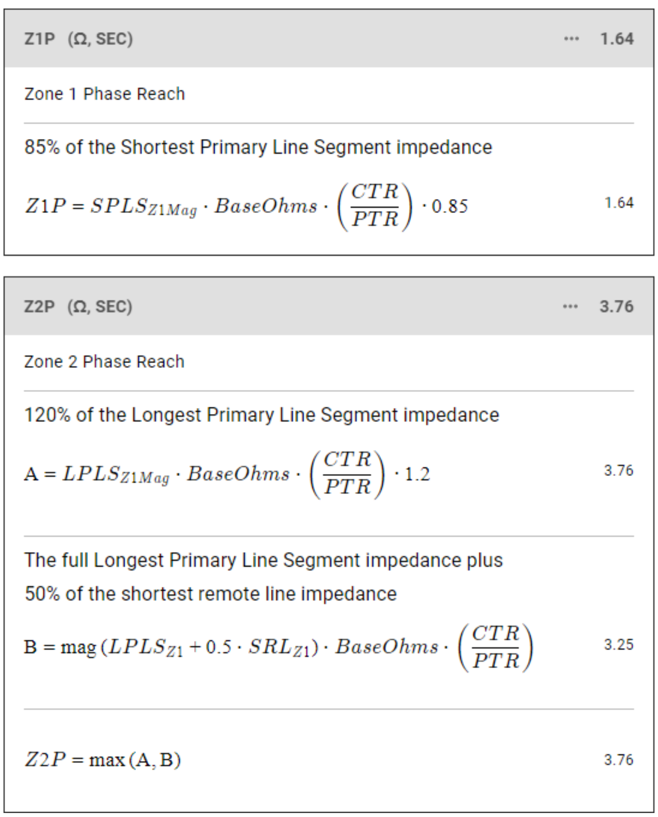

Fig. 7. Zone 1 and Zone 2 Reach Calculation

In addition to the characteristic impedance of a line, apparent impedance can be obtained through automated simulation of faults based on knowledge obtained from the graph. The proper fault location can be identified along with the most meaningful contingencies (i.e., lines or other equipment to remove), and the resulting apparent impedance will be stored and become usable in its own fault apparent impedance variable.

After all relay settings have been calculated, a significant portion of the testing can be automated by utilizing many of the same techniques.

One interesting check that can be done is a reach analysis to detect underreach or overreach of specific zones. For example, in the zone calculations shown in Figure 7, the engineer opted to use only characteristic impedance for the first draft of the settings. In this follow up check of Zone 1, faults are placed at the desired reach (e.g., in this case, 80% of each path of the three terminal line), one fault per relevant contingency. The contingency which results in the lowest apparent impedance should be used as a candidate replacement to the previously calculated zone setting. By setting it based on the worst case contingency, it is ensured that no contingency will cause an overreach. Figure 8 shows the calculated zones and the shortest primary line path. All X's represent the apparent impedances of each 80% intermediate fault that was simulated (i.e., one per path, per contingency). The dashed circle represents the proposed new Zone 1, which is slightly lower than the result of the characteristic impedance calculation.

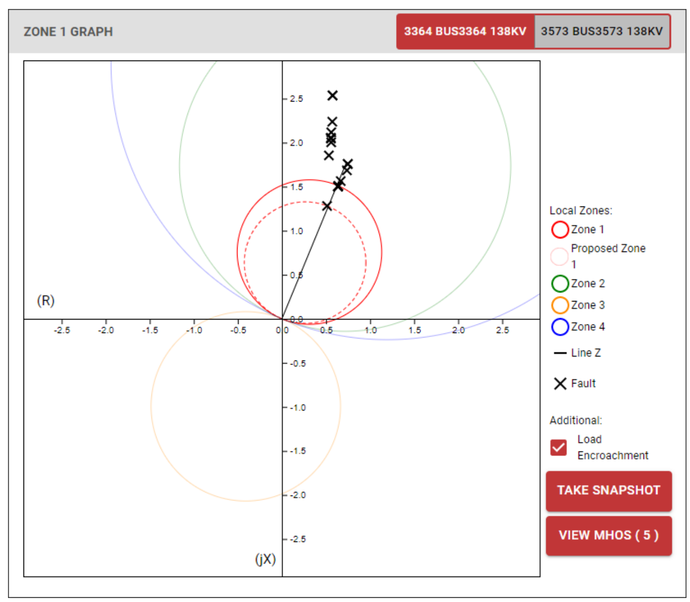

Fig. 8. Mho Circles

A similar process is followed for all zones, phase and ground, followed by a detailed reach analysis of all finalized zones, tested under all contingencies used in the initial study.

Another interesting check that can be done is a Coordination Time Interval (CTI) check, which verifies that the timing of the newly calculated settings will coordinate correctly with existing neighboring settings. A simple graph traversal decides where to place the faults to test as well as identifies which neighboring relays should be compared. Figure 9 shows a display of data from a test in which faults are placed on the one remote line of the shortest primary line path, and the primary settings are checked as a backup to the remote line settings. Blue circles indicate that the relays have operated in the proper order, while red circles indicate that the relays have either operated out of order, simultaneously, or too close to be permissible. In this case, upon further inspection of the data, it is found that the Zone 2 of the primary relay was calculated to be very high as shown in Figure 4, and the engineer has neglected to increase the time delay.

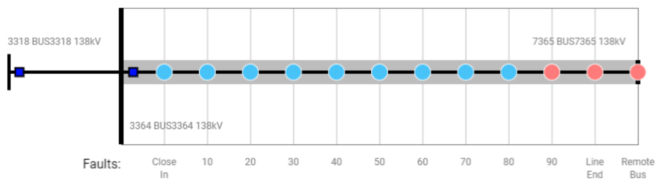

Fig. 9. Coordination Time Interval Check

VI. Conclusion

The plurality of relay misoperations have been and continue to be attributed to human error. Due to the growing size of the grid and thus, among other things, the necessity of a higher percentage of multi-terminal lines, system protection isn't getting any easier. By turning to methods of automation that both ensure a higher level of accuracy and correctness in work as well as streamline the process, we believe that human error can be driven down to an eventual minority cause of total misoperations.

References

[1] "2020 State of Reliability: An Assessment of 2019 Bulk Power System Performance" (2020, July). North American Electric Reliability Corporation.

[2] "2021 State of Reliability: An Assessment of 2020 Bulk Power System Performance" (2021, August). North American Electric Reliability Corporation.

[3] "2022 State of Reliability: An Assessment of 2021 Bulk Power System Performance" (2022, July). North American Electric Reliability Corporation.

[4] "2023 State of Reliability Technical Assessment: Technical Assessment of 2022 Bulk Power System Performance" (2023, June). North American Electric Reliability Corporation.

[5] Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, Clifford, "Introduction to Algorithms, Third Edition" https://dl.acm.org/doi/book/10.5555/1614191

VII. Author Biographies

Naveen Ganta graduated from the University of Texas at Austin with a M.S. degree in Electrical Engineering in 2017 and a B.Tech degree from the Indian Institute of Technology (IIT), Bhubaneswar in Electrical Engineering in 2015. Naveen is a registered Professional Engineer in the state of Texas and works as a Lead Engineer for SynchroGrid. Naveen performs NERC compliance studies, relay setting development, wide area coordination studies, fault and misoperation analysis and focuses in power system protection.

Luke Hankins graduated from Texas A&M University with a B.S. degree in Electrical Engineering in 2015. He is a registered professional engineer in the state of Texas and works for SynchroGrid in relay setting automation. He writes software applications in various programming languages, focusing on simplifying relay setting development. Luke also performs relay settings verification and mis-operation analysis.

Thomas Goodwin graduated from Texas A&M University with a B.S. degree in Computer Science in 2022. He works for SynchroGrid in relay setting automation. Thomas also performs relay linking services between relay settings databases and short circuit simulation software.