Auto-tuned Solution to Wide Area Coordination Issues of Distance and Directional Time Overcurrent Relay Settings

Nathan Thomas
SynchroGrid
College Station, TX USA
nthomas@synchrogrid.com

Luke Hankins
SynchroGrid
College Station, TX USA
lukeh@synchrogrid.com

Joe Perez
SynchroGrid
College Station, TX USA
jperez@synchrogrid.com

Abstract—**Relay coordination is an extremely difficult, yet vital part of a comprehensive protection strategy for modern power systems. With the recent introduction of PRC-027-1 [14] and its requirement for coordination to be reevaluated at regular intervals, innovation to reduce the burden and resources required for this activity is essential. Achieving coordination and ensuring that relays operate in a predictable manner can be quite a challenging activity, especially in highly coupled power systems with tight loops in the topology structure. In [3] we presented an early prototype of a *coordination autotuner* framework and demonstrated its use for the automatic generation of tuned pickup and time dial settings for direction time overcurrent relays on a mix of synthetic and real world grids.**

In this paper, we present improvements we have made to our autotuner framework, moving us significantly closer to a general purpose coordination autotuner, capable of performing the mundane, iterative work required during coordination studies, with guidance and feedback from the protection engineer tasked with the effort. We focus on several key areas including support for additional coordination constraints (for implementing specifics of a utility's standard), allowing more tunable parameters (e.g., overcurrent curve) and alternative methods to calculate them, fault studies with contingencies, and incorporation of distance element responses. We show usage of each in the experiments, demonstrating how they allow us to better support common cases encountered in real world coordination studies. Together, these new capabilities address a large number of simplifying restrictions in our previous work, making us increasingly confident that autotuning-assisted coordination studies is a viable and important advancement quickly coming on the horizon in system protection.

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

I. INTRODUCTION

Wide area coordination is one of the more difficult yet critical aspects of system protection that ensures relays operate in a deterministic and appropriate manner in the presence of their neighboring relays. Just as with new settings creation, the coordination problem is made more complicated by the fact that there are many approaches and utility standards to approach it, reasonably varying due to the nature of the given power system model, protection equipment, and other factors. Hence, any attempt to provide assistance to protection engineers in the form of automation must be flexible enough to adapt to a wide variety of applications.

Coordination is made even more difficult with modern microprocessor relays that have multiple protective elements' individual responses combined to define the overall response of the relay. Hence, while directional time overcurrent elements are generally considered much more difficult to coordinate than distance elements, a proper coordination must *integrate* analysis of both element types. This is necessary both to ensure proper coordination, but also to identify potential coordination solutions that otherwise might not be found with strictly separate overcurrent to overcurrent and distance to distance coordination studies (see Section IV-C for further discussion).

This complex activity of relay coordination must be performed on an ongoing basis due to the dynamic and ever-changing nature of the electrical power grid. The addition (or removal) of generation sources, new transmission lines, as well as other equipment changes means a once-coordinated system may no longer be. NERC recognized the necessity of ensuring proper coordination, addressing it in Requirement 2 of PRC-027-1 [14], which is now going into effect. With the complexity and increasing frequency of the studies, the industry is in need of innovation to streamline compliance, reduce the burden on engineering staff, and minimize the resources necessary for utilities of all sizes to achieve proper coordination. To that end, we are developing a coordination *auto-tuner* that automatically creates candidate coordinated settings based on coordination criteria defined by the engineer and automated analysis of the short circuit model.

We previously [3] presented an early version of this autotuning framework in which we focused solely on directional time overcurrent elements under very basic conditions (e.g., normal conditions) and only created settings for pickup and time dial settings. We built upon previous work in the field [4], [6]–[9], beginning with a previous, non linear optimization formulation of the problem [6] based on short-circuit analysis enabled by [1] and then transforming the problem to a mixed integer linear formulation as described in [9]. The formulation is then simplified further based on adaptations made specifically for the domain of system protection.

The goal of the work described in the paper is to improve the capabilities of the prototype, identifying and implementing features that may be potential roadblocks to the autotuner's use in real world scenarios. It is based on feedback from clients as well as obvious extensions (e.g., contingencies) necessary to achieve our vision of an automated coordination assistant which dramatically reduces the engineering burden for this activity. We describe each improvement in turn and provide experiments to demonstrate their usage.

The outline of this paper is as follows: In Section II, we provide a brief overview of our autotuning framework and then describe the common experimental setup (Section III) that will be used throughout the paper. The bulk of the paper lies in Section IV, where we discuss framework generalizations including expanded constraint support (IV-A), additional tunable parameters such as the overcurrent curve (IV-B), incorporation of distance element response (IV-C), and then contingency support (IV-D). In Section IV-E, we discuss the framework's handling of cases when full coordination is not possible and its ability to selectively recommend targeted relaxations of the coordination criteria to inform judgement decisions that the engineer inevitably will need to make. Finally, we present conclusions and discuss future work.

II. FRAMEWORK OVERVIEW

We briefly describe our autotuning framework, which is depicted in Figure 1. At *Project Inception*, coordination criteria are defined along with any additional tuneable parameter constraints (such as those described in Section IV-A), and passed to the *Settings Selection Framework*. Here, the coordination process begins by gathering topological information (coordination pairs, source/remote lines, etc), as well as fault currents, settings reach, and more from the short circuit model. This is accomplished using the algorithms and infrastructure presented in [1]. Next the *Settings Generator* takes the coordination criteria and data derived from short circuit model as input. It then computes candidate settings which are then verified by the *Coordination Verifier* as in [2]. These settings are then presented to the engineer for review and potentially further analysis leading to problem refinement and re-invocation of the framework.

In this paper, we make generalizations in the implementation of the *Settings Generator* to support new features, and the *Parameter / Constraint Definition* interface (still text based for now, but in the future graphical) is expanded to allow the engineer to drive these new features. Finally, the *Input Collector* now supports faults with contingencies as well as retrieving the reach and operation time of distance relay elements.

III. EXPERIMENTAL SETUP

For power system modeling, fault simulations, and coordination checking, we used ASPEN OneLiner V14.8 with OlxAPI (ASPEN's interface that our framework utilizes to directly interact with the short circuit model). For topological analysis and fault study generation, the core library of SARA v3.0.8 (based on [1], [2]) was employed. The autotuning framework itself is implemented in a modern C++20 codebase [22], compiled with Microsoft Visual Studio 2019 v16.6.0. All studies were run on a laptop based on a quad core Intel Core i7-8565U and containing 16GB of RAM.

All relays tuned are ground relays with single line to ground faults being used in fault studies. Both the *settings generator* and *coordination verifier* are directed to ensure coordination for a fault study regime including close-in, close-in end-opened, line-end, remote bus, and intermediate faults at every 10% of the line. When included, contingencies considered are discussed in Section IV-D. We require that no CTI violates the defined threshold ($CTI >= 0.33s$). Unless otherwise noted, the auto-tuner is directed to prefer coordination solutions that minimize the sum of each relay's response to a close-in end-opened fault on its primary line.

IV. GENERALIZING THE AUTOTUNING FRAMEWORK

In this section, we present the various extensions that we have made to our framework towards a goal of a general purpose tool for automated coordination. Though they vary in scope and implementation difficulty, each represents a critical area of improvement for the autotuner. We describe each in turn and provide experimental demonstration of how they are used in the framework.

A. Expanded Constraint Support

In [3], the primary constraint imposed on the *Settings Generator* was that of a minimum coordination time interval ($CTI >= 0.33s$). To ensure a realistic reach of the time overcurrent element, we also required that it respond to all faults on its primary line of protection, including a line-end fault. Besides lower and upper bounds on the tuned parameters of pickup and time dials, no other constraints were supported in the first version of our framework.

Of course, such limitations on constraining the solutions in the global coordination problem space are infeasible for real world applications. Utilities' protection standards regularly contain additional requirements on relay settings to be coordinated. For example, the response time of time overcurrent elements to faults is often artificially increased by utilities to ensure that it does not respond too quickly to some faults, often in anticipation of other elements (i.e. distance) of the relay responding to it first.

We are integrating a more customizable framework for constraint specifications such as the one mentioned above in our framework. Using a small symbolic language and basic mathematical operations on terms such as fault types and response times, basic constraints can be specified and translated into additional mixed integer inequality statements which are passed along with the CTI expressions and other constraints to the numerical solver. Both the essence and the implementation of this language draw from a similar technology we developed in [1], where it is used to allow a very flexible specification of transmission line relay settings equations which are translated into both topology analysis and fault simulations in short circuit programs such as ASPEN OneLiner and CAPE.

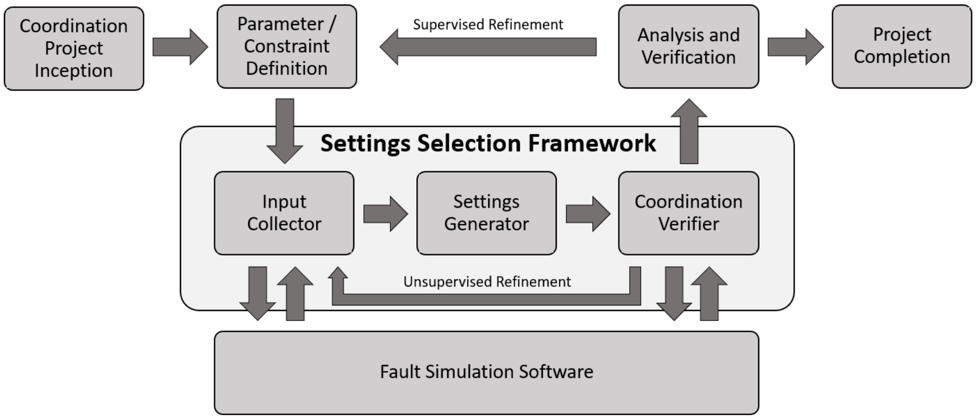

Fig. 1. Workflow for coordinated settings creation.

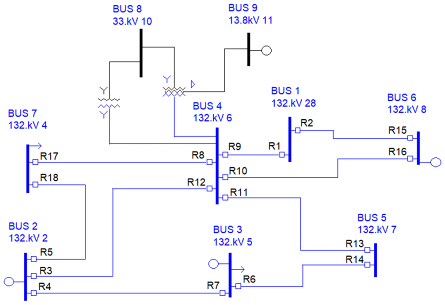

Fig. 2. One line diagram for 9-bus system.

To demonstrate the usefulness of this generalized constraint system, we define the example constraint described above, namely $PrimaryRelayTocTime_{LE\_Fault} >= 0.5$, and run the autotuner on the 9-bus power system shown in Figure 2, which is derived from a sample power system model that ships with ASPEN. Although relatively small in nature, it already represents a significant amount of work to develop and coordinate relay settings for. In this case, pickups were allowed to be in the range $I_{set} \in [1...16]$ and time dials could vary as $TMS \in [0.5, 15]$. These ranges are based on those allowed for relay configuration in an SEL 421 Relay [18]. The CT ratios were set in a straightforward manner, targeting a 20 amp secondary fault current for a close-in fault. For this study, we fix the overcurrent curve to ANSI U3 (very inverse).

In Table I, we show the baseline results without any constraint placed on line-end fault response time, showing the pickup, time dial and line-end fault response (T_{LE}) for each relay. Note, for example, the response time to $R10$ of 0.1865 seconds (11 cycles), which would cause it to respond well before the typical 20 cycle response time of the relay's Zone 2 element. In the presence of multiple solutions (common), the tuner is instructed to select the one which minimizes all relays' aggregate response time to close-in end-opened faults (in this case a total of 3.137 seconds for all 18 relays), which generally causes it to also minimize response time to line-end faults. These results are consistent with our previous findings.

TABLE I
CALCULATED SETTINGS AND RESPONSE FOR 9-BUS SYSTEM (BASELINE)

Relay	I_{set}	TMS	T_{LE} (s)
R1	7.8169	2.6402	0.3497
R2	6.7823	2.0038	0.4038
R3	15.0000	0.7609	0.4013
R4	15.0000	1.0987	0.5502
R5	15.0000	1.4380	0.3416
R6	15.0000	1.6069	0.5482
R7	14.2632	0.5099	0.3266
R8	9.8862	0.5000	0.6574
R9	15.0000	2.7469	0.4403
R10	15.0000	1.1459	0.2087
R11	13.5263	0.5000	0.1865
R12	15.0000	1.9859	0.5056
R13	15.0000	1.4023	0.3756
R14	15.0000	0.5000	0.3825
R15	15.0000	0.5000	0.4620
R16	12.0526	0.5000	0.5277
R17	15.0000	1.0837	0.6719
R18	12.7895	0.5000	0.4463

We next consider the constrained case, where we do not allow solutions with any relay responding to a line-end fault in less than 0.5 seconds. We also included a similar upper bound constraint, requiring that all relays respond to the same fault in no more than 1.25 seconds, similar to requirements commonly found in utility coordination standards. These constrained results are shown in Table II. Now, all line-end response times fall within the specified bounds. Aggregate response times to close-in end-opened increase approximately 1 second (4.142 seconds), an expected increase to satisfy the constrained problem definition.

One concern that we had as we began to implement new functionality and add complexity to the framework was that its execution time would increase to an amount deemed too great for practical use on large grids. While there is a large amount of work ahead to develop an accurate performance model and optimize performance, observations from even this simple addition are promising. The baseline case converged to an optimal solution in 0.4 seconds. Somewhat surprisingly, the more complex case with line-end constraints added took

TABLE II

Calculated Settings and Response for 9-Bus System (Line End Constrained)

Relay	I_{set}	TMS	T_{LE} (s)
R1	7.8169	3.7752	0.5000
R2	6.7823	2.4815	0.5000
R3	15.0000	0.9609	0.5068
R4	15.0000	1.2026	0.6023
R5	15.0000	2.1051	0.5000
R6	15.0000	1.8178	0.6201
R7	15.0000	0.7068	0.5000
R8	11.0710	0.5000	0.8784
R9	15.0000	3.1193	0.5000
R10	15.0000	2.7453	0.5000
R11	15.0000	1.1308	0.5000
R12	15.0000	2.3653	0.6021
R13	15.0000	1.8665	0.5000
R14	15.0000	0.6536	0.5000
R15	15.0000	0.5411	0.5000
R16	13.5263	0.5157	0.7154
R17	15.0000	1.1888	0.7371
R18	13.5263	0.5000	0.5047

half as long to find an optimal solutions (0.2 seconds).

Upon investigation of this speedup, it appears that in this case the branch and bound algorithm [19]–[21] used by our solver (and most linear optimization libraries) is able to more effectively *prune* the search space in the presence of this more precise description of what solutions in the problem space are actually viable in real world usage. This may not always be the case and we discuss potential strategies to deal with this in Section IV-D.

The example shown here is admittedly a simple constraint but is indicative of the types required by standards and demonstrates the ability to drive customization in the autotuning framework at a level of abstraction/expression intuitive to the protection engineer. Given sufficient time and resources, we anticipate incorporating this into a graphical user experience for streamlined coordination, similar to that based on the previous work in [1] and [2].

B. Wider Support for Tunable Parameters

We have expanded our framework's capabilities to support more tunable parameters than just the time dial and pickup. The first application of this expanded framework has been to tune the time overcurrent relay's curve parameter. While many utilities' standards recommend a specific curve parameter (e.g., ANSI U3) to be used in most relay setting applications, they typically allow deviation from this default to a more inverse curve to assist in more difficult coordination cases.

Intuitively, the approach is a straightforward extension of the relay response time modeling performed in the CTI constraints, adding an additional, tunable discrete parameter with one value for each possible curve considered for each relay. In addition to the standard ANSI and IEC curves, the framework supports vendor specific curves, and these are added as needed in usage of the autotuner with client's power system models. Just as with pickups and time dials, any subset of supported curve parameters can be specified for use by the autotuner, including the automatic population of these choices on a per relay basis, based on the relay type found during the topological inspection of the system model.

TABLE III

Calculated Settings and Response for 9-Bus System (Line-End Constrained) with Curve Selection (U1-U5)

Relay	$Curve$	I_{set}	TMS	T_{LE} (s)
R1	U3	7.8169	3.7752	0.5000
R2	U4	6.7823	2.6467	0.5000
R3	U4	15.0000	0.7516	0.5000
R4	U4	15.0000	0.7983	0.5000
R5	U4	15.0000	2.0698	0.5000
R6	U4	15.0000	1.3798	0.5422
R7	U4	15.0000	0.5386	0.5000
R8	U4	9.2938	0.5073	0.7880
R9	U4	15.0000	3.8846	0.5000
R10	U4	15.0000	3.1128	0.5000
R11	U4	15.0000	0.9248	0.5000
R12	U4	15.0000	1.9444	0.5182
R13	U4	15.0000	1.7486	0.5000
R14	U4	15.0000	0.5000	0.5062
R15	U4	13.5263	0.5096	0.5000
R16	U4	11.3158	0.5260	0.6499
R17	U4	15.0000	0.7665	0.6136
R18	U4	12.0526	0.5000	0.5228

We begin by revisiting the 9-bus experiment discussed in the previous section, employing the same experimental setup for the *line-end constrained study* (only line-end response times between 0.5 and 1.25 seconds allowed), but this time allow each relay to be configured with any of the ANSI U1 through U5 curves. In Table III, we show the results from this experiment in similar form to previous results but with the selected curve parameter now displayed for each relay. Note that U4 is picked for all but one relay, causing more relays' line-end response time to approach 0.5 seconds and reducing the aggregate close-in end-opened response time by about 27% to 2.99 seconds.

While improvements such as those gained in these cases may be deemed insufficient to deviate from a utility's philosophy, the ability to rapidly run such "what if" scenarios (the autotuner still runs in less than 1 second), inspect the magnitude of improvement in protection system responsiveness, and make an informed decision based on quantitative analysis is very powerful. There may also arise cases where a single curve's usage does not only return suboptimal solutions from the autotuner, but instead is proven not to lead to any correctly coordinated solution. We discuss one such situation next.

Consider a case where a utility places a stricter requirement on the minimum allowed response time constraint in our running example on the 9-bus system. If the constraint is changed slightly to $PrimaryRelayTocTime_{LE\_Fault} >= 0.7$ and only a U3 curve is allowed, the autotuner quickly reports that no solution can be found and will also report which constraints are causing difficulties (discussed in more detail in Section IV-E). In Table III, results are shown where the autotuner is able to find a solution to this more strictly constrained problem by considering ANSI U1 to U5 curves, remaining relatively responsive by matching the aggregate

TABLE IV

CALCULATED SETTINGS AND RESPONSE FOR 9-BUS SYSTEM (LINE-END CONSTRAINED, ≥ 0.7) WITH CURVE SELECTION (U1-U5)

Relay	$Curve$	I_{set}	TMS	T_{LE} **(s)**
R1	U2	7.8169	2.9733	0.7000
R2	U4	6.7823	3.7053	0.7000
R3	U4	15.0000	1.0522	0.7000
R4	U4	15.0000	1.1177	0.7000
R5	U4	15.0000	2.8977	0.7000
R6	U4	15.0000	1.7813	0.7000
R7	U4	15.0000	0.7541	0.7000
R8	U4	10.4786	0.5000	1.0581
R9	U3	15.0000	4.3670	0.7000
R10	U4	15.0000	4.3579	0.7000
R11	U4	15.0000	1.2948	0.7000
R12	U4	15.0000	2.6267	0.7000
R13	U4	15.0000	2.4481	0.7000
R14	U4	15.0000	0.6914	0.7000
R15	U4	15.0000	0.5623	0.7000
R16	U4	12.0526	0.5425	0.7796
R17	U4	15.0000	0.8905	0.7129
R18	U4	13.5263	0.5112	0.7000

response time to close-in end-opened faults established in the 0.5 second line-end constrained case.

Additional Tunable Parameter Improvements

We have made additional tunable parameter improvements to the autotuner, but space and time do not not allow us to present experiments. First, we have implemented support for automatic CT ratio selection, using an approach similar to that for curves. While it is not sensible to use this broadly in a wide area coordination study, targeted enabling of this improvement when changes may already need to be made (e.g., a potential for saturation has been found in pre-coordination model analysis) has been shown to provide more flexibility to the autotuner to find solutions.

Another improvement is support for alternative pickup parameter selection strategies. Originally, the autotuner took a minimum and maximum pickup value and uniformly sampled the range for a fixed number of candidates (default 20). The autotuner then ensured that a pickup was chosen that would respond to all faults on the line where it served as a primary or backup relay. Many utilities use a more direct approach, choosing a pickup by using the current obtained from a fault study (usually line-end or remote bus fault) and applying an appropriate scaling factor. These types of pickup values can be set using the approach described in [1] and integrated into the autotuner's *Parameter Definition* input (Figure 1).

C. Incorporating Distance Element Response

Distance elements are relatively simple to coordinate compared to overcurrent elements. With a fixed response time, their behavior is much easier to reason about, hence their popularity in system protection. Supporting distance / distance coordination (i.e., backup distance element coordination with primary distance element) can be handled by a streamlined version of the current overcurrent tuner implementation. Hence, for our initial investigation into distance elements, we chose to incorporate existing distance element relay settings into the framework and implement *heterogenous element coordination*.

Consider the case when overcurrent / overcurrent coordination is deemed infeasible, such as with a short source line containing a backup relay. While not ideal, one approach to handle this problem is to ensure overcurrent / distance coordination, namely that the instantaneous operation of Zone 1 on the primary relay provides sufficient CTI to overcome any timing violations with the local overcurrent element.

We have extended the framework to cover this case and tested it on a real world, interconnected power system located in Texas (depicted in Figure 3). Pickup and time dial settings ranges are the same as with the 9-bus case, and the CT ratios for the relays are the same as those of the relays currently in use for this power system. The autotuner is allowed to consider U1 to U5 curves and required to have minimium line-end fault response time of 0.25 seconds and maximum of 1.25 seconds. With these constraints, the autotuner's solver determines that there is no solution for completely coordinated solutions (though we will see in Section IV-E there may be solutions that an engineer deems acceptable). However, by instructing the autotuner to consider Zone 1 responses, a fully coordinated solution is found in 3.45 seconds and is shown in Table V.

TABLE V

CALCULATED SETTINGS AND RESPONSE FOR REAL WORLD SYSTEM: LOCAL ZONE 1 RESPONSE INCORPORATED

Relay	$Curve$	I_{set}	TMS	T_{LE} **(s)**
R1	U4	10.4737	0.5331	0.2500
R2	U4	8.3252	0.5128	0.3533
R3	U2	16.0000	0.6185	0.4172
R4	U4	14.4211	1.0816	0.2500
R5	U4	16.0000	0.5000	0.2505
R6	U4	15.2105	0.5390	0.3280
R7	U1	1.0000	2.5028	0.3743
R8	U2	16.0000	0.5574	0.4558
R9	U4	14.0110	0.5000	0.2832
R10	U4	16.0000	0.5915	0.2500
R11	U1	1.0000	3.3123	0.6401
R12	U1	1.0000	3.3205	0.6476
R13	U1	1.0000	1.7103	0.2500
R14	U1	15.2105	1.6030	0.9416
R15	U1	1.0000	6.3453	1.1163
R16	U1	1.0000	7.8298	1.2102
R17	U1	1.0000	6.4754	1.1701
R18	U2	4.6165	0.5000	0.7269
R19	U2	4.0140	0.5000	0.5852

The implementation of this feature is relatively straightforward. Distance element response times are captured as part of the fault study performed on the short circuit model, and this time is compared to that of candidate overcurrent parameter values and used if the modeled overcurrent operation time will result in a CTI violation. While there remains a fair amount of work to fully incorporate distance elements into the autotuner, this current addition demonstrates the viability of the autotuner to model more sophisticated actions taken by protection engineers during a coordination study.

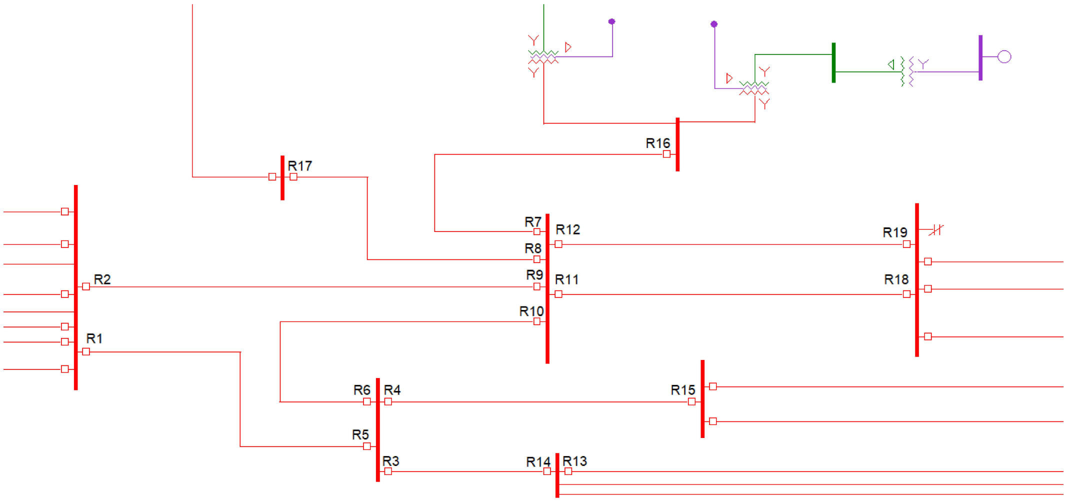

Fig. 3. One line diagram for real world system.

D. *Contingency Support*

Coordinating in the presence of contingencies is a critical feature for our autotuner to be of use in real world applications. Our short circuit analyzer in [1] has already supported this feature for some time; hence it was only a matter of implementing support of extending the *Settings Generator* to support them. Contingencies are implemented as additional constraints as described in Section IV-A, treated uniformly as the normal condition case and for all fault studies requested in the coordination specification. Results on the real world grid are shown in Table VI. The incorporation of distance element responses from the previous section remains enabled, and the experimental setup is the same, except that all remote lines are now taken as contingencies (source lines, transformers, and generation are also supported uniformly by the framework).

TABLE VI
CALCULATED SETTINGS AND RESPONSE FOR REAL WORLD SYSTEM: REMOTE LINE CONTINGENCIES AND LOCAL ZONE 1 RESPONSE INCORPORATED

Relay	*Curve*	I_{set}	TMS	T_{LE} **(s)**
R1	U4	12.8421	0.5000	0.3567
R2	U4	10.7253	0.5000	0.6048
R3	U4	16.0000	0.6116	0.3097
R4	U4	16.0000	0.8982	0.2500
R5	U4	7.3158	1.9739	0.2500
R6	U4	7.3158	2.1365	0.3381
R7	U4	13.6316	0.5352	0.2500
R8	U4	16.0000	0.7092	0.4559
R9	U4	12.8322	0.5273	0.2500
R10	U4	14.4211	0.7226	0.2500
R11	U4	5.3412	0.5000	0.2538
R12	U4	5.0941	0.5104	0.2500
R13	U4	15.2105	0.5290	0.2500
R14	U3	6.5263	3.7403	0.8043
R15	U2	1.0000	4.9669	0.9358
R16	U2	8.8947	1.7022	0.7302
R17	U2	4.9474	1.7023	0.7491
R18	U4	3.0054	0.5069	0.2500
R19	U4	2.9380	0.5000	0.2523

The results are similar to the previous section, but clearly have changed in the presence of contingencies. One interesting note is a marked increase in execution time to approximately 6 minutes on a laptop testbed. While this is still relatively responsive, we are looking into several approaches to reducing execution time. First, the solver does not return after the first viable solution is found, but instead continues until it has a provably optimal solution. We have observed that this optimization phase easily exceeds half of total execution time and is likely unnecessary as long as all coordination constraints are satisfied. Next, we have several domain specific strategies to identify constraints that are most likely to be limiting conditions and can prioritize their resolution through directives to the solver. Finally, we have yet to employ multi-threading / parallelization in the solver, a common technique used in these types of applications to dramatically improve performance.

E. *Permitting Minor Violations*

In large wide area coordination studies, there will inevitably be times when complete coordination is not possible. The engineer attempts to find the best coordination solution possible and then must judge whether the remaining violations are acceptable. We have implemented a technique in the autotuner similar in spirit to this approach. If the autotuner proves that complete coordination cannot be achieved, it attempts to find a solution that minimizes the number and size of violations, returning this solution along with a report of the violated constraints to the user for review.

We revisit the unsolvable real world grid configuration from Section IV-C, disabling distance response inspection and increasing the minimum line-end fault response time requirement to 0.5 seconds to further exacerbate the problem. The autotuner once again reports complete coordination is not possible but then suggests the settings in Table VII and reports the concessions made to create them. First, two line end minimum response time violations are identified (and are

TABLE VII
Calculated Settings and Response for Real World System: Coordinated with Minor Violations

Relay	$Curve$	I_{set}	TMS	T_{LE} **(s)**
R1	U4	12.8421	0.5269	**0.3759**
R2	U4	10.1564	0.5000	0.5332
R3	U4	16.0000	0.9874	0.5000
R4	U4	16.0000	1.5628	**0.4350**
R5	U2	13.6316	0.9487	0.5000
R6	U4	16.0000	0.8162	0.5521
R7	U1	1.0000	3.3430	0.5000
R8	U3	15.2105	1.0677	0.5000
R9	U4	14.0110	0.8828	0.5000
R10	U2	16.0000	0.8524	0.5000
R11	U1	1.0000	4.4004	0.8503
R12	U1	1.0000	4.4113	0.8603
R13	U4	16.0000	0.9558	0.5000
R14	U3	8.8947	3.6903	1.1911
R15	U2	1.0000	7.9612	1.5000
R16	U2	10.4737	2.8255	1.5000
R17	U1	1.0000	8.3010	1.5000
R18	U2	3.6648	0.5412	0.5000
R19	U1	3.6967	1.1860	0.6048

denoted in bold red in the table). Next, six CTI violations are reported, spread across three primary relays as shown in Table VIII. These violations are very minor, and the engineer is now equipped with empirical data to decide whether to allow them or deem the overcurrent to distance coordination resolution previously described as preferable.

TABLE VIII
CTI Violations for Real World Model with Line-End Fault Response Time ≥ 0.5s

Primary Relay	**Fault Type**	**CTI**
R1	Close In EO	0.2767
R1	Close In	0.2637
R1	Interm. 10%	0.2988
R2	Close In EO	0.3000
R2	Close In	0.2933
R4	Close In	0.3247

We believe the ability of the autotuner to function in the presence of incomplete coordination is a key and innovative addition to the framework, and we are continuing work to refine this feature. For example, not all violations are equal, so we are implementing a weighting scheme which will allow the engineer to direct the autotuner to prefer some violations (N-1/N-2) if necessary and avoid others (N-0).

V. Conclusions and Future Work

In this paper, we described a series of significant improvements to our coordination autotuner framework, making it a much more viable automation tool for use in real world system protection activities. We described the implementation of these additions and presented experimental results showing how they address common issues in system coordination. Additional constraint support allows greater criteria customization, and expanded parameter tuning capabilities increase the flexibility of the tool to find coordinated settings. The incorporation of distance elements and contingency support make the tool able to perform much more realistic coordination scenarios. Finally, coordination with minimized violations represents a powerful new avenue for the autotuner to define and quantify the tradeoffs that must often be made in protecting real power systems.

We have already discussed next steps in some of the preceding sections. In addition to these, we want to continue to run the autotuner on more real world grids, as this identifies additional areas for improvement and provides invaluable feedback on how the tool can be used by engineers. As we continue to improve its capabilities and generalize the input specification, we eventually envision a graphical user interface to initiate studies and visualize the potential solutions the auto-tuner creates. This will streamline interaction with the tool, making it possible for engineers to more quickly converge on better solutions for increasingly complex coordination applications.

References

[1] M.J. Boecker, G.T. Corpuz, J. Perez, and L. Hankins. “Novel approach to relay setting development.” 2017 70th Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2017.

[2] C. Thomas, J. Perez, L. Hankins, and H. Tribur. “An innovative and automated solution for NERC PRC-027-1 compliance.” 2018 71st Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2018.

[3] N. Thomas, L. Hankins and J. Perez, “Simplifying Wide Area Coordination of Directional Time Overcurrent Relays Using Automatic Settings Selection ” 2019 72nd Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 2019, pp. 1-6.

[4] J. Moirangthem, S. Dash, and R. Ramaswami. “System-wide protective relay setting and coordination in large-scale transmission systems-a review.” Journal of Electrical Engineering. Vol 11, 2011.

[5] D. Birla, R.P. Maheshwari, and H. Gupta. “Time-Overcurrent Relay Coordination: A Review.” International Journal of Emerging Electric Power Systems, 2005.

[6] A.J. Urdaneta, R. Nadira, and L.G. Perez Jimenez. “Optimal coordination of directional overcurrent relays in interconnected power systems.” IEEE Transactions on Power Delivery 3.3 (1988): 903-911.

[7] D. Birla, Dinesh, R.P. Maheshwari, and H. Gupta. “A New Nonlinear Directional Overcurrent Relay Coordination Technique, and Banes and Boons of Near-End Faults Based Approach.” IEEE Transactions on Power Delivery, 2006. 1176 - 1182.

[8] Y. Damchi, J. Sadeh, and M. Habib. “Applying Hybrid Interval Linear Programming and Genetic Algorithm to Coordinate Distance and Directional Over-current Relays.” Electric Power Components and Systems, 2016. 44. 1-12.

[9] Y. Damchi, M. Dolatabadi, H. Mashhadi, and J. Sadeh. “MILP approach for optimal coordination of directional overcurrent relays in interconnected power systems.” Electric Power Systems Research, 2018. 158. 267-274.

[10] F. Razavi, H. Abyaneh, M. Al-Dabbagh, R. Mohammadi, and H. Torkaman. “A new comprehensive genetic algorithm method for optimal overcurrent relays coordination.” Electric Power Systems Research, 2008. 78. 713-720.

[11] H. Kilickiran, B. Kekezoglu, and N. Paterakis. “Reinforcement Learning for Optimal Protection Coordination.” 2018 International Conference on Smart Energy Systems and Technologies (SEST), Sevilla, 2018, pp. 1-6.

[12] S. M. Madani and H. Rijauto, “A new application of graph theory for coordination of protective relays,” in IEEE Power Engineering Review, vol. 18, no. 6, pp. 43-45, June 1998.

[13] Thomas M. Mitchell. 1997. Machine Learning (1 ed.). McGraw-Hill, Inc., New York, NY, USA.

[14] “PRC-027-1: Coordination of Protection Systems for Performance During Faults.” (2015, November). Retrieved February 22, 2018, from North American Electric Reliability Corporation.

[15] “Performing Wide-Area Sensitivity Studies in CAPE.” Presented at the CAPE User’s Group Meeting: June 23-24, 2009.

[16] "System Wide Area Coordination Analysis." Presented at the CAPE User's Group Meeting: June 23-27, 2014.
[17] ASPEN Oneliner. https://www.aspeninc.com/web/software/oneliner.
[18] SEL-5030 acSELerator QuickSet Software. https://selinc.com/products/5030/
[19] A. H. Land and A. G. Doig. "An Automatic Method of Solving Discrete Programming Problems" Econometrica, vol. 28, no. 3, 1960, pp. 497-520.
[20] J. Little, K. Murty, D. Sweeney, and C. Karel. 1963. "An Algorithm for the Traveling Salesman Problem." Oper. Res. 11, 6 (December 1963), 972-989.
[21] D. Morrison, S. Jacobson, J. Sauppe, and E. Sewell. 2016. "Branch-and-bound algorithms." Discret. Optim. 19, C (February 2016), 79-102.
[22] ISO/IEC. (2020). "ISO International Standard ISO/IEC 14882:2020(E), Programming Language C++." [Working draft]. Geneva, Switzerland: International Organization for Standardization (ISO). Retrieved from https://isocpp.org/std/the-standard

VI. AUTHOR BIOGRAPHIES

Nathan Thomas earned B.S. and Ph.D. degrees from Texas A&M University in 1999 and 2012, respectively, both in Computer Science. He has an extensive background in high performance computing for large-scale engineering and scientific applications. He is also interested in machine learning and how it can be used to maximize system performance. Nathan cofounded and leads development at SynchroSoft, the software and automation division of SynchroGrid.

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 the area of relay setting automation. He writes software applications in various programing languages, focusing on simplifying relay setting development. Additionally, Luke oversees relay settings verification and has assisted with mis-operation analysis.

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.