DETERMINING ONE-SHOT CONTROL CRITERIA IN THE WESTERN NORTH AMERICAN POWER GRID WITH PARTICLE SWARM OPTIMIZATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Gregory AE Vaughan

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science of Electrical and Computer Engineering

May 2019

Purdue University

Indianapolis, Indiana

To my parents, George and Lisa, and to my sister Laura.

Engineering is attaining function whilst avoiding faults

ACKNOWLEDGMENTS

This thesis would have never happened without the guidance, support, and assistance of many individuals in and around the Indianapolis campus who contributed invaluable help in the development and completion of this thesis.

First and foremost, my thanks go out to my advisor Dr. Steven Rovnyak. Since I matriculated to IUPUI in 2014, he has been a steadying force in the engineering department and an advocate for me in the university. He has provided me experience and assistance in the classroom and in the laboratory, and was instrumental in developing this thesis problem, data, and solution.

I also am grateful to my advisory committee members Dr. King and Dr. Dos Santos for their time and support contributed to the successful completion of this thesis.

I would like to thank Dr. Ben Miled for her continued assistance and for guiding me through much of the preparatory work needed in the machine learning aspects of this thesis.

Of course, I would never have made it this far without the support and guidance of Sherrie Tucker, who made sure that I was always meeting my deadlines and helped me manage the lab sections I led.

I would also like to acknowledge several of the students who have made my time at IUPUI better, including but not limited to Nathaniel Cantwell, Sami Uzzama, Luke Pell, and Nathan Schwartz for all their help.

Last but not least, I would like to thank my parents who have given up so much of their time and supported me throughout my college education. They have been an invaluable resource for me as I made the transition from mathematics to engineering, and then into my graduate studies.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

c_1	cognitive multiplier
c_2	social multiplier
f	frequency
g	possible number of generator / fault combinations
m	possible number of time slices
$J_{\Delta f}$	moving average index
$J_{60/\lfloor N_s \rfloor}$	Fourier coefficient index
N	current time slice index
N_s	time divisions
$N_{s,1}$	first time division
$N_{s,2}$	second time division
T	thresholds
T_1	first threshold
T_2	second threshold
T_P	*df/dt* threshold for temporarily blocking controls
w	inertial weight

ABBREVIATIONS

AN	Add Negative (Admittance)
AP	Add Positive (Admittance)
FN	False Negative
FP	False Positive
ISGA	Integrated Square Generator Angle
LP	Linear Programming
MW	Megawatt
PMU	Phase Measurement Unit
PSO	Particle Swarm Optimization
RMSBA	Root Mean Square Bus Angle
RMSBA_A	RMSBA of add admittance event
RMSBA_D	RMSBA of generator disconnection event
RMSGA	Root Mean Square Generator Angle
ROCOF	Rate of Change of Frequency
TN	True Negative
TP	True Positive
WECC	Western Electric Coordinating Council

ABSTRACT

Vaughan, Gregory AE M.S.E.C.E., Purdue University, May 2019. Determining One-Shot Control Criteria in the Western North American Power Grid with Particle Swarm Optimization. Major Professor: Steven Rovnyak.

The power transmission network is stretched thin in Western North America. When generators or substations fault, the resultant cascading failures can diminish transmission capabilities across wide regions of the continent. This thesis examined several methods of determining one-shot controls based on frequency decline in electrical generators to reduce the effect of one or more phase faults and tripped generators. These methods included criteria based on indices calculated from frequency measured at the controller location. These indices included criteria based on local modes and the rate of change of frequency.

This thesis primarily used particle swarm optimization (PSO) with inertia to determine a well-adapted set of parameters. The parameters included up to three thresholds for indices calculated from frequency. The researchers found that the best method for distinguishing between one or more phase faults used thresholds on two Fourier indices. Future lines of research regarding one-shot controls were considered.

A method that distinguished nearby tripped generators from one or more phase faults and load change events was proposed. This method used a moving average, a negative threshold for control, and a positive threshold to reject control. The negative threshold for the moving average is met frequently during any large transient event. An additional index must be used to distinguish loss of generation events. This index is the maximum value of the moving average up to the present time and it is good for distinguishing loss of generation events from transient swings caused by other events.

This thesis further demonstrated how well a combination of controls based on both rate of change of frequency and local modes reduces instability of the network as determined by both a reduction in RMSGA and control efficiency at any time after the events.

This thesis found that using local modes is generally useful to diagnose and apply one-shot controls when instability is caused by one or more phase faults, while when disconnected generators or reduced loads cause instability in the system, the local modes did not distinguish between loss of generation capacity events and reduced load events. Instead, differentiating based on the rate of change of frequency and an initial upward deflection of frequency or an initial downward deflection of frequency did distinguish between these types of events.

1. INTRODUCTION

1.1 Problem Statement

Power generation and transmission in North America is affected by faults that occur anywhere in the transmission network [1]. Losing generation capacity or excessive load requirements can also cause catastrophic network-wide failures. The resultant blackouts and brown-outs cost more than 100 billion dollars of annual productivity [2]. When one or more phases short to ground due to lightning strikes or tree damage, or if a generator trips unexpectedly, the effect can adversely affect power transmission across wide swatches of the interconnection.

We begin with a simplified model of the electrical network of Western North America, as administered by the Western Electric Coordinating Council. The WECC model consists of a network containing 176 nodes called buses. There are also 29 generators connected to select nodes. A previous study [3] indicates that generators can be grouped into clusters that swing together during transient events. Thus, each of the 29 generators are located in one of four clusters. In this thesis, transient faults last no more than a sixth of a second, while disconnected generators can remain disconnected for hours or days.

We seek to find solutions to two related problems. The first problem is to detect whether the electrical network has a local fault that should be controlled by load-shedding without using a simulation to determine the effectiveness of those controls.

The second problem is to simulate the effects of the proposed controls on the 176-bus model and improve the control algorithms until they have a high level of success controlling the types of simulations that we are interested in and not controlling all other types of simulations. Previous methods [4] have been shown to order one-shot controls in a noiseless environment. We wish to design an algorithm that is also robust in noisy environments while still responding selectively to disconnected generators.

1.2 Background Material

Electrical generators create three-phase alternating voltage, which drives current that is carried on the transmission lines between buses. Under normal operating conditions, the phase between generators is synchronized. The nominal operating frequency of all generators in North America is 60 Hz.

Whenn a fault between two buses occurs, there is a change in magnitude, frequency, and phase in all generators' terminal voltage. Some faults cause the phase between generators to fall out of synchronization. We call these faults unstable. Faults that do not cause a synchronization failure are called stable. We label each fault as unstable or stable by looking at the maximum global difference in rotor angle between any of the 29 generators connected to various buses throughout the network.

We desire to apply one-shot load-shedding control to the network. This control will help the network avoid wide-spread instability if there is a tripped generator. The control decision and load-shedding will occur within individual consumer appliances such as household clothes dryers and air conditioners. We also must manage the cost of the hardware required to make the control decision. This means we must limit the computational complexity of any algorithm we use.

One current method to determine if we should apply a load-shedding control is to check if the average change in frequency goes over a hard-coded threshold during any time period. If the average change in frequency is less than some negative value, then we assume that load shedding should be applied. For short intervals, the moving average change approximates *df/dt* calculated from the point to point differences.

We assert that it is possible to determine whether or not a generator trip within the network is local or non-local by looking at the spectrum of frequency change. From this spectrum, we determine if the frequency data indicates a generator trip that is nearby the measurement location.

Rovnyak [5] indicates that,

> The presence of local modes in the frequency measurement at a particular location indicates a transmission level event has occurred nearby. Nearby in this context usually means the event occurs in the same or a neighboring state. The local modes observed so far are around 0.8 Hz whereas the whole interconnection mode is approximately 0.3 Hz.

We want to determine if 0.8 Hz is the best frequency for indicating a nearby event. If not, then we wish to find a better frequency or combination of frequencies which indicate a nearby event. We also want to determine the thresholds that are used to determine whether the event requires control. If the event requires control at any time slice at a given location, then we declare that the location requires a control.

Mei, *et. al.* [3], suggested a four-cluster representation of the WECC generators in which the generators in each cluster are more tightly synchronized with generators in their cluster than they are with generators in other clusters. In this thesis, we define 'nearby' to mean that a measurement location is nearby if the disconnected generator is in the same cluster as the measurement location. A controller designed to act in response to detecting a nearby loss of generation event is the *clustered mode* method.

Ultimately, we wish to ascertain what the best parameters are for both the *df/dt* method using indices calculated from frequencies. We also wish to determine whether any of the clustered mode methods are better than the *df/dt* method in selectively responding to nearby loss of generation events.

1.3 Related Works

1.3.1 Fault Detection

The literature in [4–13] describes other methods of finding the appropriate time to apply one-shot and consecutive controls. These include considering a swift decrease in phase, frequency, square bus angle, or bus voltage, and separately detecting both an event and an imminent loss of control. We closely review [7–13].

All of [4, 7–11] concern the use of *f* and *df/dt* to initiate load-shedding as a last resort to avoid blackouts and permanent turbine damage. In [12] and [13], the respective authors extend that research by attempting to apply various PSO methods to both find an optimal solution, and do so in an efficient, real-time manner.

The earliest work examined, Durkin, *et. al.,* [7] noted that commercially-available relays in 1969 allowed for either frequency cutoff or rate of change of frequency (ROCOF) cutoffs, but not both. Durkin designed an analog solid-state load-shedding relay that allows the user to easily hard-code *f*, ROCOF, and time delay thresholds.

More than two decades later, we see that Shih, *et. al.,* [8] discuss microprocessor-based relay that incorporates communication with the overall network controllers to allow for more aggressive load-shedding. This is an improvement on Durkin, *et. al.,* as it allows for operator control of load shedding and non-linear conditions in (*f*, *df/dt*) phase space.

Shih, *et. al.,* [8] also compared various settings, including both conservative and aggressive load-shedding based on only *f*, and also both *f* and *df/dt*. Shih determined that it was possible to allow for reduced load-shedding combined with a smaller maximum frequency drop for a given fault. This design incorporated all of the capabilities of the Durkin design and also showed computer simulations of several different relay settings.

We now turn our attention to specific applications of (*f*, *df/dt*) based under-frequency load-shedding in wide-area networks. Chuvychin, *et. al.,* [9] examined the Latvian regional network. The authors focus on fixing sequential faults in a network and implementing a non-economic method to resolve loss of generation issues.

Chuvychin, *et. al.,* asserted that sequential faults in a given network were common in the Latvian power grid, and that their load-shedding scheme was based on frequency level and multiple cascading faults. Electrical generators have both a fixed operating cost and a variable cost that increases faster than the increase in electrical output. Therefore, there is some electrical generation rate that minimizes the cost per monetary unit, so there is some generation level that is most economical.

Chuvychin, *et. al.,* [9] suggested that generators change from the most economic generation rate and increase to the maximum generation rate in an underfrequency emergency, and then return to normal economical operations once the state of the generators is restored. These two methods work together to automatically remediate under-frequency emergencies until operators stabilize the faulty generator(s) and manually reset the adaptive load-shedding relays.

Continuing in the theme of using *f* and *df/dt* to shed electrical load, You, *et. al.,* (2003) [10] consider clustered *f* and both unclustered and clustered *df/dt* to automatically break up the systems into load-rich or generator-rich islands. You asserts it is good practice to break networks into islands in order to isolate the contingency to a smaller portion of the network. Generator-rich islands have positive *df/dt* and are easily handled by governors, while load-rich islands require more consideration.

The principal finding of [10] was that generators with a relatively steep *df/dt* have less time to respond to a fault before the generators pass into a frequency region which permanently damages turbines. The authors then compared the clustered *f* and unclustered *df/dt* to determine whether they should use aggressive or conservative load shedding.

This approach attempted to match load shedding to the most immediate need. The authors modeled a simulation showing that this load-shedding scheme reduced maximum frequency deviations and quickly restored the generator to nominal operating frequency. This compared favorably to other methods which either took longer to restore the generator or exhibited catastrophic failure.

This is similar to the result in [8], but expanded the previous result because You, *et. al.,* restricted their focus to the local island, rather than observing the network as a whole. This was a superior approach as it ignored regions which were on different sides of the generator-surplus/generator-deficit line in (*f*, *df/dt*) phase space.

We consider another national network similar to [9], as Bevrani, *et. al.,* concentrated on the Australian national network in [11]. The impetus for the authors was the Australian network's disproportionate reliance on renewable energy resources. The authors restated previous works regarding under-frequency response of systems to generator loss.

In agreement with [8], they demonstrate that *df/dt* is proportional to the size of the loss of generation. Furthermore, load shedding at generators with large *df/dt* is advantageous compared to shedding load evenly across the network.

The primary contribution of [11] was an examination of a simplified network consisting of both fixed-output generators and solar- and wind-powered renewable energy sources. Bevrani found that taking the global average ROCOF $\Delta f/\Delta t$, which is our index $J_{\Delta f}$ averaged across k time slices was a better indicator of system stability than the backward difference *df/dt* over one time slice.

Bevrani also determined that controlling the fixed generators using a PI controller was insufficient to maintain acceptable frequency stability since the renewable energy sources are inherently variable. Therefore, the authors proposed load shedding as an "emergency control dynamic" when the slower PI system was not able to keep the average system frequency close enough to the nominal frequency. In similar fashion to the other papers reviewed here, Bevrani proposed load shedding as a method to re-establish frequency control when other methods are not fast enough to prevent under-frequency damage to the turbines.

Of immediate importance to this thesis is Nilchi's work on decision trees [4]. This work established that a maximum generator angle difference of 300 degrees is a good conservative estimate on the threshold between stable and unstable events. It further developed the Integrated Square Generator Angle (ISGA) index as the measure of network stress.

For equal numbers of generators and equal simulation times, our RMSGA is proportional to the square root of ISGA. In addition, Nilchi used point-to-point angle differences to calculate derivatives. We expanded this concept to moving averages of bus and generator angles for various window lengths.

1.3.2 PSO Techniques

Many of the previous methods for ordering one shot control in power systems were designed using decision trees. We examine two works that describe the current use of artificial intelligence, and specifically PSO in the field. In [12], Tarafdar Hagh and Galvani use both standard PSO as described by Eberhart and Shih [14] and linear programming (LP) followed by PSO. The authors assert that LP methods are fast and useful when solving linear problems, while PSO is good at solving general problems, but is slower than LP.

In particular, Tarafdar Hagh and Galvani compared the use of LP, PSO, and sequential LP-PSO methods to determine nearly minimal load shedding to a critical contingency as fast as possible on the standard IEEE 14-bus test system. The LP techniques found solutions that do not satisfy all of the constraints of the non-linear system, while PSO was slow to converge. The authors first carried out LP to find an approximate solution, and then generated particles near that solution and used PSO with a relatively small number of iterations to find a correct solution.

Recalling [12], we see that in [13], El-Zonkoly, *et. al.,* used comprehensive learning particle swarm optimization (CLPSO) to determine which buses to open or close. Using these buses, El-Zonkoly, *et. al.,* found the most useful intentional islanding. In particular, the key difference is that while Tarafdar Hagh and Galvani combined standard PSO with LP, El-Zonkoly, *et. al.,* instead used a comprehensive learning PSO combined with a complicated utility function that combined factors such as actual customer payments per MW and whether generators were owned by investors or public utilities.

CLPSO makes particles sometimes gravitate to the p_{best} of other particles rather than the particle's own p_{best}. The probability of gravitating to the particle's own p_{best} increases as the simulation runs. This prevents particles from setting into local minima early in the simulation.

As in [12], El-Zonkoly, *et. al.,* used a variant on PSO to decide a good set of parameters for a given problem. In [12], the problem was to find a set of new generator settings subject to multiple constraints that minimized the amount of load shed, while in [13], the problem was to find a useful set of opened or closed buses that minimized the cost of intentional islanding.

Trelea [15] addressed the question of convergence of a PSO with one parameter. Trelea told us that our use of $c_1 = c_2 = 2$ leads to oscillatory behavior of the particle around the mean of the particle best and best overall particle parameter value as long as the inertial weight w is between $3-2\sqrt{2}$ and 1, and that the method should converge as long as $0 < w < 1$. Further, Trelea showed that small w leads to fast convergence to some local minimum by searching in a narrow region, while a large w leads to slow convergence by searching over a wide area.

2. METHODOLOGY - PHASE I

2.1 Inputs

In Phase I, we wish to address the first problem of this thesis, that is, demonstrating that we can find a well-adapted algorithm to apply one-shot controls for transient faults so that we may order controls using similar algorithms in Phase II to both transient faults and disconnected generators. To that end, we used a test bed of 480 1ϕ faults and 480 3ϕ faults scattered throughout the system.

Our model gives voltages and angles at 120 samples per second for 6 seconds including the fault. This is a common test bed of the 176-bus WECC model used by several authors [4,5]. Each simulation returns the voltage magnitudes and rotor angles for all 29 generators before and after the fault.

2.2 Pre-Processing

We label each fault as stable or unstable by determining the global maximum difference in phase angle between any two generators at any time during the simulation. We use the generator angle data to determine if each fault is stable or unstable. We also know from the geographical map or cluster data [2] whether each generator/fault are in the same cluster or not.

We classify each generator / fault combination CONTROL if the fault is unstable and the fault is in the same cluster as the generator, and NOCONTROL otherwise. We will then train the agent with generator angle data and the CONTROL / NOCONTROL state. We then calculate Fourier indices at each discrete time slice during the simulation. We will use the Fourier indices to determine if a given generator / fault combination should be controlled.

2.3 Algorithms

We compare six algorithm classes, the *df/dt* class, the FT class, the 2FT+ class, the 2FT- class, the 2FT> class, and the 2FT< class. The *df/dt* methods vary the length of a moving average and a threshold value to determine if a fault requires control at any time slice. The *df/dt* class is similar to the class of methods tested in [4]. We denote the length of the moving average as N_s.

The one-frequency clustered mode (FT) methods vary a threshold value and the lengths of a convolution with both discrete sine and cosine functions to determine if the fault requires control at any time slice. We denote the length of the convolution as N_s,$N_{s,1}$, or $N_{s,2}$, as appropriate for the class. We examine the FT methods based on [5] as a method of distinguishing the magnitude of local oscillations present in the simulation from remote oscillations.

The 2FT-, 2FT+, 2FT> and 2FT< (two-frequency clustered mode) methods vary two convolution lengths and both threshold values, and then combine them in different values to determine if the fault requires control at any time slice. A generator / fault combination is positive if it requires control and is negative if it does not require control. Each method uses its parameters to label each combination as positive or negative. We simulate each combination to determine if the combination is actually positive or negative.

We also know that there is a cost associated with applying unnecessary controls and with not applying a control when it is necessary. We can assign a cost function that models the cost to false positives and false negatives as described in section 3.4.

2.4 Algorithm Justification

The goal of phase I was to determine if there was a best method in the FT class for distinguishing local, unstable faults from other faults. We considered that there might be multiple high Fourier indices associated with a generator / fault combination and tried to take advantage of this without knowing what the shape of the Fourier spectrum was. We decided that local and unstable generator / fault combinations might be signaled by two large Fourier indices. The 2FT> class checks for two Fourier indices to both exceed thresholds, while the 2FT+ class checks for two Fourier indices to exceed the threshold together.

Alternatively, it might be that even though one Fourier index is large, the presence of another large Fourier index at a particular frequency indicates a remote or stable generator / fault combination. The 2FT- and 2FT< classes apply CONTROL only if a large second index does not inhibit a large first index.

2.5 Complexity Analysis

We let n be the number of possible N_s values, g be the number of generator/fault combinations, and s be the number of particle evaluations required to carry out PSO. Furthermore, we know each *df/dt* and FT method consists of a (N_s, t) pair, each 2FT+ and 2FT- method consists of a $(N_{s,1}, N_{s,2}, t)$ triple, and each 2FT< and 2FT> method consists of a $(N_{s,1}, N_{s,2}, t_1, t_2)$ quadruple.

Table 2.1.: Evaluations per Class

Class	# of evaluations for exhaustive search
df/dt	$\Omega(n)$
FT	$\Omega(n)$
2FT+	$\Omega(n^2)$
2FT-	$\Omega(n^2)$
2FT>	$\Omega(n^2 g^2)$
2FT<	$\Omega(n^2 g^2)$

We can find the best *df/dt* and FT methods using exhaustive search by testing each of the n possible values of N_s and determining the optimal threshold for that N_s. Similarly, we can find the best 2FT+ and 2FT- methods using exhaustive search by testing each of the n^2 possible values of $N_{s,1}$ and $N_{s,2}$ and determining the optimal threshold for that $(N_{s,1}, N_{s,2})$ pair.

On the other hand, using exhaustive search to find the optimal $(N_{s,1}, N_{s,2}, t_1, t_2)$ combination for 2FT< and 2FT> requires testing each combination separately. The number of evaluations needed to find optimal parameters for each class via exhaustive search is summarized in Table 2.1.

Since $n \ll g$ and we observe that n^2 is of similar size to s, we use exhaustive search to find the optimal *df/dt*, FT, 2FT+, and 2FT- methods, while we use PSO to find well-adapted methods for the 2FT> and 2FT< classes.

2.6 Outputs

We wish to measure how unstable a given event is. To determine this, we adopt a variant of the ISGA index from [4] to evaluate the instability. This index is the root mean square bus angle (RMSGA). The RMSGA allows us to compare catastrophic failures of different amounts of generation in the simplified 176-bus WECC model.

We calculate the RMSGA as follows:

$$\text{RMSGA} = \sqrt{\frac{1}{TM}\int_0^T \left(\sum_i M_i(\delta_i(t) - \delta_{\text{coa}}(t))^2\right) dt}$$

where

$$M = \sum_i M_i,$$

$$\delta_{\text{coa}} = \frac{\sum_i M_i \delta_i(t)}{M}$$

T is the length of the simulation in seconds, M_i is the machine inertia of generator i, and $\delta_i(t)$ is the generator angle of generator i.

3. IMPLEMENTATION - PHASE I

3.1 Terminology

Previous research [4] on this problem has developed a test bed of 960 faults, 480 one-phase faults and 480 three-phase faults, where each fault reports angle data for 29 generators and 17 phase measurement units (PMUs). Furthermore, each fault is *unstable* if the angle difference between two generators is more than 300 degrees at any time during the simulation and *stable* otherwise.

Each fault is associated with a transmission line between two buses in our 176-bus network. Lastly, each bus and generator is in one of four clusters [3]. For each combination of generator and fault, the criteria we use for nearby is whether the fault and generator are in the same cluster.

3.2 Pre-Processing

We run the simulations for the 960 faults for six seconds at 120 samples per second and obtain generator angle data for each of the $960 \times 29 = 27840$ fault/generator combinations. We smooth and differentiate the fault/generator angle data for all times using a four-element finite impulse response filter to get 718 frequency values per fault. We denote these values as $f[N]$ for $1 \leq N \leq 718$.

3.3 Labels

A generator/fault combination should be labeled as CONTROL if the fault is unstable and the generator is in the same cluster as the fault, and NOCONTOL otherwise. Analysis of the data indicates that there are 3702 fault/generator combinations that should be labeled as CONTROL and 24138 fault/generator combinations that should be labeled as NOCONTROL. We can now define three types of methods:

- A *perfect* method correctly labels all 3702 CONTROL combinations as true positives (TP), and all 24138 NOCONTROL combinations as true negatives (TN).
- A *terrible* method incorrectly labels each CONTROL combination as a false negative (FN), and each NOCONTROL combination as a false positive (FP).
- A *lazy* method labels each fault/generator combination as NOCONTROL and no fault/generator combination as CONTROL, or *vice versa*.

A lazy method that labels all combinations as CONTROL will correctly give 3702 TP, while a lazy method that labels all combinations as NOCONTROL correctly gives 24138 TN. In this way, we set the cost of the two possible lazy methods equal. Therefore, the cost of a lazy method is halfway between a terrible method and a perfect method.

We use the generator frequency data and the CONTROL/NOCONTROL labels to develop six different classes of methods. Each of these classify a fault/ generator combination as a TP, TN, FP, or FN, and has a cost associated with the method.

3.4 Costs

For each method, we assign a cost to the method by assigning a cost to each label. If the method labels the fault/generator combination correctly, then the label has a zero cost. However, if the method labels the fault/generator combination incorrectly, then the label has a positive cost. We assign one fixed cost for each FN combination and a different fixed cost for each FP combination. The cost of a method is the total cost of the labels of all 27840 fault/generator combinations.

We normalize our costs so that a perfect method has cost 0, a terrible method has cost 1, and a lazy method has cost 0.5. Costs of $1/(2 \times 24138)$ for an FP combination and $1/(2 \times 3702)$ for an FN combination satisfy all of these conditions.

3.5 Classes of methods

We evaluate the various methods using exhaustive search and particle swarm optimization.

3.5.1 Exhaustive Search

We evaluate the *df/dt* and FT methods using exhaustive search. Each *df/dt* method has two parameters [5]. The parameters are integer N_s and non-positive real t. For each time N, a *df/dt* method takes a moving difference $J_{\Delta f,N_s}$ of frequencies calculated as

$$J_{\Delta f,N_s}[N] = \sum_{n=N+1-N_s}^{N} \frac{f[n]}{N_s} - \sum_{n=N+1-2N_s}^{N-N_s} \frac{f[n]}{N_s} \tag{3.1}$$

that is, it is the difference of the moving averages over the previous $2N_s$ values, where the frequency at any $n < 0$ is fixed at 60.

A *df/dt* method with parameters N_s and t labels a generator/fault combination as CONTROL if $J_{\Delta f,N_s}[N] < t$ for any N, and otherwise labels the generator/fault combination as NOCONTROL. The N_s index may be elided if it is understood from context.

We now have two lemmas, each of which are proved by induction:

Lemma 3.5.1 *If $\{a_i\}_{i=1}^{n}$ is a finite real sequence and t is real, then*

$$\bigvee_{i=1}^{n} (a_i < t) \iff \left(\min_{1\leq i\leq n} a_i\right) < t \tag{3.2}$$

Lemma 3.5.2 *If $\{a_i\}_{i=1}^{n}$ is a finite real sequence and t is real, then*

$$\bigvee_{i=1}^{n} (a_i > t) \iff \left(\max_{1\leq i\leq n} a_i\right) > t \tag{3.3}$$

Lemma 3.5.1 tells us that a *df/dt* method with parameters N_s and t label a generator/fault combination as CONTROL if

$$\left(\min_{1\leq i\leq 718} J_{\Delta f,N_s}[N]\right) < t \tag{3.4}$$

Therefore, we calculate each of the 27840 $\min_{1\leq i\leq 718} J_{\Delta f,N_s}[N]$ once and compare them to various threshold values t rather than checking all 718 time slices for each t. Further, we can find a best t for any N_s by first finding the 27840 minima, sorting them, and then testing a value of t between each consecutive pair of minima in the sorted list, as well as one t past each end of the list of minima. The (N_s,t) which yields the minimum cost is the best *df/dt* method.

Each FT method has two parameters. The parameters are integer N_s and non-negative real t. For each time N, we calculate the following column vectors of height $2N_s$:

$$v_C = \begin{bmatrix} \cos(1\pi/N_s) \\ \cos(2\pi/N_s) \\ \vdots \\ \cos(2N_s\pi/N_s) \end{bmatrix}, v_S = \begin{bmatrix} \sin(1\pi/N_s) \\ \sin(2\pi/N_s) \\ \vdots \\ \sin(2N_s\pi/N_s) \end{bmatrix} \tag{3.5}$$

$$v_{60} = \begin{bmatrix} 60 \\ 60 \\ \vdots \\ 60 \end{bmatrix}, F[N] = \begin{bmatrix} f[N-2N_s] \\ f[N-(2N_s-1)] \\ \vdots \\ f[N-1] \end{bmatrix} \tag{3.6}$$

Note that only $F[N]$ depends on the time slice N.

We also calculate

$$Y_s[N] = \frac{1}{2N_s} < F[N] - v_{60}, v_s > \tag{3.7}$$

$$Y_c[N] = \frac{1}{2N_s} < F[N] - v_{60}, v_c > \tag{3.8}$$

Where $< a,b >$ is the dot product of a and b. We now have

$$J_{60/N_s}[N] = \sqrt{Y_s^2[N] + Y_c^2[N]} \tag{3.9}$$

An FT method with parameters N_s and t labels a generator/fault combination as positive if

$$J_{60/N_s}[N] > t \tag{3.10}$$

for any N and otherwise labels the generator/fault combination negative.

Lemma 3.5.2 tells us that for an FT method with parameters N_s and t, we can label a generator/fault combination positive if

$$\left(\max_{1 \leq i \leq 718} J_{\Delta f, N_s} [N] \right) > t \tag{3.11}$$

As with *df/dt* methods, we can find a best t for any N_s by first finding the 27840 maxima, sorting them, and then testing a value of t between each consecutive pair of maxima in the sorted list, as well as one t past the end of the list. We declare that the (N_s, t) pair which yields the minimum cost is the best *FT* method.

3.5.2 Particle Swarm Optimization

The remaining methods use PSO with two or four parameters. In PSO, we try to minimize some function of a set of parameters by evolving the parameters around a feasible parameter space.

Let $(p_1, \ldots, p_m)$ be a set of parameters and $c(x_1, \ldots, x_m)$ be a real function from $\mathbb{R}^m$ into a subset of $[0,1]$. We initialize a set of n particles $(p_1, \ldots, p_m)_i$ and velocities $(v_1, \ldots, v_m)_i$ where i runs from 1 to n and

$$(p_1, \ldots, p_m)_i \in [i_1, I_1] \times \ldots \times [i_m, I_m]$$

and

$$(v_1, \ldots, v_m)_i \in [-(I_1 - i_1), (I_1 - i_1)] \times \ldots \times [-(I_m - i_m), (I_m - i_m)]$$

We also limit the magnitude of the velocity of each parameter as described in [14], using different maximum velocities for different method classes. We assign $\left(p_{\text{best},1}, \ldots, p_{\text{best},m}\right)_i \leftarrow (p_1, \ldots, p_m)_i$, evaluate $\text{cost}_i = c((p_1, \ldots, p_m)_i)$, and set g_{best} such that

$$\text{cost}_{g_{\text{best}}} = \min_{1 \leq i \leq n} (\text{cost}_i) \tag{3.12}$$

that is, g_{best} is an index of a minimal-cost $(p_1, \ldots, p_m)_i$.

We define $r(a,b)$ as a random variable with uniform distribution on $[a,b]$. We now iterate for K iterations. At each iteration, we set the inertia w linearly decreasing from 0.9 to 0.5. We then update each $(v_1,\ldots,v_m)$ and $(p_1,\ldots,p_m)$ using the formulas:

$$\begin{aligned}(v_1,\ldots,v_m)_i \leftarrow w \times (v_1,\ldots,v_m)_i + r(0,2)\left((p_{\text{best},1},\ldots,p_{\text{best},m})_i - (p_1,\ldots,p_m)_i\right) \\ + r(0,2)\left((p_{\text{best},1},\ldots,p_{\text{best},m})_{g_{\text{best}}} - (p_1,\ldots,p_m)_i\right)\end{aligned} \tag{3.13}$$

and

$$(p_1,\ldots,p_m)_i \leftarrow (p_1,\ldots,p_m)_i + (v_1,\ldots,v_m)_i \tag{3.14}$$

We then compare cost_i and $c\left((p_1,\ldots,p_m)_i\right)$. If $c\left((p_1,\ldots,p_m)_i\right) < \text{cost}_i$, then we set

$$(p_{\text{best},1},\ldots,p_{\text{best},m}) \leftarrow (p_1,\ldots,p_m) \tag{3.15}$$

and

$$\text{cost}_i \leftarrow c\left((p_1,\ldots,p_m)_i\right) \tag{3.16}$$

We also find g_{best} such that

$$\text{cost}_{g_{\text{best}}} = \min_{1\leq i\leq m}(\text{cost}_i) \tag{3.17}$$

The theory of PSO asserts that after some large number of iterations, the $(p_1,\ldots,p_m)$ may converge to some point with a locally minimal cost. Eberhart states that the PSO method finds well-adapted sets of parameters more quickly than other optimization methods such as simulated annealing or genetic algorithms [16]. Trelea [15] tells us that our choice of decreasing w means that early iterations will search widely for a global minimum, while later iterations will search narrowly to find the local minimum near the current average of the best particle value and the global best particle value.

3.5.3 2FT Methods

We examine two more classes of 2FT methods. The 2FT+ and 2FT- methods keep track of three parameters, $N_{s,1}$, $N_{s,2}$, and t. The cost function $c(N_{s,1},N_{s,2})$ is the c function in the generic PSO algorithm described above. This function returns $(1,t)$ if $\lfloor N_{s,1}\rfloor \notin [1,120]$ or $\lfloor N_{s,2}\rfloor \notin [1,120]$. Otherwise, the cost function returns the best cost and threshold as defined below.

The cost function labels each fault/generator combination CONTROL if

$$J_{60/\lfloor N_{s,1}\rfloor}[N] + J_{60/\lfloor N_{s,2}\rfloor}[N] < t_{N_{s,1},N_{s,2}} \text{ for } \textit{2FT+}$$

$$J_{60/\lfloor N_{s,1}\rfloor}[N] - J_{60/\lfloor N_{s,2}\rfloor}[N] < t_{N_{s,1},N_{s,2}} \text{ for } \textit{2FT-}$$

for any $1 \leq N \leq 718$ and NOCONTROL otherwise. The cost function then sums the cost of each mislabeled combination and applies Lemma 3.5.2, so it can compare thresholds to the maximum of the sum or difference function over all time slices.

The cost function sequentially tests thresholds between consecutive pairs of values in the sorted list of

$$\left(\max_{1\leq N\leq 718}\left(J_{60/\lfloor N_{s,1}\rfloor}[N] + J_{60/\lfloor N_{s,2}\rfloor}[N]\right)\right) \text{ for 2FT+}$$

$$\left(\max_{1\leq N\leq 718}\left(J_{60/\lfloor N_{s,1}\rfloor}[N] - J_{60/\lfloor N_{s,2}\rfloor}[N]\right)\right) \text{ for 2FT-}$$

In this manner, the function c described above finds and returns the lowest cost and lowest-cost threshold for any $(N_{s,1}, N_{s,2})$.

We use PSO with 10 particles and 300 iterations. Each particle consists of the three parameters $p_{N_{s,1}}, p_{N_{s,2}}$, and p_t. The velocity maxima on $N_{s,1}$ and $N_{s,2}$ are set to 119, the difference between the maximum and minimum N_s values.

We initialize each particle and velocity using the following procedure:

```
**for all** particles **do**
    $(p_{N_{s,1}}, p_{N_{s,2}}, p_t)_i \leftarrow (r(1,120), r(1,120), 0)$
    $(\text{cost}, p_t)_i \leftarrow c(p_{N_1}, p_{N_2})$
    $(v_{N_{s,1}}, v_{N_{s,2}})_i \leftarrow \left(r(-\frac{119}{2}, \frac{119}{2}), r(-\frac{119}{2}, \frac{119}{2})\right)$
**end for**
```

At each iteration of the PSO algorithm, we update the inertia w, velocities, particles, cost_i, $p_{\text{best},i}$ and g_{best} using equations $(3.13) - (3.17)$ from the generic PSO formulas above.

The *2FT>* and *2FT<* methods take four real parameters, $N_{s,1}$, $N_{s,2}$, t_1, and t_2. The cost function $c(N_{s,1}, N_{s,2}, t_1, t_2)$ for these methods returns 1 if $N_{s,1} \notin [1,120]$ or $N_{s,2} \notin [1,120]$.

Otherwise, the cost function returns the best cost as defined below. The cost function labels each fault/generator combination as CONTROL if

$$J_{60/\lfloor N_{s,1}\rfloor}[N] > t_1 \wedge J_{60/\lfloor N_{s,2}\rfloor}[N] > t_2 \,\text{for 2FT>} \tag{3.18}$$

or

$$J_{60/\lfloor N_{s,1}\rfloor}[N] < t_1 \wedge J_{60/\lfloor N_{s,2}\rfloor}[N] < t_2 \,\text{for 2FT<} \tag{3.19}$$

at every time slice N from 1 to 718, and NOCONTROL if the statement is never true for any N from 1 to 718. We then add up the cost of each incorrectly labeled fault/generator combination to determine the cost of the set of parameters, and then return that cost. Since we are trying to find two different thresholds, we cannot apply Lemma 3.5.2.

We use PSO with 10 particles and 500 iterations to find a well-adapted $N_{s,1}, N_{s,2}, t_1, t_2$ combination that minimizes the cost function. We choose these values because testing shows that the PSO converges to similar costs over different runs using these values.

The particles and velocities are initialized as

```
**for all** particles **do**
    $(p_{N_{s,1}}, p_{N_{s,2}}, p_{t_1}, p_{t_2})_i \leftarrow (r(1,120), r(1,120), r(0,1), r(0,1))$
    $\text{cost}_i \leftarrow c\left(p_{N_{s,1}}, p_{N_{s,2}}, p_{t_1}, p_{t_2}\right)_i$
    $(v_{N_{s,1}}, v_{N_{s,2}}, v_{t_1}, v_{t_2})_i \leftarrow \left(r(-\frac{119}{2}, \frac{119}{2}), r(-\frac{119}{2}, \frac{119}{2}), r(-\frac{1}{2}, \frac{1}{2}), r(-\frac{1}{2}, \frac{1}{2})\right)$
**end for**
```

The maximum velocity for $N_{s,1}$ and $N_{s,2}$ are again set to 119 and the maximum velocity for t_1 and t_2 is set to 1, the difference between the maximum and minimum thresholds.

Just as with the 2FT+ and 2FT- methods, we again update the inertia w, velocities, particles, $\text{cost}_i, p_{\text{best},i}$ and g_{best} at each iteration using the generic PSO formulas above.

With these definitions, we know that the best 2FT+, 2FT>, and 2FT< methods will have score no greater than the best FT method. If $(\bar{N_s}, \bar{t})$ are the parameters for the best FT method, then $(\bar{N_s}, \bar{N_s}, \bar{t}, \bar{t})$ are parameters for a 2FT+ method with the same cost, $(\bar{N_s}, X, \bar{t}, 0)$ are parameters with the same cost for any $1 \leq X < 121$, and $(\bar{N_s}, X, \bar{t}, 1)$ are parameters for a 2FT< method with the same cost for any $1 \leq X < 121$.

4. RESULTS - PHASE I

4.1 Convergence of 2FT> and 2FT< methods

Since the PSO algorithm is neither complete nor optimal, we do not know if the PSO runs will converge at all, and if they do converge, we also do not know if the parameters found are optimal. We carried out ten runs of the 2FT> and 2FT< methods to determine the convergence of each run and what the runs converge to.

We use PSO to find a well-adapted set of parameters that minimizes the cost of labeling fault/generator combinations using equation 3.18. Since neither 2FT< nor 2FT> methods can use either Lemma 3.5.1 or 3.5.2, evaluating the cost of each particle takes significantly longer for 2FT> and 2FT< methods compared to 2FT+ and 2FT- methods.

We now examine cost and particle convergence for a typical 2FT> run. The worst and best cost behavior of a typical 2FT> run is shown in Figure 4.1.

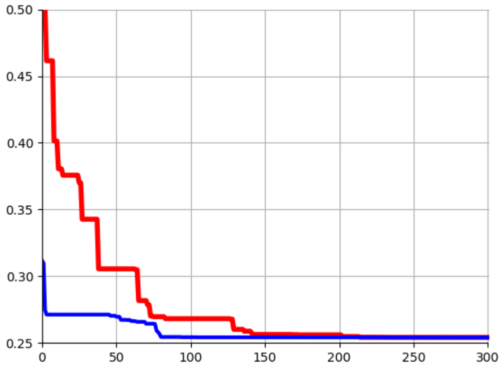

Fig. 4.1.: 2FT> Clustered Noiseless Convergence of worst cost and best cost

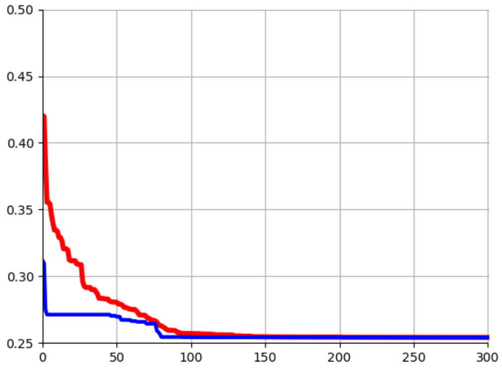

Fig. 4.2.: 2FT> Clustered Noiseless Convergence of average cost and best cost

The red line is the worst cost at each iteration, and the blue line is the best cost at each iteration. We see that the worst-cost particle takes until after iteration 200 to converge to the best-cost particle, even though we found the best-cost particle around iteration 75.

We also see the convergence of average cost and best cost of the 2FT> run in Figure 4.2. The red line is the average cost at each iteration and the best cost is the blue line at each iteration. The cost of the average particle was near to the cost of the best particle around iteration 130. This implies that we can implement an early cutoff of the PSO if we want to exit the PSO early when the average cost becomes very near to the best cost.

In each of Figures 4.4, 4.5, 4.6, 4.7, 4.9, 4.11, 4.10, and 4.12, the green dots represent the parameter value at the given iteration, the solid blue line is the value of the parameter with the best cost for that particle at that and all previous iterations, and the dashed red line is the value of the parameter with the best cost for any particle at all previous iterations.

In Figures 4.3 and 4.8, the green dots are the cost at the given iteration, the solid blue line is the best cost for that particle at that and all previous iterations, and the dashed red line is the best cost for any particle at that and all previous iterations.

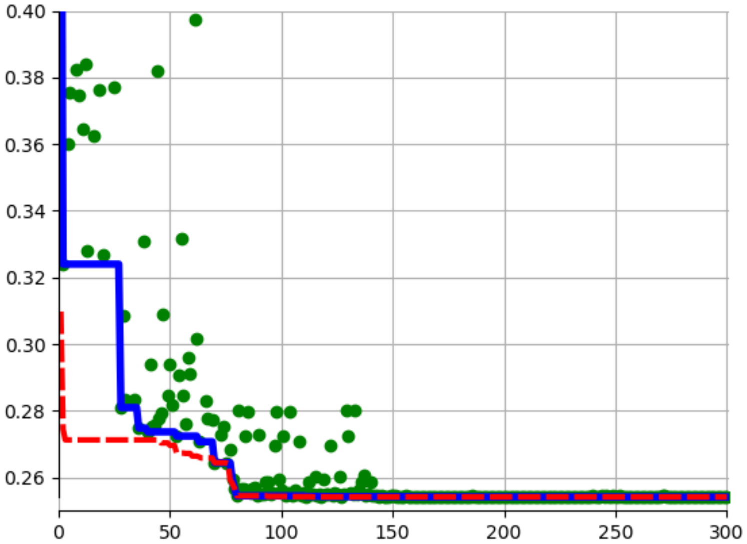

Fig. 4.3.: 2FT> Noiseless Convergence of cost for particle 2

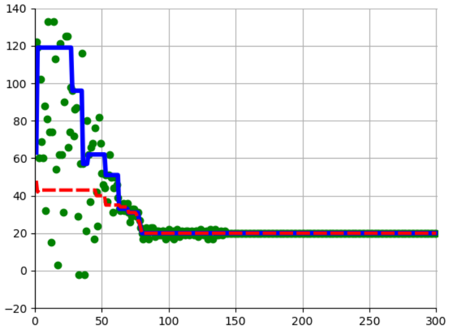

Fig. 4.4.: 2FT> Noiseless Convergence of $N_{s,1}$ for particle 2

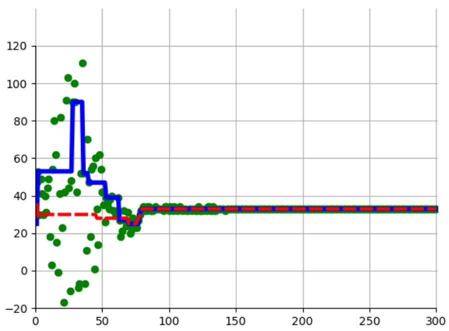

Fig. 4.5.: 2FT> Noiseless Convergence of $N_{s,2}$ for particle 2

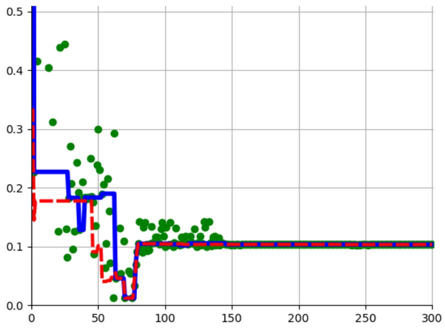

Fig. 4.6.: 2FT> Noiseless Convergence of T_1 for particle 2

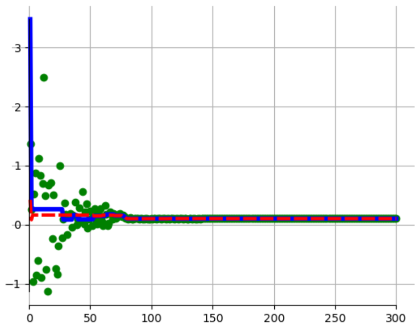

Fig. 4.7.: 2FT> Noiseless Convergence of T_2 for particle 2

We see in Figures 4.4 - 4.7 that when the best value for a parameter for a particle is not identical to the global best value of the same parameter, then the parameter values will oscillate between and near the best value of the particle and the global best value. The magnitude of the oscillation is approximately proportional to the height difference between the best value of the particle and the global best value.

Figures 4.8 - 4.9 detail the parameter values and cost for a typical 2FT< particle and appear to exhibit similar behavior except for the range of iterations from about 118 to 145.

The sudden change in the parameters of the 2FT< shown in Figures 4.8 - 4.9 from iteration 118 to iteration 145 arises from several small changes to the best value of the cost function over this region of iterations. Over this range of iterations, particle 2 is no longer close to the g_{best} particle. As the g_{best} particle changes, all of the particles, including particle 2 that we examine here, use a different g_{best} when calculating the velocity. This leads to greater variation until particle 2 once again becomes the best particle.

The value of w decreases from 0.9 at the first iteration to 0.5 at the last iteration. As the iteration count progresses, the velocities tend to settle to zero faster near the end of the runs compared to the beginning of the runs.

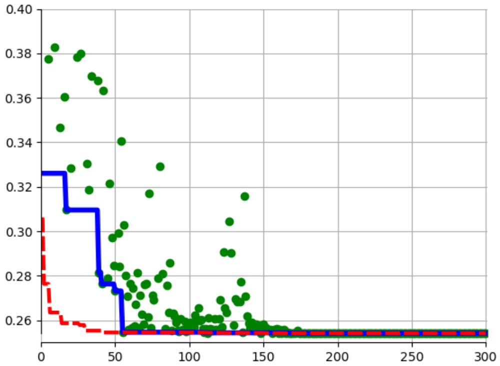

Fig. 4.8.: 2FT< Noiseless Convergence of cost for particle 0

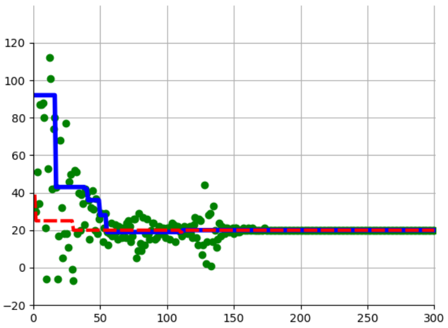

Fig. 4.9.: 2FT< Noiseless Convergence of $N_{s,1}$ for particle 0

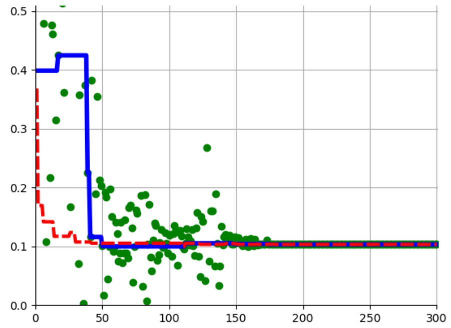

Fig. 4.10.: 2FT< Noiseless Convergence of T_1 for particle 0

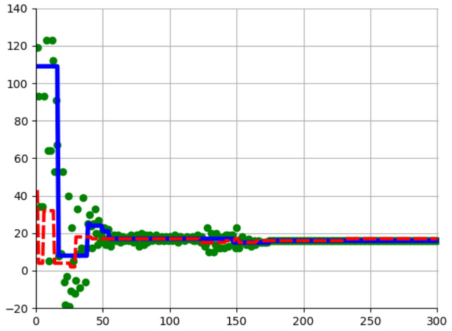

Fig. 4.11.: 2FT< Noiseless Convergence of $N_{s,2}$ for particle 0

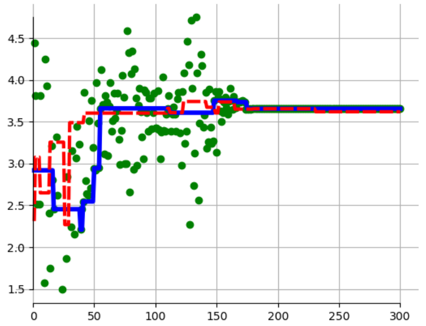

Fig. 4.12.: 2FT< Noiseless Convergence of T_2 for particle 0

4.2 Clustered Data

We consider methods that use Mei's clusters with both noiseless and noisy data.

4.2.1 Noiseless

Table 4.1.: Classes, Noiseless well-adapted parameters and costs

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	24		-0.1624380		0.2889960
FT	19		0.1076435		0.2646988
2FT+	20	120	0.1660495		0.2489950
2FT-	24	23	0.0075084		0.2518389
2FT>	**6**	**83**	**0.03582424**	**0.0281130**	**0.2439664**
2FT<	20	35	0.1085003	2.3084653	0.2555781

The best cost for any N_s using the *df/dt* method was 0.28900 at $N_s = 27$. The corresponding threshold at $N_s = 24$ was -0.162438. Figure 4.13 shows the distribution of costs for each N_s from 1 to 120, while figure 4.14 shows the distribution of thresholds over the same domain. For comparison, the simple backward difference method using $N_s = 1$ had a best cost of 0.306863 at threshold -0.017420.

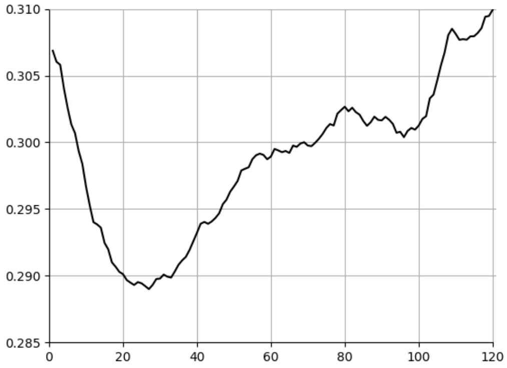

Fig. 4.13.: *df/dt* best cost for each *N*$_s$

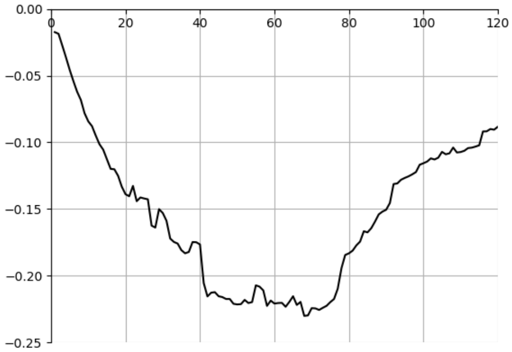

Fig. 4.14.: *df/dt* best threshold for each *N*$_s$

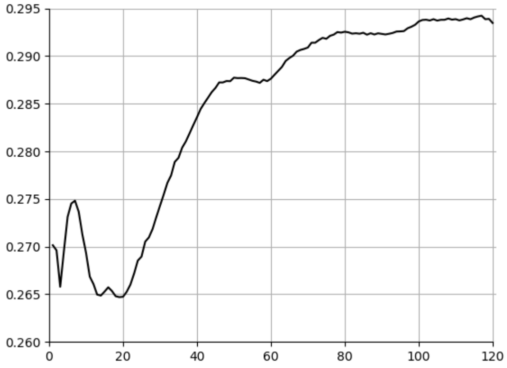

Fig. 4.15.: FT best cost for each *N*$_s$

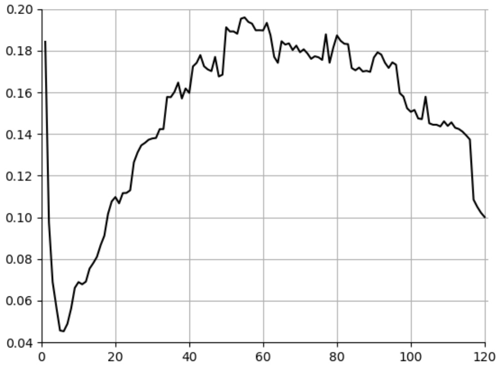

Fig. 4.16.: FT best threshold for each *N*$_s$

The best FT method correctly identified more unstable and more stable fault/generator combinations than the best *df/dt* method. The best cost for any N_s using the FT methods was 0.2646988 at $N_s = 19$. The corresponding threshold at $N_s = 19$ was 0.1076435. Figure 4.15 shows the distribution of costs for each N_s from 1 to 120, while figure 4.16 shows the distribution of thresholds over the same domain.

The 2FT+ methods have three parameters $N_{s,1}, N_{s,2}$, and t. The N_s parameters correspond to the lengths of the Fourier transforms as in the FT methods and the t parameter corresponds to the threshold in the FT methods. The best 2FT+ method had a cost of 0.2497549 at $N_{s,1} = 20, N_{s,2} = 120, t = 0.10883$. This is better than any method using only one N_s.

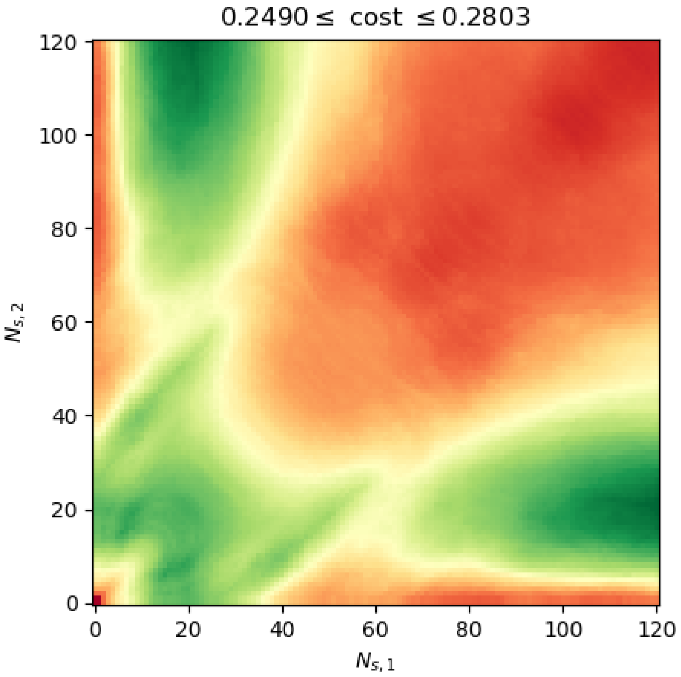

Fig. 4.17.: Heat Map for Clustered Noiseless 2FT+ methods

We see the heat map for the cost of 2FT+ methods compared to $N_{s,1}$ and $N_{s,2}$ in Figure 4.17. Green indicates a lower cost and red indicates a higher cost. We see that when both parameters are large or when one parameter is very low, the cost is high. The heat map also shows that since the defining equation for 2FT+ methods is symmetric in $N_{s,1}$ and $N_{s,2}$, then the heat map is also symmetric in $N_{s,1}$ and $N_{s,2}$.

The 2FT- methods used the same parameters as the 2FT+ methods. The best 2FT-method had a cost of 0.2556892 for the parameters $N_{s,1} = 20, N_{s,2} = 1, t = 0.0019373$.

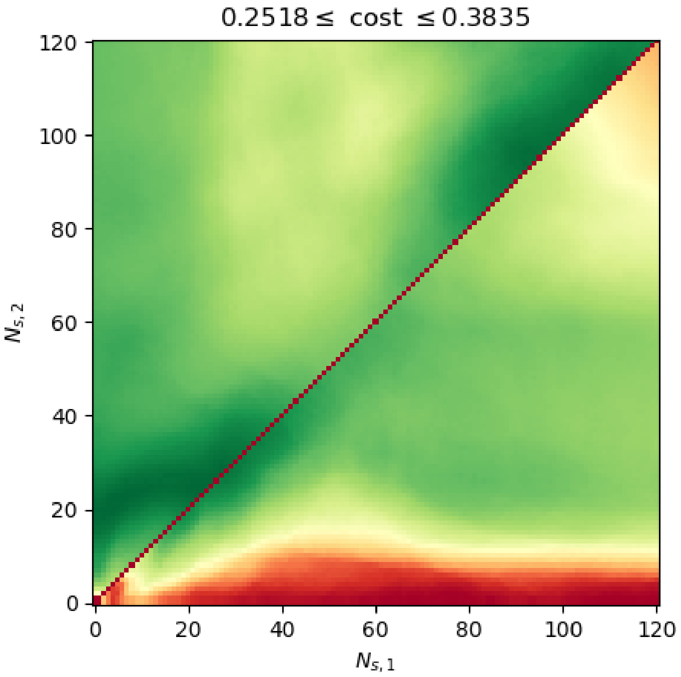

Fig. 4.18.: Heat Map for Clustered Noiseless 2FT- methods

The heat map for the 2FT- methods is shown in Figure 4.18. As with the 2FT+ heat map, we see that the green areas indicate a low cost, and the red areas indicate a high cost. Of note, the line where $N_{s,1} = N_{s,2}$ has an exceptionally high cost. This is to be expected, as along that line we see that

$$J_{\frac{60}{N_{s,1}}}[N] - J_{\frac{60}{N_{s,2}}}[N] = 0.$$

Therefore, when $N_{s,1} = N_{s,2}$, we have a lazy method for any threshold as defined above. We also see that when $N_{s,2}$ is near or less than 12, we have high costs for all 2FT- methods, and that these methods are not symmetrically distributed.

Our best local, noiseless 2FT< method has $N_{s,1} = 21, N_{s,2} = 35, T_1 = 0.10850, T_2 = 2.30847$ and a cost of 0.25558. Finally, our best local, noiseless 2FT> method has $N_{s,1} = 6, N_{s,2} = 83, T_1 = 0.03582, T_2 = 0.02811$ and a cost of 0.24396.

We summarize the best noiseless values that we found for each class of methods in table 4.1.

4.2.2 Noisy

From Dahal [17], we obtained angle data obtained from measuring a nominal 120V, 60Hz wall outlet at 60 samples per second for a 24-hour period on one day in each of April, May, July, August, September, and October 2018. The actually measurement was on voltage data measured at 720 samples per second and transformed to angle data at 60 samples per second by Dahal.

Table 4.2.: Standard Deviations of Measured Angle Data for Samples Sets of Length 360

Month	Data Points	Max. σ	Med. σ
Apr.	5168640	.5688	.1311
May	5496344	.5905	.1299
Jul.	5313960	.5627	.1601
Aug.	5209080	.4717	.1156
Sep.	5095560	.5279	.1181
Oct.	5077560	.5613	.1246

Table 4.2 shows the sample size, maximum standard deviation, and median standard deviation in each daily data set. Since the noise from the actual data suggests that the angle data has standard deviation of less than 10^0 and the median noise has standard deviation greater than 10^{-1}, we estimate our noise as white Gaussian noise with two different data sets. One data set has mean 0 and a standard deviation of 10^0, while the other has mean 0 and a standard deviation of 10^{-1}. This provides both a high and low estimate of the noise of a standard non-faulted real-world signal.

Table 4.3.: Classes, well-adapted parameters and costs with $\sigma = 10^0$

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	24		-0.147511		0.289462
FT	21		0.1085088		0.2555781
2FT+	**20**	**120**	**0.1697239**		**0.2497549**
2FT-	20	1	0.1042292		0.2556892
2FT>	102	19	0.0209582	0.0212494	0.2535303
2FT<	21	1	0.0253148	0.7885971	0.2555781

Table 4.4.: Classes, well-adapted parameters and costs with $\sigma = 10^{-1}$

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	27		-0.161878		0.289016
FT	20		0.1035182		0.2540916
2FT+	21	120	0.1701634		0.2489967
2FT-	27	26	0.0072455		0.2522862
2FT>	**10**	**44**	**0.0100958**	**0.0195636**	**0.2462258**
2FT<	20	47	0.0255056	0.6247670	0.2540916

We now compare results between Tables 4.1, 4.3, and 4.4. We see that the best method with $\sigma = 10^0$ noise added has a cost of 0.2497549, which is 0.31% worse than the best method with no noise added. More importantly, when we apply the best 2FT+ noiseless method to the data with $\sigma = 10^0$ noise added, we find that the cost is 0.2511758, and when we do the same thing with the data with $\sigma = 10^{-1}$ noise added, the cost is 0.2500713. Therefore, the 2FT+ method with parameters ($N_{s,1} = 20$, $N_{s,2} = 120, t = 0.1660495$) is a superior local method to all other local methods examined here when considered using data similar to that from the real world.

4.3 Non-Clustered Data

4.3.1 Noiseless

We now consider the effect of ignoring the local data from [3] and only train N_s and thresholds with instability and not locality. This significantly improves our ability to predict an unstable fault using only data from a single generator. We see in Table 4.5 the best method is 2FT> with $N_{s,1} = 119, N_{s,2} = 2, T_1 = 0.0884221, T_2 = 0.0118327$. This constitutes a large reduction in cost by not using local data.

Table 4.5.: Classes, Non-Clustered noiseless well-adapted parameters and costs

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	11		-0.130921		0.225603
FT	20		0.1280526		0.1864107
2FT+	18	2	0.1213684		0.1850943
2FT-	10	120	0.1658127		0.1772580
2FT>	**119**	**2**	**0.0884221**	**0.0118327**	**0.1758442**
2FT<	20	79	0.1280535	2.51650418	0.186410

4.3.2 Noisy

We also calculated best methods using the same noisy data used with local methods. We again found for both high-noise and low-noise data that the best method was from the 2FT> class. The results are summarized in Table 4.6 and Table 4.7. We also found that using the best noiseless, non-local method on the noisy data yielded only a negligible increase in cost compared to the noiseless cost.

Table 4.6.: Classes, Non-local well-adapted parameters and costs with $\sigma = 10^0$

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	14		-0.157117		0.226741
FT	19		0.1367440		0.1865920
2FT+	17	118	0.2131507		0.1779083
2FT-	19	8	0.1068498		0.1866349
2FT>	**115**	**12**	**0.0902601**	**0.1006538**	**0.1776748**
2FT<	19	64	0.13674349	1.51716236	0.1865920

Table 4.7.: Classes, Non-local well-adapted parameters and costs with $\sigma = 10^{-1}$

Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
df/dt	11		-0.130779		0.225589
FT	18		0.1291923		0.1862841
2FT+	10	120	0.1667698		0.1774239
2FT-	20	7	0.1114471		0.1852960
2FT>	**83**	**5**	**0.0960018**	**0.0324950**	**0.1769399**
2FT<	18	74	0.1291856	3.15816419	0.1862841

5. METHODOLOGY - PHASE II

5.1 Introduction

We move from only training a decision algorithm to also evaluating the effectiveness of a similar algorithm from the *df/dt* class of methods. In phase II, we establish that many disconnect generator events can be modelled by shedding different amounts of load, and that we can control events that simulate loss of generation capacity while issuing fewer or no controls for other types of events. Furthermore, we would like to design an algorithm that sheds load efficiently across PMUs within the disconnected generator's cluster and is robust in a noisy environment.

5.2 Labels

In this phase we expand our test bed to include events with two faults as well as those which simulate disconnecting generation capacity or disconnecting load. We label the 480 one-phase faults from the original test bed as type 1 and the 480 three-phase faults from the original test bed as type 3. We construct the set of type 13 events by selecting a one-phase fault followed by a three-phase fault, where the three-phase fault was found by selecting two pairs of faults from a sample of 15 from the 40 pairs of faults that comprise the one-phase faults above.

We choose each combination of two pairs of buses from this list. If the same pair of buses is selected, then we generate a one-phase fault on that pair of buses from cycle 36 to cycle 40. If different pairs of buses are selected, then we generate a one-phase fault on the first pair of buses from cycle 36 to cycle 40 followed by a three-phase fault on the other pair of buses two seconds later from cycle 156 to cycle 160. This gives us 225 type 13 faults. Of the 225 faults, 140 are stable before controls and 85 are unstable before controls.

We simulate the disconnection of generation capacity by adding positive admittance at a bus near a generator, while we simulate the disconnection of load requirements by adding negative admittance at a bus near a generator. We label disconnection of generator capacity events as type AP, while disconnection of load events as type AN. We created AN and AP events at buses near each generator with MW of admittance added or subtracted at $50, 55, \ldots, 850$ MW. This yields 4669 events of both type AN and type AP. We see in Section 5.4 that we can simulate disconnected generators by adding admittance.

We also define the effectiveness of a method acting on an event by dividing the change in RMSGA by the total MW shed during that event. The effectiveness of a method for class of events is the average effectiveness of all events of that class in the test bed. A high effectiveness value indicates that the method did a better job of inexpensively easing the stress on the network compared to a method with a low effectiveness value.

5.3 Simulation Stability

Figures 5.1, 5.2, 5.3, 5.4, and 5.5 depict examples of unstable type AP, 1, type 3, type 13, and type AN, respectively. A downward deflection of the generator speed indicates an incipient underfrequency condition. We note in particular that the 1, 3, and 13 events all exhibit sinusoidal behavior at some generators with either an initial upward or downward deflection. The unstable AN events have a consistent initial upward deflection, while the AP events have a consistent initial downward deflection.

We note that we desire to strictly control events such as Figure 5.1. On the other hand, we do not necessarily desire to control events such as Figures 5.2, 5.3, 5.4, and 5.5.

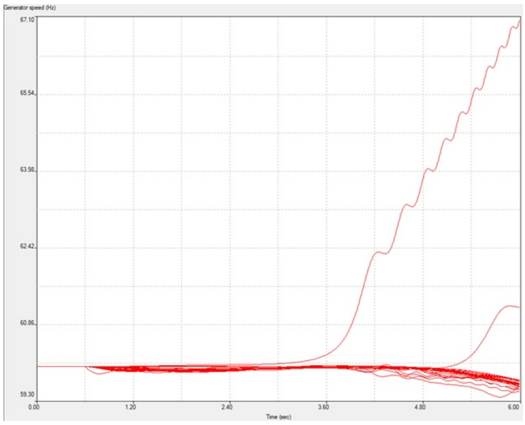

Fig. 5.1.: Type AP event at generator NAVAJO of 835 MW

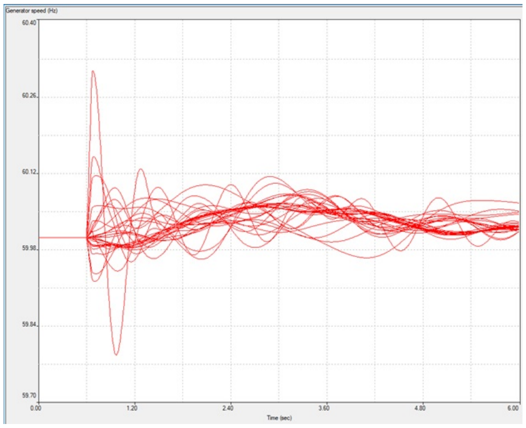

Fig. 5.2.: Type 1 event between buses MALIN1 and MALIN2

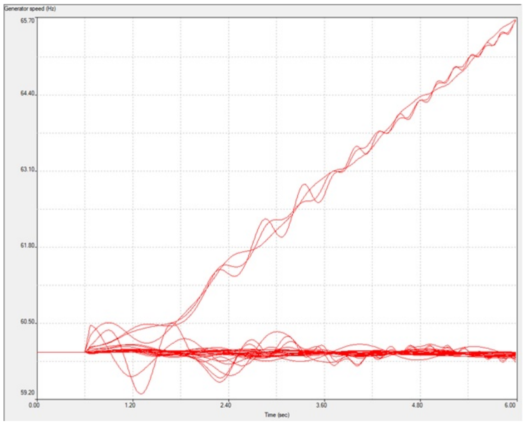

Fig. 5.3.: Type 3 event between buses FOURCORN and SAN JUAN

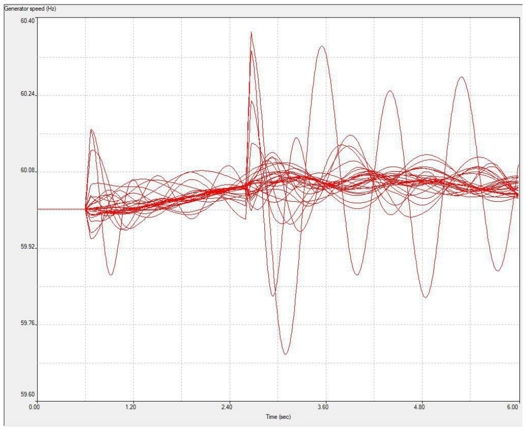

Fig. 5.4.: Type 13 event between GRIZZLY8 and GRIZZLY9 and between ROUNDMT and ROUND3

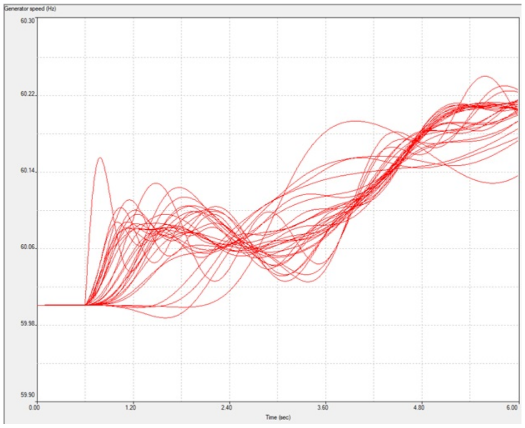

Fig. 5.5.: Type AN event at generator NAVAJO of -835 MW

5.4 Simulating Disconnected Generators

We wish to simulate disconnection of a generator under load by adding admittance at a bus near the generator. The simplified 176-bus WECC model amalgamates several physical generators at some generator locations, so adding admittance near a generator has a similar effect to disconnecting a portion of the generation capacity at the amalgamated generation location.

For each generator for which disconnecting the generator does not lead to instability across the entire network, we compared an accurate representation of disconnecting the generator with adding an equivalent amount of real and reactive power at a nearby bus. We determined the appropriate levels of real and reactive power directly from the loadflow of the 176-bus WECC model.

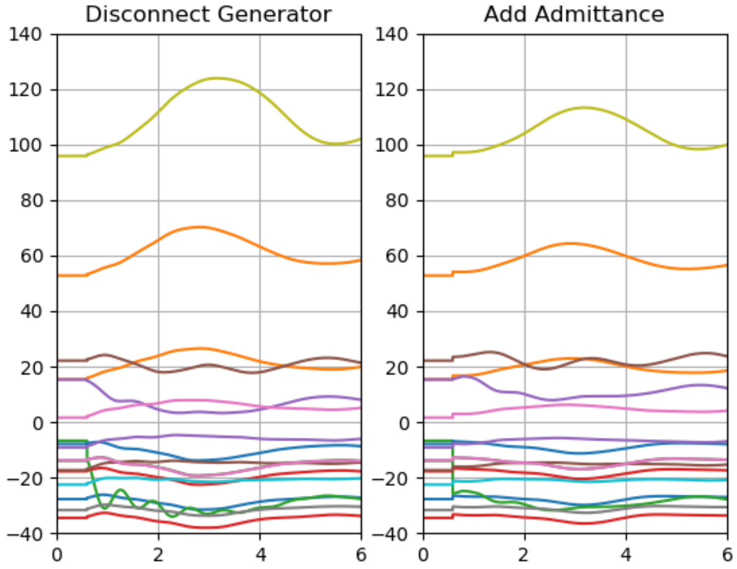

Fig. 5.6.: Comparison of RMSBA at Coronado Disconnect Event

As an example, the individual generator angles used to calculate the RMSBA obtained by disconnecting the Coronado generator in Apache County, AZ, and by adding the equivalent admittance from the loadflow at a bus near the Coronado generator are compared in

Figure 5.6. We can see both by inspection and by the RMSGA values that the simulations yielded similar results. The disconnected generator simulation has an RMSBA of 52.13, while the similar AP event has an RMSBA of 51.49.

5.5 Methods for Comparison

We wish to compare our best methods to several other methods. The first is the well-known point-to-point *df/dt* method [4]. This method is our *df/dt* method with the window-length fixed at 1 cycle and a well-adapted threshold.

We also examine the *ideal response* to compare our tested methods against the best possible response of the network. The ideal response is the set of controls where every PMU in the system triggers a control 6 cycles after a fault occurs anywhere in the system. Unlike the other methods developed in this thesis, the ideal response requires knowledge of the states of the generators and PMUs across the network. The ideal response quickly reduces load requirements across the network, so it should be near the maximum possible possible reduction in RMSGA for a given event.

We calculate the most well-adapted point-to-point *df/dt* method and the ideal response. We compare each of these methods to the methods developed in this thesis to determine if it is an improvement over the point-to-point *df/dt* method and how close it gets to the maximum possible RMSGA reduction.

6. IMPLEMENTATION - PHASE II

6.1 Pre-Processing

The phase II test bed includes not only the 480 one-phase fault events, 480 three-phase fault events, and 210 combined one-phase fault and three phase fault events, but also 4669 events that simulate disconnecting generation capacity and 4669 events that simulate disconnecting load. Therefore, we extended the phase I pre-processing from section 3.2 by running uncontrolled simulations of the 4669 generation capacity disconnection events and the 4669 load disconnection events and recording the bus angles and generator angles for each PMU and generator for six seconds at 120 samples per second.

Further, the pre-processing for phase II includes all of the pre-processing using in section 3.2 and extends it by pre-computing *df/dt* values for moving average lengths from 1 to 120, This is similar to the way that we pre-computed Fourier values for N_s lengths from 1 to 120.

6.2 Testing

We started with 1170 type 1, type 3, and type 13 events as well as 4669 type AP events and 4669 type AN events. For each class of methods, we trained the various parameters of the class using a PSO that gave equal weights to the total false positive and total false negatives as in section 3.5.3.

The training test bed for each PSO consisted of $\frac{2}{7}$ of the type 1, 3, and 13 events as well as $\frac{2}{5}$ of the type AP and AN events chosen systematically from each type of event. Once our training determined a most well-adapted set of parameters for each class using a PSO, we tested the method for ordering controls in the simulation.

Similarly, we chose the testing set to consist of $\frac{1}{7}$ of the type 1, 3, and 13 events and $\frac{1}{5}$ of the type AP and AN events. We simulated each event without controls and then allowed each method to order controls with a 100 ms delay between detecting the need for control and applying the control. We re-ran the simulation with the control until checked in a loop to see if more controls were needed until the simulation came to an end with no more controls needed.

We then wrote the results to an output file. The results for each event consisted of the following:

- Time of the simulation
- Name of the event
- Power injected for each control
- Initial RMSGA with no controls
- Initial RMSBA with no controls
- Final RMSGA with controls
- Final RMSBA with controls
- Initial stability of the generators
- Final stability of the generators
- Initial stability of the buses
- Final stability of the buses
- Number of controls issued
- Number of controls issued local to the fault

For each method, we tabulated the following for power injection levels ranging from 25 MW to 75 MW and collectively for the type 1, type 3, and type 13 events as well as for the type AN and type AP events:

- Efficiency in MW / unit reduction in RMSGA
- Number of controls issued
- Number of stabilized events
- Number of destabilized events

All methods were tested with noiseless inputs. In addition, the best methods were also examined in an noisy environment with Gaussian noise added to the bus and generator angles with mean 0 and standard deviation 10^0.

6.3 Costs

As in section 3.4, we again assigned a positive cost to mislabeled events. We collate the events that should be controlled and the events that should not be controlled separately to find that we should control 1938 unstable, local events, and we should not control the remaining 17119 events. Therefore, the cost of controlling an event that should not be controlled is $\frac{1}{2\times 17119}$, and the cost of not controlling an event that should be controlled is $\frac{1}{2\times 1938}$.

6.4 Classes of Well-Adapted Methods

In phase I, we used two fundamentally different algorithms to classify methods. The *df/dt* methods used $J_{\Delta f}$ while the FT and 2FT methods used $J_{60/\lfloor N_s \rfloor}$. We now consider classes of methods that combine these two algorithms to try to control only AP events but not type 1, 3, 13, or AN events.

We try various types of combinations. Since we only want to control events with a negative deflection at time of the fault, we use *df/dt* to find the start of a window. Within that window, we then search for local oscillations that match certain criteria similarly to what we do for the 2FT methods in Section 3.5.1.

We wish to determine a well-adapted set of parameters and amount of power to shed at a PMU, ranging from -25 to -75 MW of admittance added at a bus with a PMU. This represents the load shed by temporarily disabling consumer appliances that act as a load connected to that PMU. We wish to select well-adapted bus methods such that the method does not destabilize any previously stable events, and the method otherwise improves the efficiency of the control as much as possible .

6.4.1 *df/dt*

When we apply our best fit *df/dt* method to a test bed with a mix of one-phase faults, three-phase faults, generator disconnections, and load disconnections. We use a *df/dt* method as described in Section 3.5.1.

6.4.2 *df/dt* and 2FT>

We started by trying to find a suitable *df/dt* method and window to combine with a 2FT> method, the best class of method from phase I. This used a seven-parameter PSO. The seven parameters were N_s and T_N for *df/dt*, $N_{s,1}, N_{s,2}, T_1$ and T_2 for 2FT>, and another parameter for the window length W in cycles.

We determined if the *df/dt* moving average of N_s with threshold T_N occurred. For the next W cycles after the *df/dt* method crossed the threshold, we used the 2FT> method with parameters $N_{s,1}, N_{s,2}, T_1$ and T_2. If both of the 2FT> thresholds were exceeded for their respective lengths, then we issued a control. We trained the algorithm to only issue control for AP events where the simulated generator disconnection is in the same cluster as the PMU.

When we tested this against a subset of the 1, 3, 13, AP, and AN events, the algorithm only controlled only 1, 3, 13, and AN events, but not AP events. Therefore, this approach was quickly abandoned.

6.4.3 *df/dt* and 2FT<

This class of methods is similar to the *df/dt* and 2FT> class of methods, substituting 2FT< for 2FT>. Again, when we tested against the same subset of type 1, type 3, type 13, type AP, and type AN events, we controlled only the type 1, type 3, type 13, and type AN events, but not the type AP events, so this approach was also abandoned.

6.4.4 *df/dt* and Inverted 2FT>

This class of methods is similar to the *df/dt* and 2FT< class of methods above, but instead of find a value above one threshold and below another, we tried to find a value below each threshold. Correspondingly, the cost function labels each event CONTROL if and only if $J_{\Delta f,N_s}[N] < T$ and

$$J_{60/\lfloor N_{s,1} \rfloor}[M] < T_1, J_{60/\lfloor N_{s,2} \rfloor}[M] < T_2$$

for some $N \leq M \leq N+W$.

As we can see in section 5.3, we were unable to control AP events without controlling all events, since the AP events can only be controlled if we are trying to detect exceptionally small local oscillations. Upon this realization, we inverted the 2FT> test so that we controlled only when we had small values of $J_{60/\lfloor N_{s,1} \rfloor}$ and $J_{60/\lfloor N_{s,2} \rfloor}$.

6.4.5 *df/dt* with Blocking

We consider another method for using *df/dt* methods to control AP events. We see that all type AP events have a downward deflection in generator speed at the beginning of the event; all type AN events have an upward deflection in generator speed at the beginning of the event; and the type 1, type 3, and type 13 events have a mix of initial deflections. We want to control type AP events, while avoiding controlling type AN events. Finally, if we control any type 1, 3, or 13 events, then we would like to do so in an efficient manner. To distinguish type AP events from other events, we added another index $J_{\Delta f,T_P}$ which requires a control when

$$J_{\Delta f,N_s}[N] < T_N \wedge \forall t \in [0,N-1], J_{\Delta f,t} \leq T_P$$

is true.

In this manner, if we observe an upward deflection of a moving average of length N_s of generator speed at a generator that exceeds a threshold $T_P > 0$, then we stop trying to control the generator for a set length of time. For this thesis, we assume that the set length is longer than the length of the simulation, so we stop trying to control each bus near its generator by shedding load at the bus. However, in practice, it is necessary to allow a control after some small number of seconds after an upward deflection that exceeds the T_P threshold.

Otherwise, this class of methods distinguishes between stable and unstable events just as the *dt/dt* methods from phase I did. That means that if we see a downward deflection of the same moving average as above that falls below a threshold $T_N < 0$, then we apply a control at the bus near that generator. If neither threshold is exceeded, then we do not apply a control for that generator.

6.4.6 *df/dt* with blocking and 2FTx methods

The previous several methods used *df/dt* with another method that compared *df/dt* with two Fourier indices. We combined *df/dt* with blocking and the same Fourier-based methods. This controls only generators which trigger the $J_{\Delta f, T_P}$ index as well as some conditions on the Fourier indices.

6.4.7 Additional Methods

The point-to-point *df/dt* method has a fixed $N_s = 1$, so we only need to find a well-adapted threshold T. We could have done this using bisection, but we determined the randomness of a PSO would help avoid local cost minima. The best threshold for our training data is -0.0007380.

The ideal response methods have no parameters. We simply activate all of the controls 100 ms after the start of the event.

7. RESULTS - PHASE II

We tested the methods by shedding different amounts of load at each power injection site. The corresponding power change amounts are -25 MW, -50 MW, and -75 MW. We combined all of the type 1, 3, and 13 into the transient type. We do not care if we control this type of event, but we do want any such control to be efficient in reducing RMSGA.

For each power level and event type, we report:

- Average number of controls issued (Avg. Controls)
- Average change in RMSGA (ΔRMSGA)
- Effectiveness of the controlled expressed as the average change in RMSGA divided by the GWs of controls issued (Effectiveness)
- Percentage of unstable events that were stabilized (% Stab) OR
- Percentage of stable events that were destabilized (% Destab)

As shown in Sections 7.1 amd 7.2, we noted that the *df/dt* method and *df/dt* with 2FT methods either controlled all the PMUs, similar to the ideal non-clustered response, or they controlled none of the PMUs. As neither of these were the desired behavior, we added a blocking component as described above to both the *df/dt* methods and the *df/dt* with 2FT methods. These methods generally produced superior results.

We present sub-optimal results for the *df/dt* class in Section 7.1 and for *df/dt* and 2FTx classes in Section 7.2. The successful results using blocking *df/dt* are in Section 7.3 and 7.4. We show results for other comparison classes in Sections 7.5 and 7.6. Finally, we show that our best methods are robust in a noisy environment in Section 7.7.

7.1 *df/dt* Class

The best-adapted method for a *df/dt* method had parameters

- $N_s = 55$
- $T = -0.0138$

where N_s was the length of the moving average window and T was the threshold for the rate of change of frequency.

Tables 7.1 and 7.2 show that the best *df/dt* method on average issued more than 16 out of a possible 17 controls for unstable events event. While this method did an admirable job of efficiently stabilizing AP events, it does not distinguish between AP events and other events effectively, so the *df/dt* class is unsuitable for our purposes.

Table 7.1.: Results for *df/dt*: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	16.2	-29.91	73.85	0
-25	AN	16.9	-2.42	5.73	0
-25	AP	16.9	-44.78	105.98	51.3
-50	Transient	16.2	-55.22	68.26	0
-50	AN	16.9	-4.80	5.69	0
-50	AP	17.0	-60.78	71.47	78.6
-75	Transient	16.2	-71.88	59.24	9.1
-75	AN	16.8	-6.57	5.19	0
-75	AP	17.0	-67.83	53.19	78.6

7.2 *df/dt* and 2FTx Classes

All of the best *df/dt* and 2FTx methods had challenges of one sort or another. They are presented together with an analysis of their faults.

Table 7.2.: Results for *df/dt*: stable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Destab
-25	Transient	17.0	-0.17	0.40	0
-25	AN	17.0	0	-	0
-25	AP	17.0	-1.09	2.56	0
-50	Transient	17.0	-0.34	0.40	0
-50	AN	17.0	0	-	0
-50	AP	17.0	-1.81	2.13	0
-75	Transient	17.0	-0.50	0.39	0
-75	AN	17.0	0	-	0
-75	AP	17.0	-2.32	1.82	0

7.2.1 *df/dt* and 2FT> Class

The results for the most well-adapted *df/dt* and 2FT> method acting on unstable events are summarized in Table 7.3. The parameters for this method are:

- $N_{s_1} = 27$
- $T_1 = 0.06524$
- $N_s = 22$
- $W = 55.$
- $N_{s,2} = 2$
- $T_2 = 0.12874$
- $T = 0.01371$

The most well-adapted *df/dt* and 2FT> method was in large part the opposite of the most well-adapted *df/dt* method. Where the *df/dt* method issued controls far too often, the most well-adapted *df/dt* and 2FT> method issued controls far too seldom. The best *df/dt* method issued more than 16 out of 17 controls on average for both transient and AP events at the -25 MW level, while the best *df/dt* and 2FT> method issued less than 3 controls on average for transient events and less than 1 control on average for AP events at the -25 MW level.

The best attributes of this method were that it never applied any control to any AN event and that it never applied any controls to any stable event. Table 7.3 shows the results for

Table 7.3.: Results for *df/dt* and 2FT>: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	2.8	-0.33	4.69	0
-25	AN	0.0	0.00	-	0
-25	AP	0.2	-0.05	13.08	0
-50	Transient	2.0	-0.17	1.66	0
-50	AN	0.0	0.00	-	0
-50	AP	0.1	-0.06	7.94	0
-75	Transient	2.4	-1.20	6.74	0
-75	AN	0.0	0.00	-	0
-75	AP	0.2	-0.07	6.22	0

unstable events. No table was needed for stable events, as no stable events had any controls issued.

This method was unsuitable even with noiseless simulations. Therefore, we did not perform simulations with noisy simulations of any kind.

7.2.2 *df/dt* and 2FT< Class

Tables 7.4 and 7.5 show the results of testing the most well-adapted *df/dt* and 2FT< method using noiseless data. The parameters for this method are:

- $N_{s_1} = 56$
- $T_1 = 0.00544$
- $N_s = 45$
- $W = 61.$
- $N_{s,2} = 45$
- $T_2 = 0.02879$
- $T = 0.02404$

The most well-adapted *df/dt* and 2FT< method did an adequate job of distinguishing unstable type AP and type AN events from stable type AP and type AN events. It was also reasonably efficient when controlling unstable transient events and type AP events. It stabilized a high proportion of unstable events while not destabilizing any stable events.

Table 7.4.: Results for *df/dt* and 2FT<: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	13.1	-43.54	132.98	0
-25	AN	14.5	-5.55	15.31	0
-25	AP	10.4	-50.66	195.69	64.2
-50	Transient	13.0	-75.09	115.47	9.1
-50	AN	14.5	-6.98	9.63	0
-50	AP	10.6	-52.01	98.43	68.6
-75	Transient	12.3	-74.70	81.17	0
-75	AN	14.0	-7.50	7.14	0
-75	AP	10.7	-53.57	67.11	71.6

Table 7.5.: Results for *df/dt* and 2FT<: stable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Destab
-25	Transient	11.1	-1.58	5.66	0
-25	AN	1.2	-0.12	4.08	0
-25	AP	3.2	-0.51	6.47	0
-50	Transient	10.4	-2.71	5.24	0
-50	AN	1.2	-0.23	3.82	0
-50	AP	3.0	-0.84	5.68	0
-75	Transient	9.3	-3.55	5.10	0
-75	AN	1.1	-0.27	3.11	0
-75	AP	2.9	-1.09	5.01	0

However, this method issued far more controls to transient events than was desirable. We note that this method is inefficient at responding selectively to events that occur in the same cluster as the power injection location. As noise tends to increase the number of opportunities to control, we do not test the method on the noisy data.

7.2.3 *df/dt* and Inverted 2FT> Class

Tables 7.6 and 7.7 show the relevant data for the most well-adapted *df/dt* and inverted 2FT> method. The best values for the parameters of this method are:

- $N_{s_1} = 57$
- $T_1 = 0.17563$
- $N_s = 20$
- $W = 32$
- $N_{s,2} = 49$
- $T_2 = 0.01289$
- $T = -0.01001$

Table 7.6.: Results for *df/dt* and inverted 2FT>: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	15.5	-23.85	61.73	0
-25	AN	12.6	-6.64	21.03	0
-25	AP	10.9	-6.93	25.35	7.1
-50	Transient	15.3	-36.67	48.01	0
-50	AN	12.5	-10.85	17.36	0
-50	AP	10.8	-15.74	29.18	6.9
-75	Transient	14.6	-58.44	53.25	0
-75	AN	10.4	-9.01	11.58	0
-75	AP	8.9	-32.40	48.38	13.1

The most well-adapted *df/dt* and inverted 2FT> method issued controls only when there was a downward deflection and also the Fourier coefficient at two particular frequencies was small. Since nearly all Fourier coefficients are small most of the time, this method was similar to a *df/dt* method with the same N_s and T in that it could not distinguish between type AP and other events.

Further, since transient events have some large sinusoidal element, they were detected by the *df/dt* component more often than either type AN or type AP events. This is indicated by the larger number of average controls issued for transient events compared to both type AN and type AP events for both stable and unstable events at all power levels.

Table 7.7.: Results for *df/dt* and inverted 2FT>: stable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Destab
-25	Transient	7.3	-1.01	5.54	0
-25	AN	0.5	-0.04	3.09	0
-25	AP	0.5	-0.07	6.18	0
-50	Transient	7.4	6.78	-17.3	7.0
-50	AN	0.5	-0.07	2.89	0
-50	AP	0.4	-0.12	5.41	0
-75	Transient	7.1	17.25	-32.20	6.9
-75	AN	0.5	-0.10	2.79	0
-75	AP	0.4	-0.17	5.00	0

This method was unsuitable with noiseless simulations. Therefore, we did not test simulations with noise.

7.3 Blocking *df/dt* Class

The most well-adapted *df/dt* with blocking method was suitable in all respects. Tables 7.8 and 7.9 show that the method was able to clearly distinguish between transient, AN, and AP events. The parameters for the most well-adapted blocking *df/dt* algorithm are:

- $N_s = 65$
- $T_P = 0.02303$
- $T_N = -0.0188$

This method issued many more controls on average to type AP events compared to transient events, and many more controls on average to transient events compared to type AN events. Further, every power level stabilized at least 80% of all unstable type AP events.

As this was our best result so far, we continued using the blocking *df/dt* in conjunction with the 2FT methods. All of the blocking *df/dt* and 2FT classes had better methods than the best blocking *df/dt* method, so we did not analyze blocking *df/dt* in a noisy environment.

Table 7.8.: Results for *df/dt* class with blocking: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	9.5	-38.47	161.81	3.2
-25	AN	4.3	-6.97	64.68	0
-25	AP	16.2	-52.65	130.04	80.4
-50	Transient	9.9	-79.48	160.51	8.6
-50	AN	4.5	-9.43	41.91	0
-50	AP	16.1	-66.27	82.25	94.8
-75	Transient	9.7	-84.86	117.23	9.7
-75	AN	4.9	-10.69	28.74	0
-75	AP	16.1	-78.67	65.15	96.1

Table 7.9.: Results for *df/dt* class with blocking: stable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Destab
-25	Transient	5.4	-0.30	2.22	0
-25	AN	0.9	0	-	0
-25	AP	11.2	-0.34	1.21	0
-50	Transient	5.5	-0.56	2.04	0
-50	AN	0.8	0	-	0
-50	AP	11.2	-0.55	0.98	0
-75	Transient	6.1	-0.82	1.79	0
-75	AN	1.1	0	-	0
-75	AP	10.9	-0.69	0.84	0

7.4 Blocking *df/dt* and 2FTx Classes

Our success with blocking *df/dt* led us to explore further variants on this theme. We combined successful metrics from Phase I with the previously successful blocking *df/dt*, and call these methods the blocking *df/dt* and 2FT classes.

In Table 7.10 through Table 7.14, the columns labeled, "Controls issued", "Local Controls", "ΔRMSGA per event", and "Effectiveness" are all averages across the Power (MW) and event type.

7.4.1 Blocking *df/dt* and 2FT-

The parameters for the most well-adapted blocking *df/dt* and 2FT- class are:

- $N_{s,1} = 98$
- $T_1 = 1.8210 \times 10^{-3}$
- $T_N = -1.1012 \times 10^{-2}$
- $N_{s,2} = 87$
- $N_s = 95$
- $T_P = 1.3892 \times 10^{-5}$

The results for this method over unstable events are summarized in Table 7.10.

Table 7.10.: Results for blocking *df/dt* and 2FT- method: unstable events, noiseless

Power (MW)	Event Type	Controls Issued	Local Controls	ΔRMSGA per event	Effectiveness	% Stab
-25	Transient	2.04	0.43	-6.74	132.0	0
-25	AN	0.39	0.0	-0.39	39.35	0
-25	AP	11.43	3.27	-48.25	168.83	58.8
-50	Transient	2.04	0.43	-7.77	76.09	0
-50	AN	0.39	0.0	-0.45	23.08	0
-50	AP	11.24	3.27	-70.08	124.7	88.2
-75	Transient	2.02	0.43	-9.24	60.98	0
-75	AN	0.39	0.0	-0.52	17.59	0
-75	AP	11.07	3.27	-75.14	90.54	92.2

We see that this method did a good job of issuing many controls for type AP events, only a few for transient events, and less than a half of a control on average for type AN events. However, this method used an average of 3.27 controls in the same cluster as the event, which is fewer in-cluster controls than some methods use below. The method was also not as efficient as other methods that had an effectiveness of more than 200 for type

AP events. Therefore, we declined to analyze this class further, as other classes had more efficient representatives which use a higher proportion of in-cluster controls.

7.4.2 Blocking *df/dt* and 2FT<

The summary of the results for the most well-adapted blocking *df/dt* and 2FT< class are in Table 7.11. The parameters for this method are:

- $N_{s,1} = 10$
- $N_{s,2} = 3$
- $T_1 = 1.5576 \times 10^{-3}$
- $T_2 = 7.0362 \times 10^{-4}$
- $N_s = 94$
- $T_N = -4.5932 \times 10^{-3}$
- $T_P = 7.9473 \times 10^{-5}$

We see that this method issued the most controls for type AP events at the -25 MW level of all of the *df/dt* and 2FTx classes, as well as the most in-cluster controls. However, this method issued more controls on average for transient events than other methods issued for transient events, so we declined to research this method further.

Table 7.11.: Results for blocking *df/dt* and 2FT< method: unstable events, noiseless

Power (MW)	Event Type	Controls Issued	Local Controls	ΔRMSGA per event	Effectiveness	% Stab
-25	Transient	5.05	0.87	-35.25	279.01	2.2
-25	AN	0.83	0.0	-0.86	41.63	0
-25	AP	14.55	4.61	-74.79	205.63	92.8
-50	Transient	4.78	0.86	-62.49	261.18	7.5
-50	AN	0.82	0.0	-1.08	26.37	0
-50	AP	14.37	4.6	-81.26	113.07	95.4
-75	Transient	4.58	0.83	-99.05	288.31	7.5
-75	AN	0.80	0.0	-1.27	21.26	0
-75	AP	13.82	4.58	-86.22	83.16	97.4

7.4.3 Blocking *df/dt* and 2FT>

The summary of results for the most well-adapted blocking *df/dt* and 2FT> class are in Table 7.12. The parameters for this method are:

- $N_{s,1} = 36$
- $N_{s,2} = 10$
- $T_1 = 9.7144 \times 10^{-4}$
- $T_2 = 1.5581 \times 10^{-3}$
- $N_s = 91$
- $T_N = -4.7164 \times 10^{-3}$
- $T_P = 1.9356 \times 10^{-5}$

Table 7.12.: Results for blocking *df/dt* and 2FT> method: unstable events, noiseless

Power (MW)	Event Type	Controls Issued	Local Controls	ΔRMSGA per event	Effectiveness	% Stab
-25	Transient	6.81	1.79	-62.32	366.22	4.3
-25	AN	0.43	0.0	-0.64	60.25	0
-25	AP	14.37	4.55	-73.96	205.93	92.8
-50	Transient	6.48	1.77	-125.62	387.48	9.7
-50	AN	0.43	0.0	-0.79	37.36	0
-50	AP	14.27	4.52	-79.81	111.88	94.8
-75	Transient	6.26	1.77	-165.29	352.17	11.8
-75	AN	0.43	0.0	-1.02	31.83	0
-75	AP	13.52	4.53	-84.39	83.25	96.1

This method had good effectiveness and the highest number of average local controls. It issued many controls for type AP events and few for type AN events. However, this method over-controlled transient events and was less efficient at controlling type AP events, so we declined to investigate this method further.

7.4.4 Blocking *df/dt* and Inverted 2FT>

The summary of results for the most well-adapted blocking *df/dt* and Inverted 2FT> class are in Table 7.13. The parameters for this method are:

- $N_{s,1} = 4$
- $N_{s,2} = 35$
- $T_1 = 9.8162 \times 10^{-3}$
- $T_2 = 7.9037 \times 10^{-3}$
- $N_s = 116$
- $T_N = -1.6455 \times 10^{-2}$
- $T_P = 8.3610 \times 10^{-6}$

Table 7.13.: Results for blocking *df/dt* and Inverted 2FT method: unstable events, noiseless

Power (MW)	Event Type	Controls Issued	Local Controls	ΔRMSGA per event	Effectiveness	% Stab
-25	Transient	2.40	1.01	-14.56	242.84	1.1
-25	AN	0.37	0.0	-0.49	52.75	0
-25	AP	11.51	3.74	-63.46	220.55	78.4
-50	Transient	2.37	1.02	-19.30	163.21	1.1
-50	AN	0.37	0.0	-0.55	29.31	0
-50	AP	11.08	3.7	-74.68	134.83	92.8
-75	Transient	2.43	1.01	-32.48	178.19	3.2
-75	AN	0.37	0.0	-0.65	23.40	0
-75	AP	10.71	7.03	-77.98	97.12	94.1

We see that the blocking *df/dt* and inverted 2FT> method was much better at distinguishing type AP events from type AN and transient events. This method also rarely controls type AN and transient events. Further, it has approximately 10% higher effectiveness compared to the 2FT< and 2FT> methods. However, the inverted 2FT> method tends to favor controlling remote PMUs more than the other methods examined in detail. Since this method nonetheless satisfies most of our criteria at an acceptable level, we investigated the effects of noise on the class.

7.4.5 Blocking *df/dt* and 2FT+

The summary of results for the most well-adapted blocking *df/dt* and 2FT+ class are in Table 7.14. The parameters for this method are:

- $N_{s,1} = 4$
- $N_{s,2} = 73$
- $T_1 = 6.8175 \times 10^{-3}$
- $N_s = 115$
- $T_N = -1.6327 \times 10^{-2}$
- $T_P = 8.3834 \times 10^{-6}$

Table 7.14.: Results for blocking *df/dt* and 2FT+ method: unstable events, noiseless

Power (MW)	Event Type	Controls Issued	Local Controls	ΔRMSGA per event	Effectiveness	% Stab
-25	Transient	3.87	1.2	-30.45	314.62	1.1
-25	AN	0.37	0.0	-0.51	54.90	0
-25	AP	11.94	4.23	-68.20	228.47	83.0
-50	Transient	3.85	1.2	-58.27	302.76	2.1
-50	AN	0.37	0.0	-0.58	31.12	0
-50	AP	11.39	4.18	-76.11	133.62	93.5
-75	Transient	3.81	1.21	-83.62	292.89	6.5
-75	AN	0.37	0.0	-0.67	24.06	0
-75	AP	10.90	4.14	-79.89	97.77	94.1

The blocking *df/dt* and 2FT+ method was the most successful method, as it is able to distinguish between type AP, type AN, and transient events with high accuracy, it has an excellent effectiveness, and it triggers almost all local power injection locations when it detects an event has occurred. Because of this behavior, this method nearly maximizes the total amount of RMSGA that can be reduced in the system via our controls.

We present a sample AP event in a noiseless environment at Hayden 20.0. The vertical lines indicate we applied controls using the most well-adapted blocking with 2FT+ method in the simulation to the power injection locations shown. Only local power injection locations are shown to clearly show the effect of the controls.

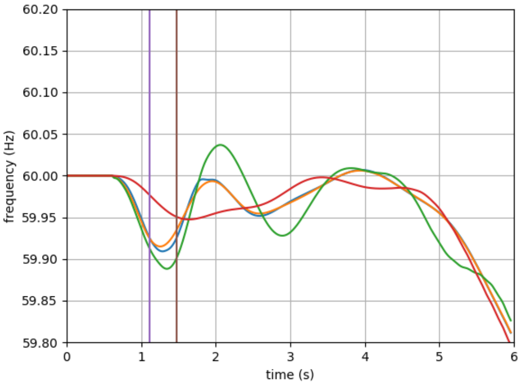

Fig. 7.1.: Type AP event at HAYDEN 20.0

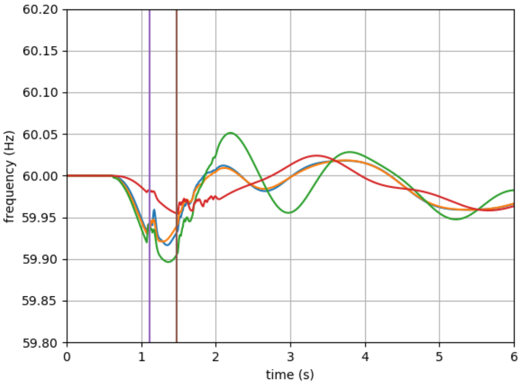

Fig. 7.2.: Type AP event at HAYDEN 20.0 with controls

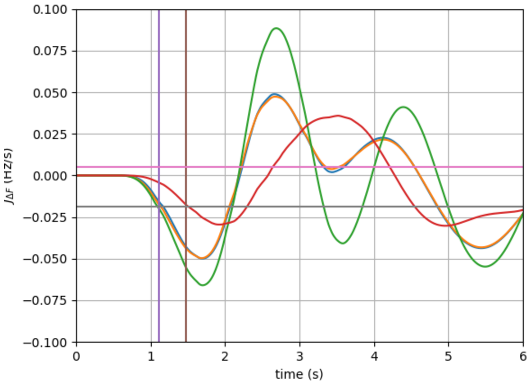

Fig. 7.3.: $J_{\Delta f}$ index with controls

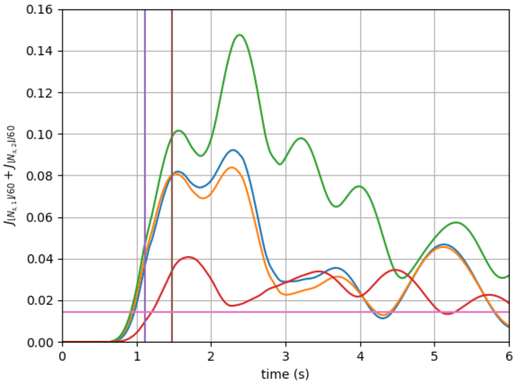

Fig. 7.4.: $J_{\lfloor N_{s,1}\rfloor/60} + J_{\lfloor N_{s,2}\rfloor/60}$ index with controls

This method also satisfies our criteria for a good method. We also tested the class against a noisy environment to determine its robustness.

7.5 Ideal Response and comparison with blocking *df/dt*

We look solely at the ideal response for type AP events, as it only makes sense to deliberately actuate controls near many or all generators automatically when there is a drop in generation capacity as opposed to a mere transient line fault or drop in load. Adding controls everywhere automatically in the event of a load reduction event introduces the possibility to make matters worse and destabilize the system or avoid stabilization where it might otherwise occur. We see the effect of the ideal response in Tables 7.16 and 7.15, and compare it with Tables 7.8 and 7.9.

Table 7.15.: Results for ideal clustered response on type AP events

Power	Type	Local Controls	ΔRMSGA	Effectiveness	% Stab / % Destab
-25	Unstable	4.61	-49.68	430.63	52.3
-25	Stable	4.49	0.36	-3.22	0.5
-50	Unstable	4.61	-71.62	310.43	84.9
-50	Stable	4.49	4.75	-21.17	3.0
-75	Unstable	4.61	-74.89	216.41	93.4
-75	Stable	4.49	20.19	-60.01	8.0

Table 7.16.: Results for ideal non-clustered response on type AP events

Power	Type	Controls Issued	ΔRMSGA	Effectiveness	% Stab / % Destab
-25	Unstable	17.0	-71.01	167.08	78.4
-25	Stable	17.0	-2.18	5.14	0
-50	Unstable	17.0	-83.71	98.48	97.4
-50	Stable	17.0	-3.90	4.59	0
-75	Unstable	17.0	-85.53	67.08	97.4
-75	Stable	17.0	-5.41	4.25	0

Though the ideal response method reduced the average RMSGA more than the most well-adapted blocking *df/dt* method, it did so at the cost of applying many more controls. This caused the effectiveness of the two methods to be approximately the same on unstable AP events. Since the most well-adapted blocking *df/dt* method applied fewer unnecessary controls, it was able to stabilize the network more often than the ideal response method.

We further distinguish the cases when we issue controls only in the same cluster as the event and when we issue controls at all locations. We see that at the 25 MW per control level, issuing controls only in in the same cluster yields an average reduction of 430.6 degrees of RMSGA per GW compared to 167.1 degrees of RMSGA per GW when all controls are issued.

We also see that using all controls at 25 MW per control stabilizes approximately eleven out of fourteen unstable events, while using controls in the same cluster stabilizes about seven out of fourteen events. Finally, at all power injection levels, the ideal clustered method destabilizes some previously-stable events.

7.6 Point-to-Point *df/dt*

Previous methods [4] used decision trees that incorporated point-to-point *df/dt*. We wish to determine how well that previous method compares to our methods.

When the Point-to-Point *df/dt* is allowed to apply a control over the length of the simulation, we found that controls were frequently being applied at the end of the simulation for many AN events. We then adjusted the method to only allow a control up to 60 cycles in the first five seconds of a simulation, which provided improved results for the sake of comparison.

The results for the Point-to-Point *df/dt* with controls issued over the entire six second simulation are in Table 7.17. The results for Point-to-Point *df/dt* with controls issued only in the first second after the fault are in Table 7.18.

In all tables in this section, the following parameters were used: $N_s = 1, T = -0.0006855$. We note that in comparison to the blocking *df/dt* method, the Point-to-Point method does not distinguish between transient events and type AP events very well, nor does it distinguish type AP from type AN events.

Furthermore, both Point-to-Point *df/dt* and a shortened Point-to-Point *df/dt* actuate many controls in stable simulations, while the blocking *df/dt* method actuates about half as many controls in stable simulations in comparison to the shortened Point-to-Point method.

Table 7.17.: Results for Point-to-Point *df/dt*: unstable events, noiseless

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	16.55	-100.69	243.31	0
-25	AN	15.88	6.83	-17.21	0
-25	AP	11.71	-55.48	189.39	71.4
-50	Transient	16.18	-173.40	214.13	9.1
-50	AN	16.13	15.98	-19.82	0
-50	AP	12.79	-78.09	122.10	78.6
-75	Transient	16.00	-282.90	235.85	9.3
-75	AN	15.50	25.10	-21.60	0
-75	AP	11.93	-94.31	105.37	91.7

Table 7.18.: Results for point-to-point *df/dt*: unstable events, noiseless, shortened

Power	Type	Avg. Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	13.55	-96.01	3.53	0
-25	AN	0.25	13.55	-0.46	0
-25	AP	7.71	-52.43	3.68	64.2
-50	Transient	14.45	-148.65	4.86	9.1
-50	AN	0.25	26.18	-0.48	0
-50	AP	8.86	-74.40	5.95	78.7
-75	Transient	14.45	-222.98	4.86	8.9
-75	AN	0.25	38.21	-0.49	0
-75	AP	9.50	-93.52	7.62	91.7

The blocking *df/dt* method was able to stabilize about 15% to 25% more simulations in AP events, which represents a significant and worthwhile improvement on Point-to-Point *df/dt*.

7.7 Best Methods in a Noisy Environment

We know from section 4.2.2 that the performance of the point-to-point *df/dt* method is degraded in the presence of typical electrical noise. We wish to determine if the most well-adapted blocking *df/dt* and 2FT+ method or the most well-adapted blocking *df/dt* and Inverted 2FT> have the same issues.

Data collected in [17] indicates that typical electrical line noise during selected days in 2018 was approximately Gaussian with standard deviation on the order of $10^{-1/2}$. Our use of a standard deviation of 10^0 simulates an unusually noisy environment.

The parameters for the most well-adapted noisy blocking *df/dt* and 2FT+ class are:

- $N_{s,1} = 17$
- $N_{s,2} = 82$
- $T_1 = 5.10025 \times 10^{-4}$
- $N_s = 108$
- $T_N = -1.8559 \times 10^{-2}$
- $T_P = 5.0195 \times 10^{-3}$

Similarly, the parameters for the noisy blocking *df/dt* and inverted 2FT class are:

- $N_{s,1} = 13$
- $N_{s,2} = 41$
- $T_1 = 4.0027 \times 10^{-3}$
- $T_2 = 1.7852 \times 10^{-2}$
- $N_s = 97$
- $T_N = -8.3711 \times 10^{-1}$
- $T_P = 2.1902 \times 10^{-3}$

Tables 7.19 and 7.20 give a clear distinction between the blocking *df/dt* with Inverted 2FT and the blocking *df/dt* with 2FT+ methods. We see that the blocking *df/dt* with Inverted 2FT method has significantly degraded performance in the presence of noise. This is likely due to the fact that the noise we added was approximately evenly distributed across the frequency spectrum, so there were very few windows where the *df/dt* condition was triggered and the two Fourier indices were under a given threshold.

In contrast with the most well-adapted blocking *df/dt* with 2FT> method, the most well-adapted blocking *df/dt* with 2FT+ method performed admirably in the presence of noise, with a reasonable distinction between type AP events and other events. The method also maintained its high effectiveness and good stabilization rates.

Table 7.19.: Results for Blocking *df/dt* with Inverted 2FT class: unstable events, $\sigma = 10^0$

Power	Type	Avg. Controls	Local Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	0.19	0.02	-1.30	269.67	0
-25	AN	0	0	0	-	0
-25	AP	1.17	0.59	-8.02	274.34	7.8
-50	Transient	0.19	0.02	-2.62	271.20	0
-50	AN	0	0	0	-	0
-50	AP	1.18	0.6	-15.02	255.26	18.3
-75	Transient	0.19	0.02	-3.56	245.42	0
-75	AN	0	0	0	-	0
-75	AP	1.18	0.6	-20.98	237.82	25.5

Table 7.20.: Results for Blocking *df/dt* with 2FT+ class: unstable events, $\sigma = 10^0$

Power	Type	Avg. Controls	Local Controls	ΔRMSGA	Effectiveness	% Stab
-25	Transient	4.47	1.21	-32.63	291.80	2.1
-25	AN	3.79	0.09	-4.52	47.76	0
-25	AP	11.98	4.22	-68.96	230.25	80.4
-50	Transient	4.41	1.22	-58.29	264.44	3.2
-50	AN	3.79	0.09	-5.76	30.39	0
-50	AP	11.39	4.18	-78.02	136.98	94.7
-75	Transient	4.34	1.21	-77.56	238.06	6.5
-75	AN	3.79	0.09	-6.73	23.69	0
-75	AP	10.88	4.1	-82.40	101.02	95.4

We provide a sample noisy AP event of 840.0 MW added at Hayden 20.0. The time of control and the index levels marked in Figures 7.5, 7.6, 7.7, and 7.8. Note that only the local PMU frequencies are shown, and the time that controls are issued are marked by

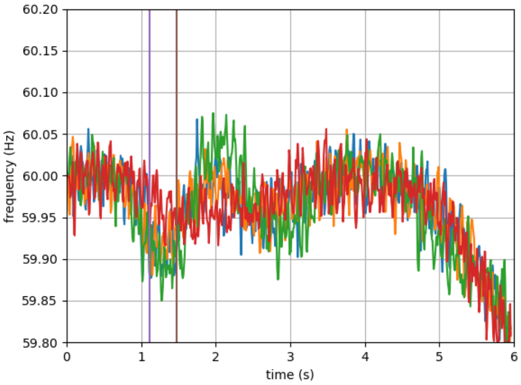

Fig. 7.5.: Type AP event with $\sigma = 10^0$ noise at HAYDEN 20.0 with no controls

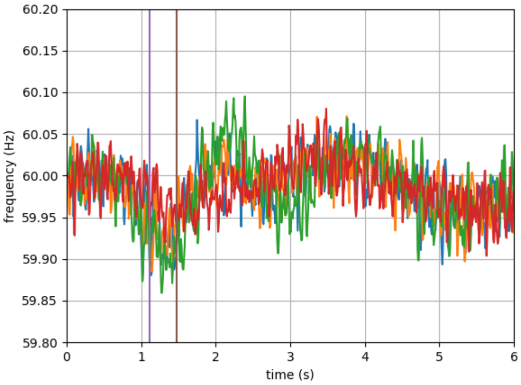

Fig. 7.6.: Type AP event with $\sigma = 10^0$ at HAYDEN 20.0 with controls

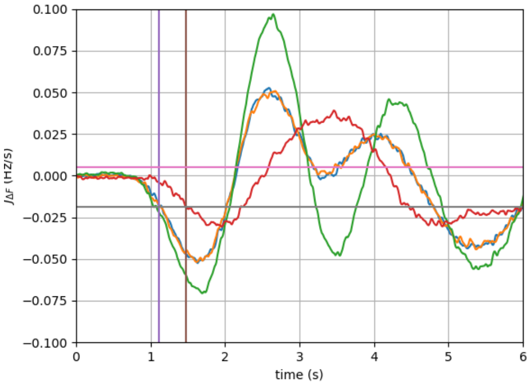

Fig. 7.7.: $J_{\Delta f}$ index with $\sigma = 10^0$ with controls

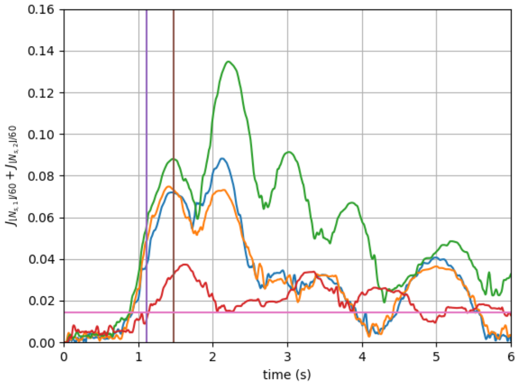

Fig. 7.8.: $J_{\lfloor N_{s,1}\rfloor/60}+J_{\lfloor N_{s,2}\rfloor/60}$ index with $\sigma = 10^0$ with controls

the vertical lines. The thresholds for the respective indices are indicated by the horizontal lines.

8. CONCLUSIONS

We determined that we can use any of several methods to detect local and unstable 1ϕ and 3ϕ faults. We were also able to distinguish AP events from other events. Finally, we verified that our AP-distinguishing method can apply one-shot controls to stabilize the network and reduce stress on the network.

Table 8.1 shows the results of inquiries from phase I. The table shows the best method found for each combination of noisy and noiseless data with methods that use clusters (local) and that do not use clusters (non-local) when we only consider one-phase and three-phase faults.

Table 8.1.: Summary of Best Classes

Local?	Noise	Class	$N_{s,1}$	$N_{s,2}$	t_1	t_2	Cost
Yes	0	2FT>	6	83	0.036	0.028	0.244
Yes	10^0	2FT+	20	120	0.170		0.250
No	0	2FT>	119	2	0.088	0.012	0.176
No	10^0	2FT>	115	12	0.090	0.101	0.178

Phase II showed that the best class for distinguishing type AP events from other types of events was the blocking *df/dt* with 2FT+ class. The parameters for the best method in this class were $N_{s,1} = 67$, $N_{s,2} = 57$, $T_1 = 1.4384 \times 10^{-2}$, the length of the moving average $N_s = 108$, the control threshold $T_N = -1.8559 \times 10^{-2}$, and the blocking threshold $T_P = 5.0195 \times 10^{-3}$, with a cost of 0.1879. Simulations indicated that this blocking with 2FT+ method was efficient and stabilized almost all unstable generator disconnection events while destabilizing no events. Each of these best methods from Phase I and Phase II were robust in simulations with Gaussian noise added that simulated a noisier-than-usual environment.

One challenge we discovered in the course of testing was the bottleneck imposed by the TSAT software package. The TSAT package can only access a critical file once per simulation, which locks out all other simulations while the file is accessed.

In the future, we could evaluate methods using the change in RMSGA rather than the cost function developed in this thesis. Since each simulation takes about one second to run and there are about 10^4 simulations needed for each particle, a single evaluation of RMSGA for all events takes hours rather than the seconds that a single evaluation of the cost function takes. If we could parallelize TSAT to run thousands of instances at once, possibly across a distributed network, then we could evaluate RMSGA directly rather than using the cost function as a proxy for the fitness of the method.

Further, we could also have more than one different level of control available in appliances. The methods in some consumers' appliances could trigger at different thresholds than other appliances, allowing for various stages of controls across the network. Determining well-adapted levels for the various stages would require PSOs with more parameters than used here, but would be similar in scope.

Finally, once we know which method we wish to implement, we can develop the method in consumer appliances using either custom FPGAs or off-the-shelf microcontrollers to carry out the $J_{\Delta f}$ and $J_{\lfloor N_s \rfloor/60}$ calculations and shut down operations that use high levels of power during detected type AP events. This would require extensive testing using recorded transient, type AN, and type AP events to ensure that consumer appliances are temporarily shut down only when needed.

REFERENCES

REFERENCES

[1] W. Sweet, "Restructuring the thin-stretched grid," *IEEE Spectrum: Technology, Engineering, and Science News*, June 2000, last accessed 10/1/2018. [Online]. Available: https://spectrum.ieee.org/energy/the-smarter-grid/restructuring-the-thinstretched-grid

[2] D. Service and Supply, "Industries that would be affected most by a power outage," last accessed 9/17/2018. [Online]. Available: https://www.dieselserviceandsupply.com/Industry_Power_Outages.aspx

[3] K. Mei, S. Rovnyak, and C. Ong, "Clustering-based dynamic event location using wide-area phasor measurements," *An introduction to biometric recognition - IEEE Journals & Magazine*, May 2008, last accessed 3/7/2019. [Online]. Available: http://ieeexplore.ieee.org/document/4483529/

[4] M. N. Nilchi, "Electric utility planning methods for the design of one shot stability controls," Master's thesis, Purdue University, IUPUI, 2012.

[5] S. Rovnyak, June 2017, NSF Grant #1711521, last accessed 8/13/2018. [Online]. Available: https://www.nsf.gov/awardsearch/showAward?AWD_ID=1711521&HistoricalAwards=false

[6] S. Rovnyak, C.-W. Liu, J. Lu, W. Ma, and J. Thorp, "Predicting future behavior of transient events rapidly enough to evaluate remedial control options in real-time," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1195–1203, August 1995.

[7] C. Durkin, E. Eberle, and P. Zarakas, "An underfrequency relay with frequency decay rate compensation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 6, pp. 812–820, 1969.

[8] L. Shih, W. Lee, J. Gu, and Y. Moon, "Application of df/dt in power system protection and its implementation in microcontroller based intelligent load shedding relay," *Conference Record. Industrial and Commercial Power Systems Technical Conference 1991*, pp. 11–17, May 1991.

[9] V. Chuvychin, N. Gurov, S. Venkata, and R. Brown, "An adaptive approach to load shedding and spinning reserve control during underfrequency conditions," *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 1805–1810, November 1996.

[10] H. You, "Self-healing in power systems: an approach using islanding and rate of frequency decline based load shedding," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 174–181, February 2003.

[11] H. Bevrani, G. Ledwich, and J. J. Ford, "On the use of df/dt in power system emergency control," *2009 IEEE/PES Power Systems Conference and Exposition*, 2009.

[12] M. Tarafdar Hagh and S. Galvani, “Minimization of load shedding by sequential use of linear programming and particle swarm optimization,” *Constraints*, vol. 19, 2011.

[13] A. El-Zonkoly, M. Saad, and R. Khalil, “New algorithm based on clpso for controlled islanding of distribution systems,” *International Journal of Electrical Power & Energy Systems*, vol. 45, no. 1, pp. 391–403, 2013.

[14] R. C. Eberhart and Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, 2001.

[15] I. C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection,” *Information processing letters*, vol. 85, no. 6, pp. 317–325, 2003.

[16] R. C. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementation*. Morgan Kaufmann, 2007.

[17] N. Dahal, unpublished data, January 2019.