

# **NOVEL APPROACH TO RELAY SETTING DEVELOPMENT**

## **Authors:**

**Gene Corpus, LCRA, Austin, Texas, 78744**

**Matt Boecker, LCRA, Austin, Texas, 78744**

**Joe Perez P.E., SynchroGrid, College Station, Texas 77845**

**Luke Hankins, SynchroGrid, College Station, Texas 77845**

**Presented before the  
70th Annual  
Texas A&M Protective Relay Conference  
College Station, Texas  
April 3rd – April 6th, 2017**

# NOVEL APPROACH TO RELAY SETTING DEVELOPMENT

Gene Corpus, LCRA, Austin, Texas, 78744, [Genardo.Corpuz@LCRA.ORG](mailto:Genardo.Corpuz@LCRA.ORG)  
Matt Boecker, LCRA, Austin, Texas, 78744, [Matt.Boecker@LCRA.ORG](mailto:Matt.Boecker@LCRA.ORG)  
Joe Perez P.E., SynchroGrid, College Station, Texas, 77845 [jperez@synchrogrid.com](mailto:jperez@synchrogrid.com)  
Luke Hankins, SynchroGrid, College Station, Texas, 77845 [lukeh@synchrogrid.com](mailto:lukeh@synchrogrid.com)

## 1. INTRODUCTION

The 2015 NERC State of Reliability Report states that 31% of mis-operations are attributed to incorrect settings/logic/design errors **Error! Reference source not found.**[1]. According to the report, one major cause of mis-operations is human error. We know today that the current relay settings development process is laborious and difficult to complete in an accurate manner with today's tools. This process requires the setting engineer to manually transfer vast amounts of data from short circuit programs to calculation sheets and finally to relay setting files. In addition, this process requires extensive resources in peer-peer review programs to minimize human error. Depending on the internal process of the utility, preventing errors might not be feasible with the current process. This paper outlines these deficiencies and presents an innovative novel method that unifies as well as drastically simplifies the relay setting process. This new approach proposes a new way to create intelligent calculation templates, eliminates human error due to data transfer, and automates the relay setting development process.

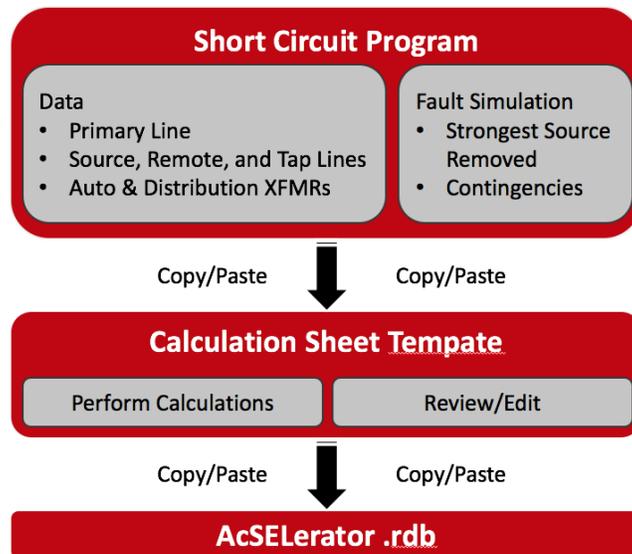
## 2. CURRENT RELAY SETTING PROCESS

Over the years, protection engineers have used many different methods to develop relay settings. During the electromechanical relay era, engineers would outline their formulas and solve their relay setting equations on a piece of paper and solve their relay setting equations. Engineers have since tried to simplify this process, and today many engineers use spreadsheets or Mathcad files to semi-automate the process. These techniques are preferred methods since they offer ways to perform user defined mathematical equations. This is done by creating equations that represent the utility protection philosophy, ultimately forming a philosophy template. This has facilitated the relay setting development process to some degree, but it is not ideal.

The process of relay setting development varies depending on the utility. Many utilities today do not have standardized relay setting calculation templates. In these cases, each setting engineer creates his/her own template that might or might not reflect the utility protection philosophy. As a result, many different versions of the philosophy can live within the same utility environment. This non-uniform process can produce settings for the same relay that differ from engineer to engineer and project to

project, resulting in incorrect settings. Not only might this lead to confusion and miscalculations, but a non-standardized process likely does not meet the guidelines for establishing a relay setting process as outlined in the forthcoming NERC PRC-027 standard [2].

To address these issues, some utilities have created standardized templates that allow them to have a uniform process of relay setting development across the entire utility. Since the standardized templates continue to be in the form of spreadsheets or Mathcad files, this type of process faces different challenges. Creating settings equations requires the use of lengthy formulas in single cells that are hard to follow and difficult to maintain. Perhaps the biggest challenge facing standardization is the multiple levels of data transfer from one program to another, which will inevitably introduce additional errors. Figure 1 below shows the transfer of data from short circuit program to a standard template calculation sheet and to the relay setting template file.



**Figure 1: Conventional Relay Setting Process Flow Chart**

The setting engineer must constantly refer to a short circuit program such as ASPEN or CAPE to obtain information about the protected line, such as line positive and zero sequence impedance, reverse and remote lines, tap lines, distribution transformer impedance, etc. Depending on the number of tap buses, tap lines, distribution transformers, and reverse and remote lines, the volume of data transfer can be significant.

Standardized templates also require engineers to perform fault simulations and collect fault information from the given application. Depending on the complexity of the utility philosophy, engineers must perform numerous fault simulations to calculate pickups, including 87L fault detectors, SOFT, LOP, time over-currents, distance element apparent impedances, along with many others. In some instances, these calculations require the engineer to perform N-M contingencies, which can result in an overwhelming number of fault simulations to be performed by hand. For example, the LCRA protection philosophy procedure for Zone 2 Ground requires several fault simulations before the ideal Zone 2

Ground setting is finally achieved. The engineer must find the minimum apparent impedance for 8 different cases, and each case contains multiple contingencies. While this method is initially more labor intensive, it minimizes coordination issues in later stages. Due to this lengthy, laborious process and the transfer of data from one program to another, the probability of corrupting the original data and losing information is very high, which ultimately leads to erroneous settings.

The current relay setting development processes using non-standardized and standardized solutions together with employing mechanics such as spreadsheets and Mathcad files are not the best choice for the configuration of critical power system infrastructure. The existing tools do not facilitate an efficient and straightforward framework to build utility protection philosophy calculations. The process is prone to massive human error, which has costly ramifications.

### 3. NEW APPROACH TO RELAY SETTING DEVELOPMENT

Protection engineers and programmers at SynchroGrid have created the Setting Automator Relay Assistant (SARA) that allows utilities to implement complex philosophy calculation templates without the need of Excel or Mathcad. This frees protection engineers from collecting data and eliminates errors due to data transfer from one program to another. SARA’s main structure is composed of template creation and application. The process for this new solution is shown in Figure 2 below.

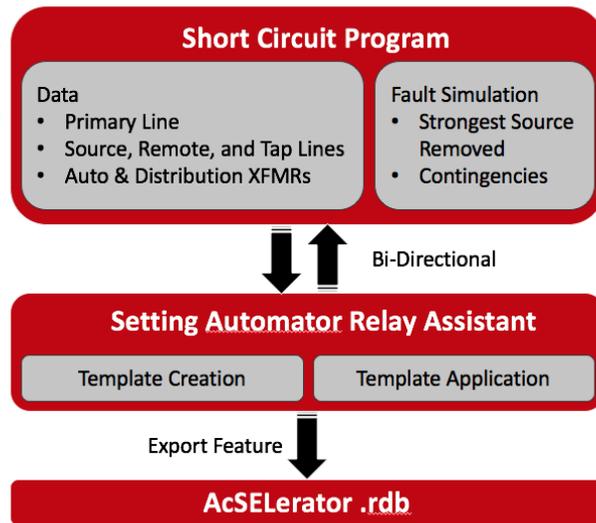


Figure 2: New Setting Automator Relay Assistant Program

#### 3.1. Template Creation:

This complex process has been dramatically simplified by allowing the setting engineer to define a set of equations and fault simulations that represent the intent of the protection philosophy in a manner that can be generally applied to any line to be protected. This forms a settings template.

#### Equation Builder:

Within the template creation section, the engineer can define equations that represents either a relay setting, a component from the grid, a user defined variable, or any mathematical function. Common variables such as CTR, PTR, PLZ1Mag (primary line impedance), BaseOhms, PLRemoteZ2 (Primary Line Zone 2 Remote Setting) and even more complex grid variables such as LTL (longest tap line), LRL (longest remote line), or SRL (shortest remote line) are readily available without the need of user intervention to calculate them. For example, the engineer could define Z1P as shown in Figure 3 below:

- Z1P	x
Zone 1 Phase Distance Reach	
$Z1P_{Case1} = 0.85 \cdot PL_{Z1Mag} \cdot BaseOhms \cdot \left( \frac{CTR}{PTR} \right)$	
$Z1P_{Case2} = PL_{Z1Mag} \cdot BaseOhms \cdot \left( \frac{CTR}{PTR} \right) - 0.1$	
$Z1P = \min(Z1P_{Case1}, Z1P_{Case2})$	

**Figure 3: Zone 1 Phase Calculation**

Creating logic decisions such as in Zone 3 Reverse shown in Figure 4 below are an efficient way to optimize relay settings. Notice the ability of this new methodology to automatically retrieve remote relay Zone 2 settings using the PLRemoteZ2Prelay variable without the need of user intervention.

- Z3MG	x
Zone 3 Ground Distance Reach	
$Z3MG_{Case1} = (PLRemoteZ2P_{relay} - PL_{Z1Mag} \cdot BaseOhms) \cdot \left( \frac{CTR}{PTR} \right) \cdot 1.5$	
$Z3MG_{Case2} = 1.2 \cdot LBL_{Z1Mag} \cdot BaseOhms \cdot \left( \frac{CTR}{PTR} \right)$	
$Z3MG = \max(Z3MG_{Case1}, Z3MG_{Case2})$	

**Figure 4: Zone 3 Reverse Setting Template**

This type of equation builder is preferred since it allows the protection engineer to closely follow all the equations and calculations that are part of the setting calculation in question.

### 3.1.1. Fault Simulations:

To accommodate a wide variety of complex philosophies, the program also allows the user to define many of the fault options already available in short circuit programs. This includes faults such as close-in, intermediate, line-end, local/remote bus, etc. The program also comes equipped with the ability to specify faults with N-M contingencies and the strongest source in or out. These fault definitions become convenient when faults for SOTF, LOP, 50G1P, and other settings are repetitive and are performed the same way for every application.

### **3.1.2. Flags:**

Flags can be used as a warning system to let the engineer know that certain calculations are out of the intent of the original design. For example, a flag can be used during Zone 2 coordination to warn the user that timers must be coordinated. Engineers can also implement flags for CT saturation warnings when the CT sees beyond 100A secondary for a close-in fault.

Once the engineer has finalized integrating all setting equations, and faults and flags that are part of the relay setting philosophy, all information is saved in a proprietary file extension STL file. The creation of a template is a one-time process that the engineer must perform for each relay model that they wish to develop settings for. Once a template is created it may be applied to any line to set multiple relays.

### **3.2. Template Application:**

The most impactful part of this new solution occurs when the engineer applies the setting template for a specific transmission line project. In this stage, the relay setting process has been simplified to a level never seen before. There are only two simple steps the user needs to take to calculate an entire set of line relay settings calculations. First, the user must choose the local and remote stations for the specific line. Second, the user must provide information for variables that changes from project to project such as CTR, PTR, line rating, etc.

When the template is invoked or called for a specific line within a grid, a transparent and bi-directional interaction with the short circuit program occurs, automatically performing data collection and fault analysis. The computed settings are then presented to the engineer in a beautiful and elegant format where engineers can visually inspect all data that has been retrieved including fault analysis performed and setting equations calculated. For special cases, the philosophy calculated setting may be overridden with the ideal setting value determined by the engineer. Finally, after the engineer has reviewed all settings and has gone through all checks, the engineer can easily export the settings to the relay setting program file.

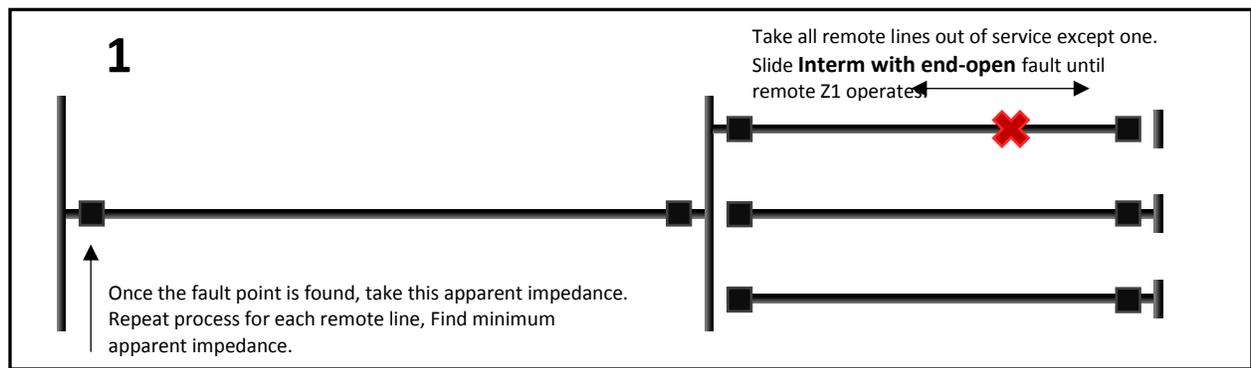
Hence, this framework greatly improves uniformity in relay settings development and frees the engineers from the tedious parts of the process, allowing them to focus on those parts where their skills are truly needed.

## **4. LCRA Application**

The efficacy of this new relay setting development process is better understood by applying it to a real example. LCRA has created a template that will allow them to quickly automate all their transmission line settings projects without the need to use spreadsheets, and copy and paste data from one program to another. In addition, because of the flexibility of the program, LCRA can perform multiple simulations, contingencies and checks to compute settings that are consistent with the utility philosophy and within the specified constraints.

For example, one of the most complex and labor intensive calculations for the LCRA protection philosophy is the calculation of Zone 2 Ground. To find the optimal Zone 2 setting, the engineer must find the minimum apparent impedance for 8 different cases for a total of 32 fault calculations.

For case 1, the engineer must take all remote lines out of service except one and perform an interim fault end-opened on the remote line until the remote Z1G operates and record the local relay apparent impedance. This is shown in Figure 5 below:



**Figure 5: Zone 2 Calculation Case 1**

This process must be repeated for every remote line in the application to find the minimum apparent impedance for all case 1 simulations. Without an automation solution, the engineer must not only perform the faults on all the remote lines manually, but he must also record the data for every operation. Fortunately, using the new intelligent template format, all these operations can be automated without user intervention. The template section for Zone 2 Ground is shown in Figure 6 below which contains the rest of the logic equations before the optimal Z2MG setting is finally achieved.

- Z2MG	x
Zone 2 Ground Phase Distance Reach	
$Z2MG_{Case1} = (0.9 \cdot (F1_{Zapparent} - F3_{Zapparent}) + F3_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$	
$Z2MG_{Case2} = (0.9 \cdot (F2_{Zapparent} - F4_{Zapparent}) + F4_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$	
$Z2MG_{Case3} = (0.9 \cdot (F5_{Zapparent} - F7_{Zapparent}) + F7_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$	
$Z2MG_{Case4} = (0.9 \cdot (F6_{Zapparent} - F8_{Zapparent}) + F8_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$	
$E = \min(Z2MG_{Case1}, Z2MG_{Case2}, Z2MG_{Case3}, Z2MG_{Case4})$	
$Z2MG_{Case5} = (1.2 \cdot \max(F9_{Zapparent}, F10_{Zapparent})) \cdot \left(\frac{CTR}{PTR}\right)$	
$Z2MG = \max(E, Z2MG_{Case5})$	

**Figure 6: Zone 2 Ground Calculation Template**

Table 1 below depicts the type of fault that must be performed by the program. Noticed that each fault has multiple contingencies and the program must find the minimum Zapp for each fault.

F1 <sub>Zapparent</sub>	With all remote lines out-of-service except for one, run a sequenced intermediate fault with end-open until the remote Zone 1 operates. Once the fault point is found, take this apparent impedance. Repeat process for each remote line, find F1 minimum apparent impedance.
F2 <sub>Zapparent</sub>	With all remote lines out-of-service except for one, run a sequenced intermediate fault with end-closed until the remote Zone 1 operates. Once the fault point is found, take this apparent impedance. Repeat process for each remote line, find minimum apparent impedance.
F3 <sub>Zapparent</sub>	With all remote lines out-of-service, run a remote bus fault, take this apparent impedance.
F4 <sub>Zapparent</sub>	With all remote lines out-of-service except for one, run a remote bus fault. Repeat process by putting in each of the remote lines and leaving the rest out-of-service. Find minimum apparent impedance.
F5 <sub>Zapparent</sub>	With all remote lines in-service, for each remote line run a sequenced intermediate fault with end-open until the remote Zone 1 operates. Once the fault point is found, take this apparent impedance. Repeat process for each remote line, find minimum apparent impedance.
F6 <sub>Zapparent</sub>	With all remote lines in-service, for each remote line run a sequenced intermediate fault with end-closed until the remote Zone 1 operates. Once the fault point is found, take this apparent impedance. Repeat process for each remote line, find minimum apparent impedance.
F7 <sub>Zapparent</sub>	With all remote lines in-service except for one, run a remote bus fault. Repeat process for each remote line, find minimum apparent impedance
F8 <sub>Zapparent</sub>	With all remote lines in-service, run a remote bus fault, take this apparent impedance.

**Table 1: Zone 2 Ground Fault Cases**

The results of the calculations are displayed to the user for final verification. This is shown in Figure 7 below. The significance of these calculations goes beyond the results. It is important to emphasize that by employing this new process through SARA, the transfer of data from one program to another has been completely eliminated and so the errors caused by this process. Furthermore, the use of the intelligent template allows the setting engineer to formulate equations much more complex than he previously could. This results in settings that are better calculated and more secure.

- Z2MG		4.42
Zone 2 Ground Phase Distance Reach		
$Z2MG_{Case1} = (0.9 \cdot (F1_{Zapparent} - F3_{Zapparent}) + F3_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$		3.521
$Z2MG_{Case2} = (0.9 \cdot (F2_{Zapparent} - F4_{Zapparent}) + F4_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$		3.587
$Z2MG_{Case3} = (0.9 \cdot (F5_{Zapparent} - F7_{Zapparent}) + F7_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$		4.84
$Z2MG_{Case4} = (0.9 \cdot (F6_{Zapparent} - F8_{Zapparent}) + F8_{Zapparent}) \cdot \left(\frac{CTR}{PTR}\right)$		5.251
$E = \min(Z2MG_{Case1}, Z2MG_{Case2}, Z2MG_{Case3}, Z2MG_{Case4})$		3.521
$Z2MG_{Case5} = (1.2 \cdot \max(F9_{Zapparent}, F10_{Zapparent})) \cdot \left(\frac{CTR}{PTR}\right)$		4.42
$Z2MG = \max(E, Z2MG_{Case5})$		4.42

**Figure 7: Z2MG Setting Calculations**

If the user believes in using a different setting, the program provides the ability to edit the settings before final release. Once the setting engineer is content with the value of the settings, he/she can export the settings straight to the relay setting file such as RDB.

LCRA believes this new solution will provide an intelligent way to uniformly standardized their relay setting templates for all their transmission system, eliminate all the errors from the data transfer used in the existing process, perform more complex calculations to find ultimate settings, and overall perform settings, peer-peer review processes and coordination studies at a fraction of the previous time.

## 5. CONCLUSION

This paper has demonstrated a sophisticated solution that introduces engineers to the future of relay setting development. The new solution provides a more advanced method to calculate settings as compared to existing methods such as spreadsheets or Mathcad programs. Settings engineers now can focus on the essence of relay settings rather than manipulation of data.

## 6. REFERENCES

[1] *Analysis of System Protection Misoperations* (Rep.). (2015, December). Retrieved February 22, 2017, from North American Electric Reliability Corporation website: [www.nerc.com](http://www.nerc.com)

[2] *PRC-027-1: Coordination of Protection Systems for Performance During Faults*. (2015, November). Retrieved February 22, 2017, from North American Electric Reliability Corporation website: [www.nerc.com](http://www.nerc.com)

## 7. BIOGRAPHY

**Joe Perez** received his B.S. degree in Electrical Engineering from Texas A&M University in 2003. Joe is the author of many relay application notes and has presented technical papers at WPRC, Texas A&M and Georgia Tech Relay Conferences. Joe is the owner of SynchroGrid, a registered professional engineer in the state of Texas and a member of PSRC, IEEE, and PES. Joe resides in the Bryan/College Station area. He can be contacted at [jperez@synchrogrid.com](mailto:jperez@synchrogrid.com).

**Luke Hankins** is from Cleveland, Texas. He graduated from Texas A&M University with a Bachelor's of Science degree in Electrical Engineering in 2015. He is currently an E.I.T. and works for SynchroGrid as a Software Engineer. He writes software applications in various programming languages, focusing on simplifying relay setting development. In addition to programming, Luke oversees relay settings verification and has assisted with mis-operation analysis.

## **8. ACKNOWLEDGEMENT**

The authors would like to thank Haley Tribur for her contributions in making this paper possible.