PARAMETER ESTIMATION OF A COMPOSITE LOAD MODEL USING A HYBRID APPROACH

BY

SIMING GUO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Adviser:

Professor Thomas J. Overbye

ABSTRACT

Accurate dynamic load models are needed to ensure meaningful transient simulation results. We investigate the use of a composite load (CLOD) model to represent an aggregate load. Two algorithms for the parameter estimation of the CLOD model are presented and analyzed. The first uses nonlinear optimization and computes a solution with 2.02% error for a validation case, but requires a 49 dB signal-to-noise ratio (SNR) to achieve this. The second method uses least squares and has an error of 11.7% for the same case, but only requires an 18 dB SNR. Higher accuracy is obtained by setting tighter bounds on the parameters used in the simulations and by using more simulations, up to a cap of approximately 100. In order to apply the above algorithms to digital fault recorder data from a real disturbance, we first convert the three-phase sinusoids to a positive sequence dynamic phasor. The resulting signal has significant noise content, which is filtered using a pseudomedian filter due to its edge-preserving quality. As part of this work, a design methodology for a pseudomedian filter is created, which accepts a cutoff frequency f_c and designs a filter with 3 dB of attenuation at f_c and a stopband at $2.75f_c$. A 120 Hz cutoff frequency is chosen for this work. However, even after post-processing, the measured and simulated signals are very dissimilar, resulting in both algorithms failing to identify a reasonable load model. We conclude that a measurement-based parameter estimation method is ill-suited for a complex, nonlinear load model.

To my parents, for always pushing me to reach higher

ACKNOWLEDGMENTS

I am grateful for the support provided by the Illinois Department of Commerce and Economic Opportunity under their Illinois Center for a Smarter Electric Grid program. I would also like to thank the Commonwealth Edison Company and American Electric Power Co., Inc., for providing resources for this work.

Thank you Professor Overbye, for guiding me on my path of learning.

Thank you mom, dad, and sister, for supporting me every step of the way.

Thank you my friends, for making the journey memorable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CaR	Compare and Resimulate
CLOD	Complex Load Model
CP	Constant Power
DAE	Differential Algebraic Equation
DFR	Digital Fault Recorder
DFT	Discrete Fourier Transform
DL	Discharge Lighting
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IM	Induction Machine
LM	Large Motor
PI/QZ	Constant current (I) real power (P), constant impedance (Z) imaginary power (Q)
PMU	Phasor Measurement Unit
PSS/E	Power System Simulator for Engineering
pu	Per Unit
SM	Small Motor
SNR	Signal-to-Noise Ratio
SSFR	Single Sinusoid Frequency Response
StC	Simulate then Compare
STFT	Short Time Fourier Transform

LIST OF SYMBOLS

α	Attenuation
$\boldsymbol{C}[n]$	The matrix proxy for simulation
ϵ	Total estimation error
f	Frequency in Hz
f_c	Cutoff frequency in Hz
f_N	Normalized frequency in cycles/window
f_s	Sampling rate in Hz
f_{stop}	Stop band frequency in Hz
H_L	A linear filter
H_{NL}	A nonlinear filter
H_{PM}	Pseudomedian filter
i	Instantaneous current in A
I	Current phasor
j	The imaginary unit, $\sqrt{-1}$
L	Length of the filter in samples
n	Sample number
N	The total number of samples
ω	Frequency in rad/s
$\boldsymbol{p}$	Parameters in the CLOD model
$\boldsymbol{p}^*$	Optimal parameter estimation solution

$\boldsymbol{p}_{\text{fict}}$	Fictional correct solution for validation
$\boldsymbol{p}_i$	The parameter vector for simulation i
$\boldsymbol{p}_{\text{real}}$	The real load model
$p(i)$	An element in $\boldsymbol{p}$
P	Real power
$\boldsymbol{P}$	The matrix of parameters
$\boldsymbol{P}_A$	The augmented matrix of parameters
Q	Imaginary power
ρ	The number of simulations used for StC
s	The Laplace variable
S	Complex power
t	Time in seconds
v	Instantaneous voltage
$v[n]$	Time-varying voltage phasor
$v[t]$	The voltage sinusoid signal
$v^*[n]$	Time-varying voltage phasor corresponding to $\boldsymbol{p}^*$
$v_{\text{meas}}[n]$	Measured time-varying voltage phasor
V	Voltage phasor
$\boldsymbol{V}[n]$	The matrix of simulated voltages
$V^+[n]$	Positive sequence time-varying voltage phasor
$V_a[n]$	a phase time-varying voltage phasor
$V_b[n]$	b phase time-varying voltage phasor
$V_c[n]$	c phase time-varying voltage phasor
W	Window length in samples
$\boldsymbol{x}$	Internal states of a load model
$\dot{\boldsymbol{x}}$	Time derivative of $\boldsymbol{x}$
$x[n]$	An input signal
$y[n]$	An output signal

CHAPTER 1

INTRODUCTION

1.1 Motivation

Simulations are a useful tool in power systems because they allow operators and researchers to perform analysis of the electricity grid for planning and operations purposes. Often, they are the only tool available, since running tests on the real electricity grid operating at peak efficiency would decrease this efficiency, and may cause interruptions to service or even damage expensive power system components. Additionally, simulations have other benefits, such as the ability to perform rapid iterations and the repeatability of results. In order for simulation results to be accurate and meaningful, however, simulations must be set up with accurate models of all components of the power system.

Power system components can be roughly placed into three groups: generation, transmission/distribution, and load. Historically, generator and transmission modeling have received the most attention [1]. For example, the most recent Siemens Power System Simulator for Engineering (PSS/E version 33) software manual lists over 100 models pertaining to machines, approximately 40 models of various transmission devices, but only has nine load models [2]. However, due to increasing stresses on the grid from sources such as congestion, aging of equipment, and renewable penetration, load modeling has also become important [1], but this lack of knowledge about loads has made load models a weak link in simulations, where conclusions are conditioned on the load model assumed [3]. For example, [4] showed that in a simulation of a particular inter-area power transfer, the tie-line stability limit found can vary by as much as 50% depending on the load model. In the absence of an accurate and verified load model, an operator in such a situation may have to be overly conservative, thus under-utilizing their resources and increasing costs [5]. A reliable algorithm for the estimation of load model parameters is the goal of this work.

1.2 The Load Modeling Problem

Broadly speaking, a load model is any model which characterizes the behavior of the load depending on the conditions at its connection point to the grid. Usually, it is a function that maps the supply voltage of the load to the complex power demanded by the load: $h : V \mapsto S$. The purpose of load modeling is to estimate $h(V)$.

The general load modeling problem is very similar to generator modeling. They are both just applications of the general field of control engineering known as system identification [6]. Essentially, we are given a plant: a system about which we know only limited information. By applying certain inputs and observing the outputs, we try to obtain knowledge about the plant and characterize it more completely. However, the key difference between generator and load modeling is scale. In the USA, there are approximately 5,800 power plants with a total of 18,000 generators [7], whereas there are over 131 million homes [8]. Additionally, each home represents a diverse range of devices from digital electronics to appliances. Thus, while we can model each of the generators individually, this is not tractable for the loads. Instead, we wish to create an aggregate model (of a city for instance) that has far fewer parameters than the addition of all the individual loads.

1.3 Aggregate Load Modeling Challenges

Aggregate load modeling poses two special challenges, which are not faced during the modeling of other components of the power system. Firstly, while individual components such as transformers or generators can be tested in isolation before they are placed in service (e.g. to find machine torque-speed curves), loads cannot. Thus, we need to develop a method which can create a model of the load while it is on-line and consuming power. The second challenge of load modeling is that aggregate loads are inherently complicated and highly nonlinear (and not even convex) [9]. This is because an aggregate load may contain many diverse components, such as transformers built into devices, power electronics, industrial motors, and constant impedance loads such as a radiator. Even if each of these loads itself is linear, or linearizable around an operating point, aggregating them together will most likely not preserve this property.

1.4 Overview of Thesis

The remainder of the thesis will proceed as follows. In Chapter 2, we introduce load models in more detail, and review major load model classifications and examples of common load models. In Chapter 3, we present the composite load model, define the parameter estimation problem and general solution approach, and then describe the validation case study. In Chapter 4, we implement two parameter estimation algorithms and compare their performance on the case study, as well as identify the merits, challenges, and considerations involved with each algorithm. In Chapter 5, we discuss the application of the parameter estimation algorithm to real world data, and the additional challenges associated with handling measurement data. Finally, we conclude in Chapter 6.

CHAPTER 2

BACKGROUND

2.1 Load Model Classifications

There are different types of aggregate load models, and different methods by which to estimate their parameters. This paper focuses on finding a *dynamic*, *time domain* load model using a *hybrid measurement/component-based*, *top-down* approach. These four concepts are discussed below.

2.1.1 Dynamic versus Static

The conditions under which a model can be used will determine if it is a static or a dynamic model. A static model is one which is only accurate for steady-state operating conditions. However, often operators are more concerned with transient stability, and wish to know if the system remains stable under a particular disturbance, such as a fault. Disturbances cause a system to transition from an initial steady state, through a transient region, to a second operating point. Only a dynamic model can capture the response during the transient region in addition to the initial and final steady states. This is important because even if a system is stable at two operating points, A and B, it can go unstable during the transition between them [3].

2.1.2 Time versus Frequency Domain

There are two general classes of methods for system identification: time domain methods and frequency domain methods. Time domain methods try to estimate a function $h : V(t) \mapsto S(t)$, by analyzing time domain input voltages $V(t)$ and power demanded $S(t)$. On the other hand, frequency domain methods estimate $H : V(\omega) \mapsto S(\omega)$ in the frequency domain. For power system applications, time domain is preferred because ultimately operators want to see the time domain response of the entire system [10] (frequency domain models are more often used in the realm of harmonic analysis), so this paper will focus on a time domain method.

2.1.3 Measurement versus Component Based

There are two time domain methods currently used for load modeling: the component-based approach and the measurement-based approach [10]. The component-based approach determines an aggregate load model at a bus from *a priori* knowledge of the type (e.g. induction motors, electric heating) and quantity of loads connected to that bus, by simply adding them together via a weighted sum [11]. Each of the load types are characterized by predetermined parameters. On the other hand, the only assumption we make in a measurement-based approach is the model structure—the actual parameters in the model are calculated using measurements taken on the grid, typically through the study of disturbances. In [12], the authors recommended that the measurement approach be used to determine the most realistic model. The hybrid method proposed in the paper takes concepts from both approaches, and will be described in detail in Chapter 4.

2.1.4 Bottom-Up versus Top-Down

Measurement-based approaches can be further divided into bottom-up and top-down methods [9]. A bottom-up method uses measurements made at each load (e.g. at each house) to create many small-scale load models, and then adds them together to create an aggregate load model [13]. Each sensor is relatively simple and can have low sampling rates, but a very large number of sensors would be required to create an aggregate load model [9]. On the other hand, a top-down method uses sensors at the feeder or substation level. These sensors are very expensive and have very fast sampling rates, but far fewer are required [9]. Since such infrastructure has already been deployed on the grid (in the form of phasor measurement units (PMUs) and digital fault recorders), we will focus on a top-down approach.

2.2 Common Load Models

There are many load models which are typically found in power system simulation packages. As stated above, they are predominately time domain models, so we list them Table 2.1 categorizes them as either static or dynamic models.

2.2.1 Static Load Models

Four common static load models are listed in Table 2.1. A constant power load has a constant power demand regardless of system voltages. This type of load can only be used for steady state analysis near nominal voltage, since transients could cause the power flow solution to fail for a constant power load. For example, a fault on the load bus would cause the bus voltage to

Table 2.1: Common static and dynamic load models. Note, all voltages are in per-unit.

Static	Constant Power	$P = P_0$ $Q = Q_0$
	ZIP	$P = P_0(\alpha_{p1}V^2 + \alpha_{p2}V + \alpha_{p3})$ $Q = Q_0(\alpha_{q1}V^2 + \alpha_{q2}V + \alpha_{q3})$
	Exponential	$P = P_0 V^{\alpha_p}$ $Q = Q_0 V^{\alpha_q}$
Dynamic	DAE System	$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \Delta V)$ $\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}, \Delta V, \Delta P, \Delta Q)$
	State Space	$\dot{\boldsymbol{x}}_p = \boldsymbol{A}_p\boldsymbol{x}_p + \boldsymbol{B}_p\Delta V$ $\Delta P = \boldsymbol{C}_p\boldsymbol{x} + D_p\Delta V$ $\dot{\boldsymbol{x}}_q = \boldsymbol{A}_q\boldsymbol{x} + \boldsymbol{B}_q\Delta V$ $\Delta Q = \boldsymbol{C}_q\boldsymbol{x}_p + D_q\Delta V$
	Induction Motor	$\Delta P + T_{pp}\frac{d\Delta P}{dt} = K_{pv}\left(\Delta V + T_{pv}\frac{d\Delta V}{dt}\right)$ $\Delta Q = K_{qv}\Delta V + K_{qp}\Delta P$

drop to zero, meaning the power delivered ($S = VI^*$, where * denotes the complex conjugate) could not possibly be non-zero.

The ZIP and exponential load models have a power demand that does vary with voltage. The ZIP model represents the load as a second degree polynomial in voltage, and the exponential load model represents it as an exponential function [11]. While these two models are typically classified as static load models, their voltage dependence allows them to be used in transient studies [14], though they are limited. This is partly because each of these models alone is relatively simple and has relatively few parameters. This limits their potential accuracy, since [15] found a close correlation between the number of parameters (i.e. degrees of freedom) in a model and the accuracy of the model.

2.2.2 Dynamic Load Models

The three dynamic load models listed here were developed specifically for transient simulations. The reader should note that while the static models are functions of V, the dynamic models are functions of ΔV. This is because dynamic models describe the behavior of a system as it deviates from a nominal operating point (P_0, Q_0, and V_0), and these models usually become less accurate for larger deviations. Additionally, while these models have many more parameters and are thus capable of higher accuracy [15], they also require more measurements in order to determine those parameters with sufficient accuracy [16].

The differential algebraic equation (DAE) system model and the state space model are general frameworks that can be used to describe a dynamic system. They make use of a vector of state variables $\boldsymbol{x}$ in order to track the internal states of the model, but the output is still just ΔP and ΔQ. The DAE model represents the load as a combination of differential equations and algebraic equations, and was used in [17]. The state space model is a subset of the DAE model, in that it uses a linear system of equations. It can be implemented either directly, as in [18], or it can be further transformed into a transfer function using $H(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + D$, where $\boldsymbol{I}$ is the identity matrix, as done in [19].

The induction machine (IM) model [20] describes the behavior of an induction motor. There are various degrees of complexity of IM models; the one shown is a relatively simple first order model (T_{pp}, K_{pv}, T_{pv}, K_{qv}, and K_{qp} are the five model parameters). The IM model is conceptually different from the previous two, because it is not a generic model, but is a model specific to power systems. While other specialized models exist, the IM model is highlighted here because induction motors account for 57% of electricity use in the United States [21]. The reader may note that, with the exception of the IM model, the other models are all very generic structures, and not specific to power systems. This is not ideal since we are not fully utilizing the knowledge at hand. Chapter 3 proposes a solution.

CHAPTER 3

CLOD MODEL HYBRID PARAMETER ESTIMATION PRELIMINARIES

In Section 2.1.3, two time domain approaches were described: measurement-based and component-based. Component-based approaches only used *a priori* knowledge of the grid, while measurement-based approaches avoided all such assumptions. In this paper, we study a hybrid approach, which attempts to capture the benefits of both methods. It accomplishes this by using the component-based approach to create the structure of the load model: one that is specific to power systems and unlike the mostly generic models in Table 2.1. Then, it uses the measurement-based approach to determine the parameters of the model.

3.1 The CLOD Model

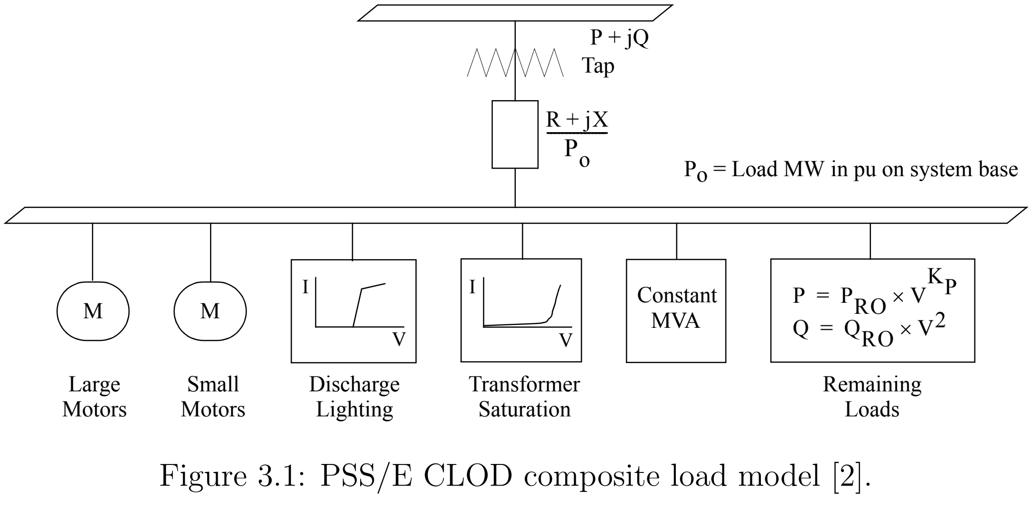

Figure 3.1: PSS/E CLOD composite load model [2].

One class of load models not yet discussed is composite (or complex) load models. These load models are so named because they are an aggregation of multiple models (henceforth called submodels), including, but not limited to, those listed in Table 2.1. The CLOD model, as defined in PSS/E [22], belongs to this class of models and is shown in Fig. 3.1. The CLOD model represents the load as an aggregation of the following five submodels:

- Induction motors: Labeled Large Motors and Small Motors, these two submodels are each characterized by typical torque-speed, current-speed, and power factor-speed curves.

- Discharge lighting: For voltages above 0.75 pu, the real power is modeled as constant current, and the imaginary as exponential with an exponent of 4.5. As voltage decreases below 0.75 pu, both P and Q drop linearly until the light is completely extinguished below 0.65 pu voltage.

- Transformer losses: These are further divided into core losses (saturation) and copper losses $\left(\frac{R+jX}{P_0}\right)$. Transformer losses may be neglected since they are relatively insignificant [3].

- Constant MVA: Constant real and reactive power consumption.

- Remaining loads: The real power is modeled as constant current ($K_P = 1$ in Fig. 3.1) and the imaginary power as constant impedance. This is abbreviated as PI/QZ.

In the CLOD model, the internal parameters of each of the submodels are already defined as above. What we wish to calculate is the percentage allocation for each of the load categories.

The structure of the CLOD model was the contribution of the component-based approach, as it uses *a priori* knowledge about the dominant load classes on the grid to create a load model that is very specific to power systems [21, 3]. The measurement-approach is now applied to determine the load parameters (the percentage allocations).

3.2 The Parameter Estimation Problem and Solution

In order to determine the dynamic load model parameters, we study the response of the power system during disturbances. These disturbances can include generator outages, load steps, and line faults. The type of measurements used can include voltage magnitudes, angles, line flows, and frequencies, but will inevitably be tied to the type of disturbance considered. For example, it is well known that the coupling of real power to voltage magnitude is much weaker than to voltage angle [23], so for a load step, the angle measurements are preferred for the analysis. For this paper, we study the effect of faults on bus voltage magnitudes. For a given fault on the system, the approach taken is similar to that described in [16]:

1. Guess the load model parameters, $\boldsymbol{p}$, with $p(i) \geq 0$ and $\sum_i p(i) = 1$.

2. Simulate the fault in PSS/E using the load model in Step 1 to generate voltage waveforms at each bus. Concatenate the waveforms from all the buses, forming $v_{\boldsymbol{p}}[n]$.

3. Compare $v_{\boldsymbol{p}}[n]$ with the actual waveform measured by sensors on the grid, $v_{\text{meas}}[n]$.

4. Repeat Steps 1 to 3 with successively better guesses of $\boldsymbol{p}$ using least squares.

Mathematically, this can be restated as the following least squares minimization problem:

$$\begin{aligned} \boldsymbol{p}^* = & \underset{\boldsymbol{p}\geq 0}{\operatorname{argmin}} \left\{ ||v_{\boldsymbol{p}}[n] - v_{\text{meas}}[n]||_2^2 \right\} \\ & \text{s.t.} \sum_i p(i) = 1 \end{aligned} \tag{3.1}$$

A perfect parameter estimation algorithm would calculate a $\boldsymbol{p}^*$ equal to the real parameters, $\boldsymbol{p}_{\text{real}}$. However, since $\boldsymbol{p}_{\text{real}}$ is unknown, the best realistic algorithm would calculate $\boldsymbol{p}^*$ such that $v_{\boldsymbol{p}^*}[n] = v_{\text{meas}}[n]$. Herein lies a crucial assumption: there must be a one-to-one mapping between the parameters, $\boldsymbol{p}$, and the measurements, $v[n]$, for both the simulation *and* the real power grid. That is, both the fault and the simulation must be:

1. Deterministic: A given $\boldsymbol{p}$ should always produce the same $v[n]$.
2. Injective: Two different sets of parameters $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ should not produce the same $v[n]$.

For the real power grid, neither determinism nor injectivity can be guaranteed, and it is not something we can control. For the simulation, determinism is a given as long as we do not use stochastic methods such as Monte Carlo or purposely introduce randomness (e.g. white noise). Injectivity of the simulation, while not guaranteed, can be controlled. As an extremely simplistic example, if the simulation is linear, then different inputs of $\boldsymbol{p}$ will result in different outputs $v[n]$. When the simulation (or a component, such as the load model) is highly nonlinear and not unimodal, injectivity of the simulation also cannot be guaranteed. The implications of this will be discussed in further detail in Section 4.1.3.

3.3 Validation

In order to validate our algorithm, we require a set of measurements for which we know the real load parameters. We can then use the difference between the estimated parameters and the real parameters to judge the algorithm's efficacy. However, such data does not exist. Therefore, we need to generate a set of fictional voltage data from a set of fictional parameters, which we will label $v_{\text{fict}}[n]$ and $\boldsymbol{p}_{\text{fict}}$, respectively, and pretend this is the "real" load model on the system. For this work, the power system used was the 37 bus case provided in [23], shown in Fig. 3.2. The disturbance simulated was a three-phase balanced fault at bus Demar69 (highlighted), with fault impedance $Z = j1\text{e}{-12}$ pu and cleared after 0.5 seconds. The fictional load model parameters arbitrarily assigned are listed in Table 3.1. The five bold ones are the parameters the algorithm will solve for.

Table 3.1: Parameters designated as the fictional load model. $\boldsymbol{p}_{\text{fict}} = \begin{bmatrix} 0.24 & 0.19 & 0.3 & 0.09 & 0.18 \end{bmatrix}^T$

Parameter	Value
Large motor	**24%**
Small motor	**19%**
Discharge lighting	**30%**
Transformer saturation	Neglected
Constant MVA	**9%**
PI/QZ	**18%**
Transformer R and X	Neglected

In order to quantify the performance of the algorithm, we need to define an error metric. Since the CLOD parameters are represented in a vector, a natural metric would be the Euclidean distance between the estimate and the correct solution:

$$\epsilon = ||\boldsymbol{p}_{\text{fict}} - \boldsymbol{p}^{*}||_2 \tag{3.2}$$

We will refer to this as the total estimation error.

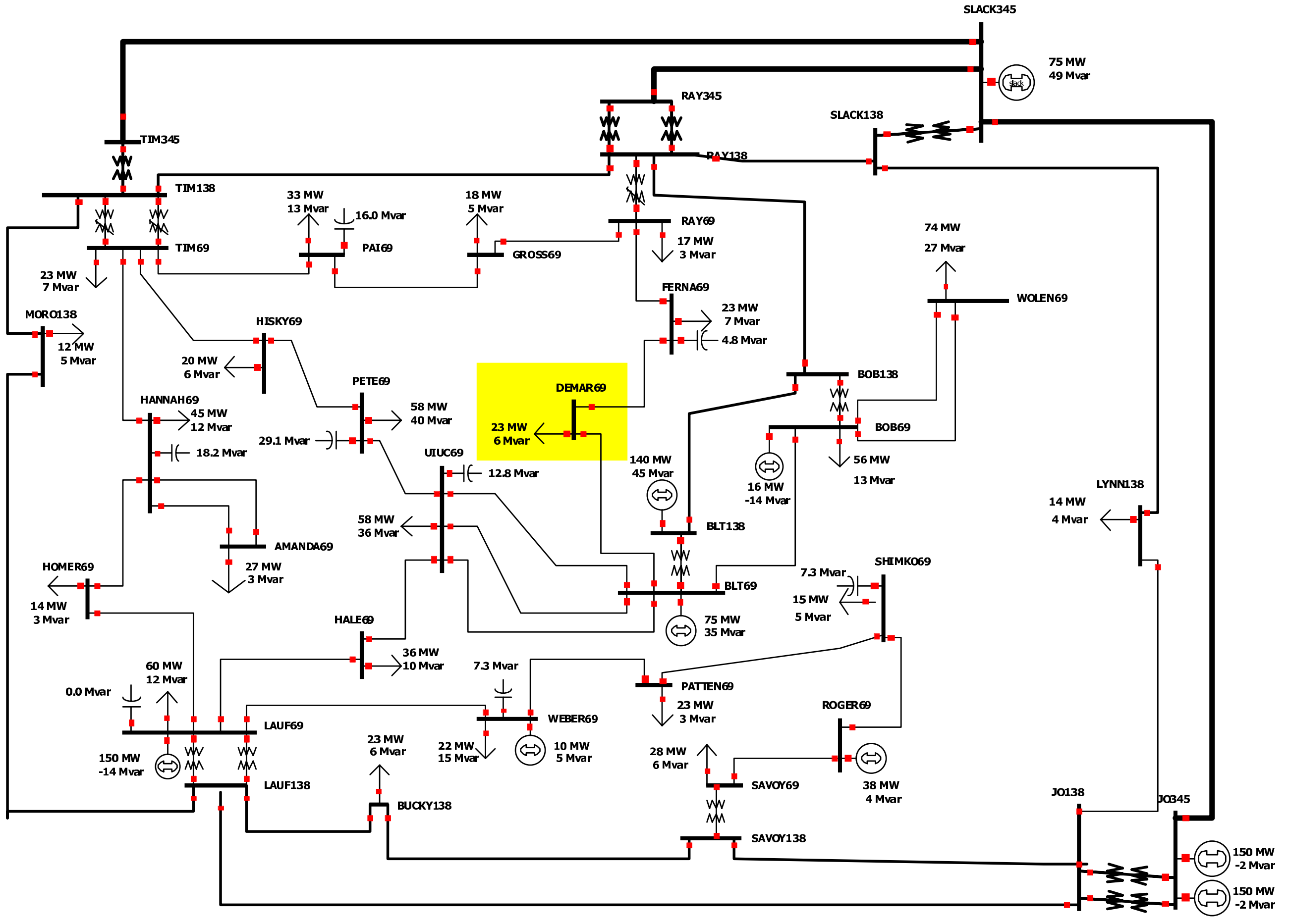

Figure 3.2: 37 bus case from [23] used for validation (fault bus highlighted).

CHAPTER 4

PARAMETER ESTIMATION IMPLEMENTATIONS

This paper will investigate two algorithms for calculating $\boldsymbol{p}^*$: Compare and Resimulate, and Simulate then Calculate. For each, the algorithm will be described, followed by an analysis of its performance on the parameter estimation problem, and ending with a comparison of the two methods as well a discussion of obstacles to be overcome.

4.1 Compare and Resimulate

4.1.1 Algorithm

In Compare and Resimulate (CaR), we solve (3.1) using a general nonlinear optimization routine. For this work, the MATLAB function *fminsearch* was used, which uses a Nelder-Mead simplex algorithm [24]. This algorithm makes use of a simplex, and uses evaluations of the cost function at the vertices to successively shrink and reflect the simplex to converge to the solution. This method was used because it is a derivative-free algorithm, to minimize the risk of divergence given the nonlinear nature of the objective function.

4.1.2 Results and Discussion

The starting guess for *fminsearch* was set to 20% for each of the five parameters to be estimated. Plotted in Fig. 4.1 are the values of $\boldsymbol{p}$ at each iteration of the algorithm, and the horizontal lines are the values of $\boldsymbol{p}_{\text{fict}}$. Fig 4.1 shows the algorithm converging to within a tolerance of $||\boldsymbol{p}_i - \boldsymbol{p}_{i-1}||^2 <$ 1e−4 in 188 function evaluations, where $\boldsymbol{p}_i - \boldsymbol{p}_{i-1}$ is the change in p between the previous and the current iterations. Table 4.1 lists the final solution, and the Absolute Error $\boldsymbol{p}^* - \boldsymbol{p}_{\text{fict}}$. The reader can see that the algorithm successfully estimates the parameters with a very low error. The runtime of CaR for this validation case study was 42 minutes.

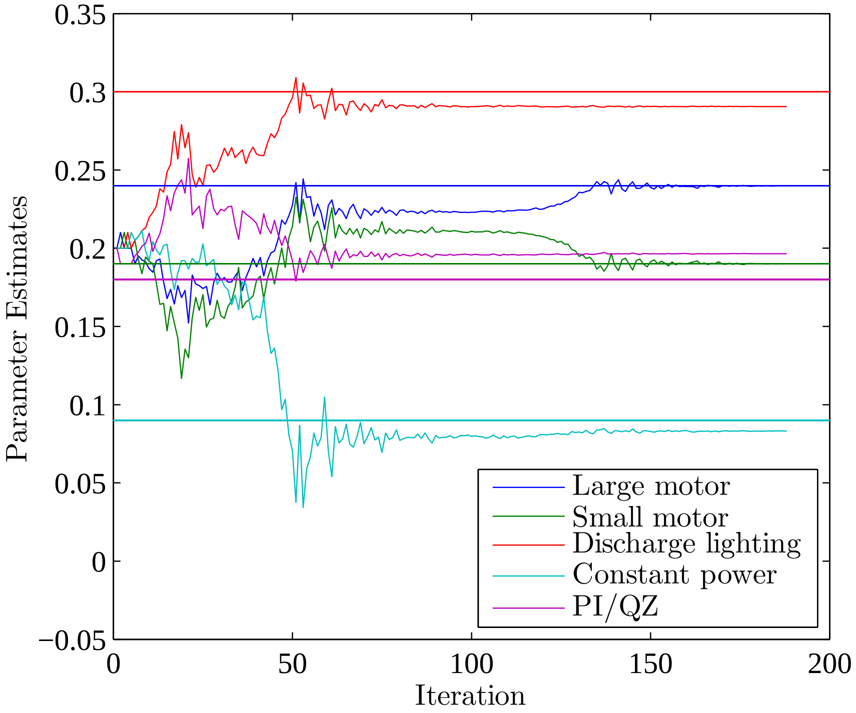

Figure 4.1: Convergence of the CaR algorithm on $\boldsymbol{p}^*$ in validation study.

4.1.3 The Challenge of Noise

From Table 4.1, it would seem that this algorithm performs well. However, since this validation was done using simulated data, one aspect not yet considered is measurement error or noise, which would be present if real sensors were used. Fig. 4.2 shows the results of 426 separate simulations, each with

Table 4.1: CaR solution and error in validation study.

PARAMETER	CAR SOLUTION	ABSOLUTE ERROR
Large motor	23.98%	−0.021%
Small motor	19.00%	+0.00089%
Discharge lighting	29.06%	−0.94%
Constant MVA	8.31%	−0.69%
PI/QZ	19.60%	+1.60%
	TOTAL ERROR	2.02%

a different set of parameters $\boldsymbol{p}_i$, compared against the $v_{\text{fict}}[n]$ generated from $\boldsymbol{p}_{\text{fict}}$. The X-axis value for each one of the 426 data points is Parameter Error, as defined in (4.1), and the Y-axis value is the Residual, as defined in (4.2).

$$\text{Parameter Error} = ||\boldsymbol{p} - \boldsymbol{p}_{\text{fict}}||_2 \tag{4.1}$$

$$\text{Residual} = \sum_{n=1}^{N} (v[n] - v_{\text{fict}}[n])^2 \tag{4.2}$$

Ideally, we would like to see that as the parameter error increased (i.e. $\boldsymbol{p}$ increasingly different from $\boldsymbol{p}_{\text{fict}}$), the difference in the simulated waveform would also increase monotonically. While this general trend is seen in Fig. 4.2, it is clear that a monotonic increase does not exist. In other words, two vastly different sets of parameters may produce waveforms that are nearly identical.

To see this effect more clearly, we can show how the cost function defined in (3.1) changes as we move away from $\boldsymbol{p}_{\text{fict}}$. Since the parameter space is 5-dimensional, we will resort to showing 2-dimensional cross sections. In each subfigure inside Fig. 4.3, two parameters are selected, and a contour of the $\log(\cdot)$ of the residual is plotted as those two parameters are varied. Of the remaining three parameters, two are set to their values in $\boldsymbol{p}_{\text{fict}}$. The remaining parameter is designated the slack parameter, and its value changes

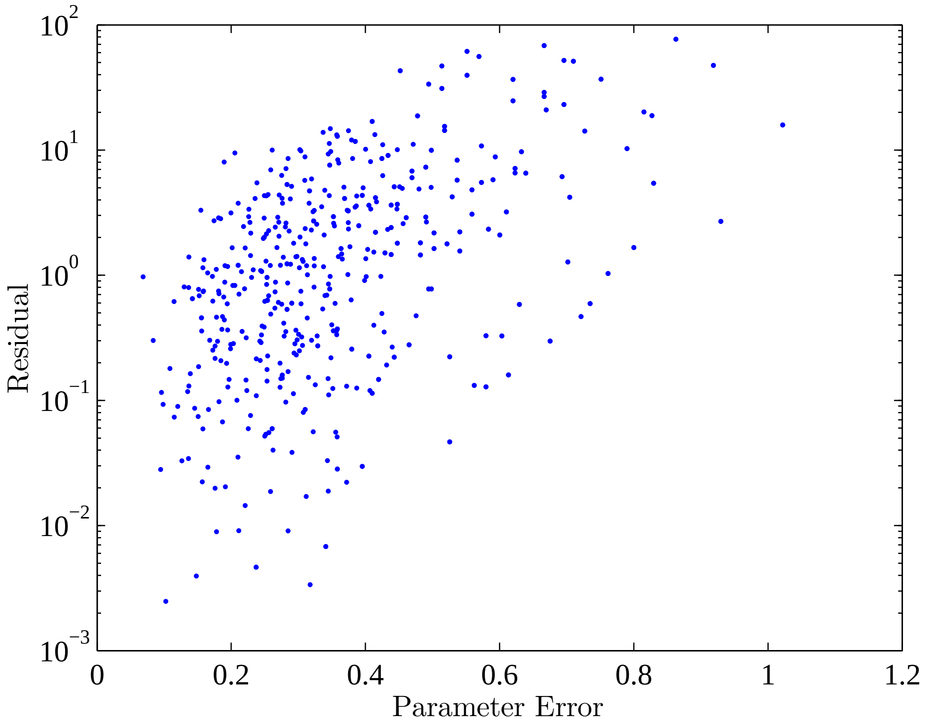

Figure 4.2: The residual of 426 simulations as a function of the parameter distance. Note that the Y-axis uses a logarithmic scale.

in order to preserve $\sum_{i=1}^{5} p(i) = 1$. For example, in the first figure, Small Motors and Large Motors are plotted, and Discharge Lighting and Constant Power are set to 30% and 9%, respectively (see Table 3.1). The dotted lines in each subfigure identify the values of Large Motor and Small Motor in $\boldsymbol{p}_{\text{fict}}$, so they are located at 24% and 19%, respectively. For all the subfigures in Fig. 4.3, the PI/QZ load was chosen as the slack parameter. When the analysis is performed with other load types as the slack parameter, the results are similar and are thus omitted.

There are two features in Fig. 4.3 that are of interest. Firstly, many of the contours show multiple local minima. This agrees with the preliminary analysis from Fig. 4.2.

The second point to note in Fig. 4.3 is that we can identify two distinct

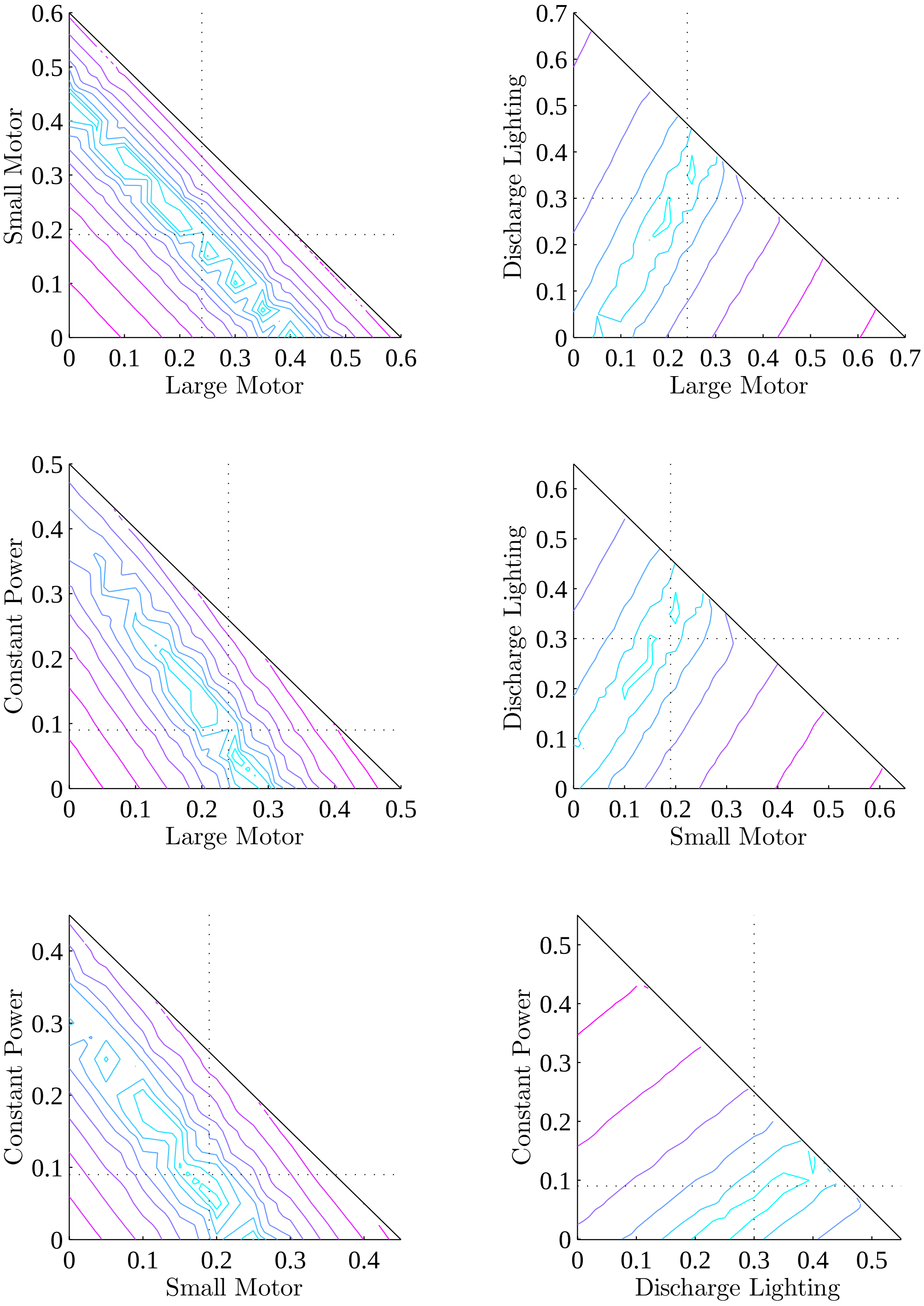

Figure 4.3: Contours of the value of the cost function (on a log scale) near $\boldsymbol{p}_{\text{fict}}$. The PI/QZ load model serves as the slack parameter.

contour shapes. In the left column, the contours are highly elongated in the northwest-southeast direction, while in the right column, the contours are elongated in the southwest-northeast direction. In the former case, this signifies that the two load types plotted behave similarly. For example, moving load between Large Motors, Small Motors, and Constant Power has relatively little effect on the Residual, and by implication, the resulting simulated transient response. On the other hand, contours elongated in the other direction signify that the two load types are disparate. That is, Discharge Lighting acts very differently compared to Large Motors, Small Motors, or Constant Power loads. Additionally, because PI/QZ loads are the slack parameter, we can also conclude from these contours that Discharge Lighting and PI/QZ loads behave similarly (a plot of Discharge Lighting versus PI/QZ indeed shows contours in the northwest-southeast direction).

We can see this effect in the time domain as well. In Fig. 4.4, we plot the voltage at a bus close to the fault bus. We can see that Discharge Lighting and PI/QZ loads recover quickly, while the Large Motor, Small Motor, and Constant Power loads recover much more slowly and overshoot the pre-fault voltage more. Since we now have two groups of loads that perform similarly, the injectivity condition established in Section 3.2 may no longer hold, resulting in the algorithm possibly converging to a grossly incorrect $\boldsymbol{p}^*$.

With the addition of measurement error or noise, this effect is further exacerbated. To quantify the effect of noise on the algorithm's performance, we can add artificial noise to the measurements, and then view the resulting change in the algorithm's parameter estimation error. In this analysis, we assume that the noise is *white*, meaning that it contains all frequency components equally. Intuitively, an increase in the amplitude of the noise, or a

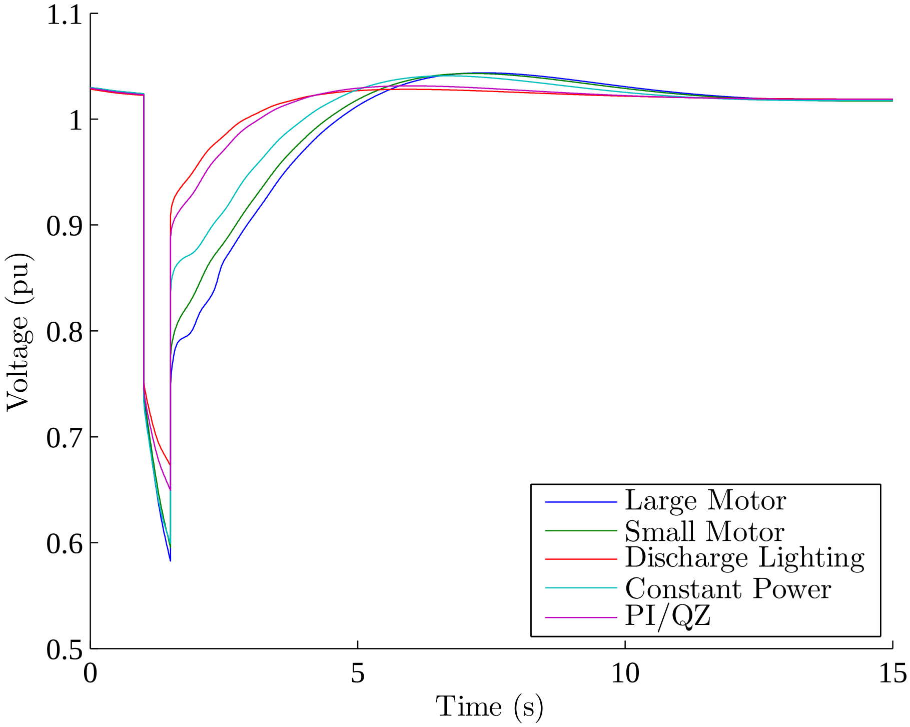

Figure 4.4: The response of each submodel to the validation disturbance.

decrease in the signal-to-noise ratio (SNR) should cause an increase in the estimation error. This is indeed seen in Fig. 4.5. However, Fig. 4.5 also shows that an SNR of 49dB or higher is necessary for the error to stabilize; below this value, the error behaves erratically as the varying levels of noise cause the global minimum to alternate unpredictably between one of several candidate local minima. An SNR of 49 dB is not easy to achieve, as it states that the total noise from all sources (including the sensor measurement error, as well as all external sources of noise) can have an amplitude of at most 0.355% of the desired signal amplitude.

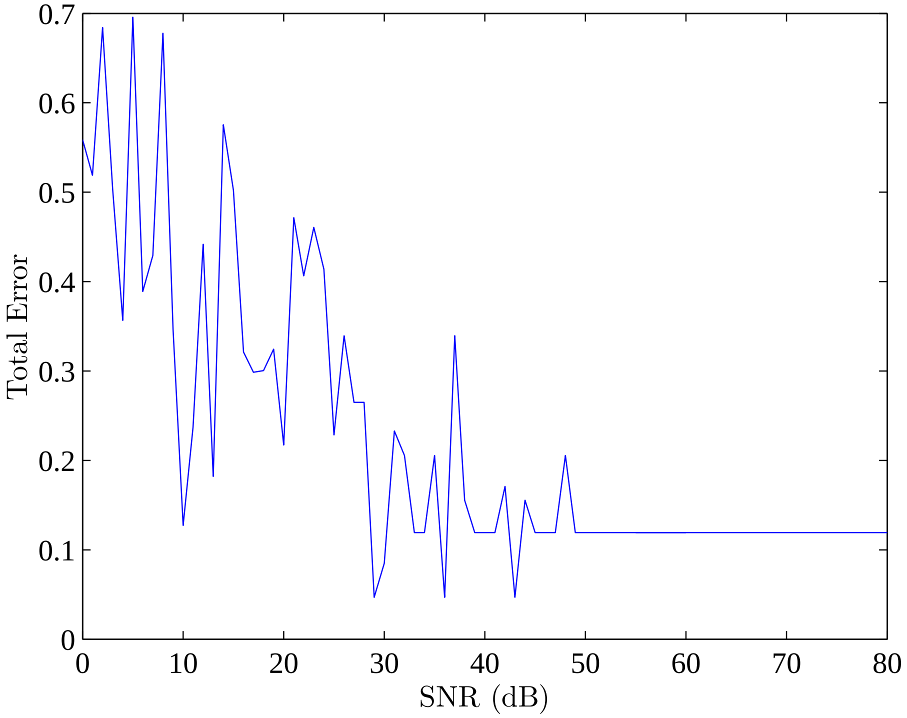

Figure 4.5: The total estimation error for CaR as a function of the measurement SNR.

4.2 Simulate then Calculate

Besides the sensitivity to noise stated above, another issue plaguing the CaR algorithm is the computation time required. Because the objective function contains a PSS/E simulation, that simulation must be performed for each of the 188 function evaluations, for the new value of $\boldsymbol{p}_i$ at that iteration, which leads to the 42 minute computation time. Next we introduce an alternative algorithm, Simulate then Calculate, based on [25].

4.2.1 Algorithm

In Simulate then Calculate (StC), we create a proxy which emulates the simulation, but has a much shorter computation time. It uses a combination of least squares fitting and matrix multiplication, both of which are fast calculations (compared to a simulation). The proxy works as follows:

1. Define ρ sets of parameters with a chosen distribution, and with $\rho >> 5$, creating $\boldsymbol{P} \in \mathbb{R}^{\rho \times 5}$. Also define $\boldsymbol{P}_A$ as the augmented matrix $[\ \mathbf{1} \mid \boldsymbol{P}\ ] \in \mathbb{R}^{\rho \times 6}$.

2. Simulate the fault for each of the ρ parameter sets, creating waveforms $\boldsymbol{V}[n] \in \mathbb{R}^{\rho \times N}$, where N is the number of samples in the simulation period.

3. Solve $\boldsymbol{P}_A \boldsymbol{C}[n] = \boldsymbol{V}[n]$ for $\boldsymbol{C} \in \mathbb{R}^{6 \times N}$ in the least squares sense: $\boldsymbol{C} = \left(\boldsymbol{P}_A{}^T \boldsymbol{P}_A\right)^{-1} \boldsymbol{P}_A{}^T \boldsymbol{V}$.

4. Solve for $\boldsymbol{p}^* = \operatorname{argmin}_{\boldsymbol{p}} \{||v_{\boldsymbol{p}}[n] - v_{\text{meas}}[n]||_2^2\}$
$= \operatorname{argmin}_{\boldsymbol{p}} \{||[\ 1 \mid \boldsymbol{p}\ ] \cdot \boldsymbol{C}[n] - v_{\text{meas}}[n]||_2^2\}$.

The matrix $\boldsymbol{C}[n]$ generated in Step 3 is the proxy, since it maps parameters to waveforms. We augment a column of ones to the parameter matrix $\boldsymbol{P}$ in order to provide one additional degree of freedom, since we are attempting to replace a nonlinear process (a simulation) with a linear one (the matrix $\boldsymbol{C}$). Additionally, the reader may note that Step 4 is now just a linear least squares problem, which has the analytical solution $\boldsymbol{p}^* = (v_{\text{meas}} - \boldsymbol{C}_1)\, \boldsymbol{C}_{2\text{-}6}{}^T \left(\boldsymbol{C}_{2\text{-}6} \boldsymbol{C}_{2\text{-}6}{}^T\right)^{-1}$, where $\boldsymbol{C}_1$ is the first row of $\boldsymbol{C}$, and $\boldsymbol{C}_{2\text{-}6}$ are the second through sixth rows.

4.2.2 Results and Discussion

For the validation case, 126 simulations were used ($\rho = 126$). Since this algorithm replaces an optimization that includes a simulation at every iteration with a series of matrix multiplications, we expect the computation to be much faster. Indeed, we do see a large reduction in computation time: the StC algorithm ran in 3.2 seconds. However, due to this linearization, the accuracy of the algorithm suffered, with an increase in total error from 2.02% to 11.7%, shown in Table 4.2.

Table 4.2: StC solution and error in validation study, using 126 simulations with a bounded uniform distribution.

PARAMETER	CAR SOLUTION	ABSOLUTE ERROR
Large motor	24.44%	+0.44%
Small motor	16.97%	−2.03%
Discharge lighting	25.60%	−4.40%
Constant MVA	5.11%	−3.89%
PI/QZ	27.88%	+9.88%
	TOTAL ERROR	11.7%

One may note that, while the solution algorithm ran in 3.2 seconds, we had to perform 126 simulations beforehand as an input into the algorithm, which would not have saved significant time compared to the 188 simulations the CaR algorithm ran. However, these 126 simulations can be saved, and never need to be repeated. They can be reused if, at a later time, we make improvements to the algorithm and wish to recalculate the load parameters. Alternatively, if we wish to improve the parameter estimation by using a larger set of simulations, we can simply add on to the simulations we already performed.

Aside from reduced computation time, the second motivation of this algorithm was to reduce the impact that noise had on the solution. The reader may recall that for the CaR algorithm, an SNR of 49 dB was required. Fig. 4.6 shows how the error for the StC algorithm changes as a function of SNR. As the reader can see, in this case an SNR of only 18 dB is required to ensure consistent results. This corresponds to a noise content with 35 times higher amplitude. Thus, the StC algorithm is also far more robust to measurement noise than the CaR algorithm.

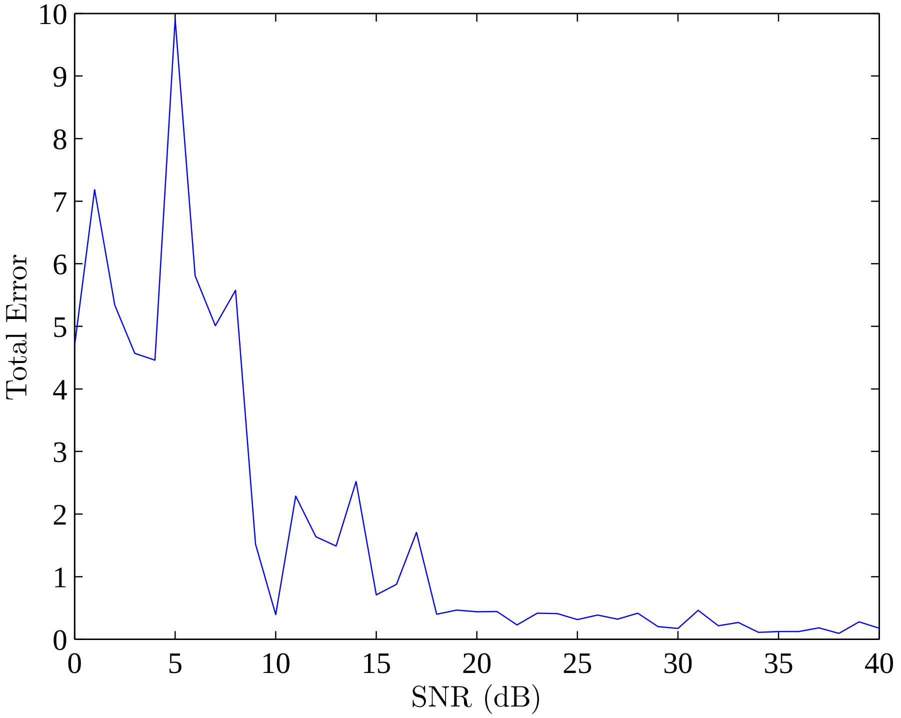

Figure 4.6: The total estimation error for StC as a function of the measurement SNR.

One further consideration for the StC algorithm is how many simulations are required for us to be confident about the parameter estimation solution. For five degrees of freedom, theoretically only five points (i.e. simulations) are required to define a unique solution, but our confidence in this solution would be poor. More simulations would improve our confidence, but at the expense of computation time and storage space. The blue plot in Fig. 4.7 shows how the total error (ϵ) changes when we increase the number of simulations used (ρ). The red plot shows the error (or reduction thereof) that each additional simulation introduces: $\dfrac{d\epsilon}{d\rho}$. The reader can see that with fewer than 20 simulations, the error can change drastically with each new additional simulation—meaning our confidence in the solution is poor. However, after about 100 simulations, with a few exceptions, more simulations did not significantly change the result of the algorithm. Thus, we conclude that in this case, 100 or more simulations is sufficient.

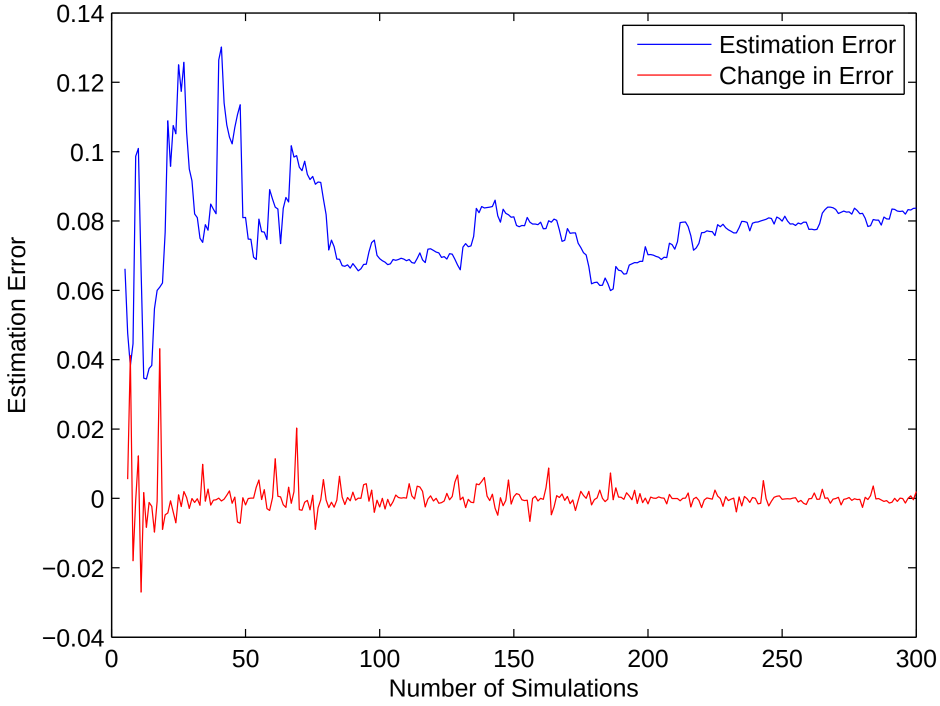

Figure 4.7: The total estimation error for StC as a function of the number of simulations used is shown in blue. The derivative of the estimation error is shown in red.

4.2.3 Challenge of Parameter Bounds

Like the CaR algorithm, this algorithm also has a vulnerability which is not immediately apparent from the results. In Step 1 in the StC algorithm, we choose sets of parameters to be simulated, based on a chosen distribution. For Table 4.2, 126 sets of parameters were created based on a uniform distribution bounded by $[0.05, 0.5]$. That is, for each of the parameters in $\boldsymbol{p}$, it must lie between 5% and 50%, while still adhering to $\sum_i p(i) = 1$. These limits were chosen arbitrarily based on the reasoning that out of 5 load types, one particular type should not dominate (i.e. greater than 50%), and none should be so insignificant as to be less than 5%. An alternative approach would be to use an unbounded uniform distribution, where the bounds are relaxed to 0% and 100%. The parameter estimation result obtained for the unbounded distribution is shown in Table 4.3, where we can see that three of the five parameter estimates are grossly inaccurate.

Table 4.3: StC solution and error in validation study, using 126 simulations with an unbounded uniform distribution.

PARAMETER	STC SOLUTION	ABSOLUTE ERROR
Large motor	20.49%	−3.51%
Small motor	16.26%	−2.74%
Discharge lighting	−2.98%	−32.98%
Constant MVA	−11.95%	−20.95%
PI/QZ	78.17%	+60.17%
	TOTAL ERROR	71.88%

In order to further study how prudently established bounds on the parameter distribution can improve the performance of the algorithm, Fig. 4.8 shows how the total estimation error changes as we impose tighter bounds. On the horizontal axis, the parameter distribution bound is the radius of the hypersphere centered on the correct solution, $\boldsymbol{p}_{\text{fict}}$, where the hyphersphere defines the bounds on each set of parameters $\boldsymbol{p}_i$ that we use for the StC algorithm. The reader can see a clear increase in the error as the bounds are relaxed, which confirms our initial hypothesis that the tighter we set the bounds, the better our estimation error. In other words, the more we know about our system before using the StC algorithm, the better the results will be from using the algorithm.

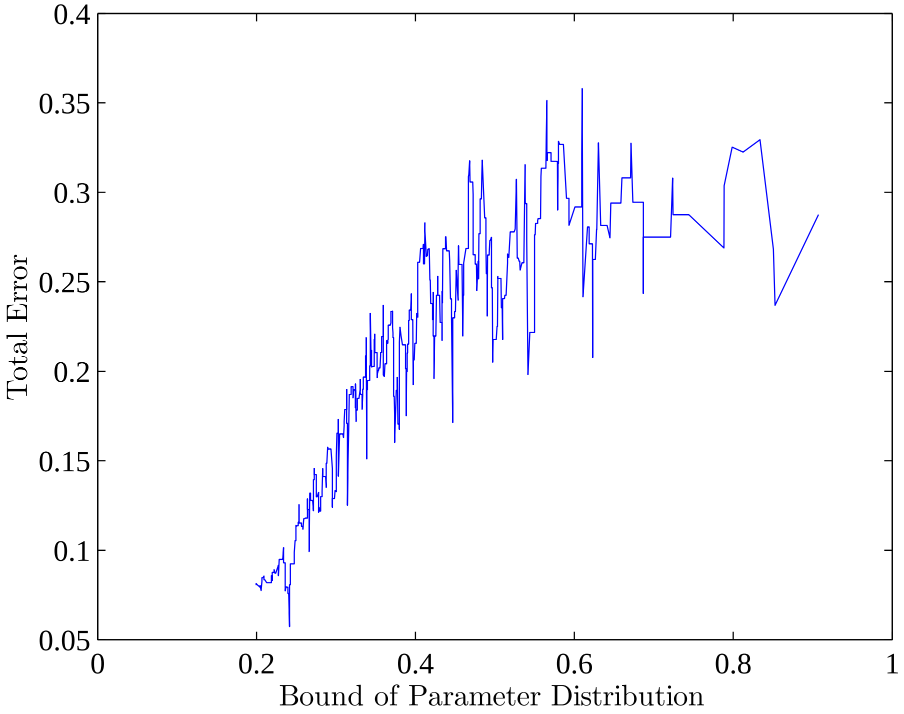

Figure 4.8: The total estimation error for StC as a function of the bound on the parameter distribution.

One note regarding the above analysis that should be pointed out is that, as the bounds are relaxed, a larger number of simulations are encompassed within the growing hypersphere. In order to remove the effects seen in Fig. 4.7, this analysis was performed using a constant 100 simulations, chosen at random from within the bounding hypersphere.

CHAPTER 5

APPLICATION TO REAL DATA

In Chapter 4, two algorithms for solving the parameter estimation problem were presented, and were verified using the simulated validation case study. In this chapter, we will apply the algorithms to real measurements from an actual disturbance on the system.

5.1 Converting Three-Phase Time Domain to Phasor Domain

The analysis performed in Chapter 4 assumed that the measurement data is in the same form as the simulation data. That is, both were in the form of time-varying positive sequence phasors. However, of the sensors currently deployed on the grid, only PMUs output such measurements. Other sensors such as digital fault recorders (DFRs) output the raw three-phase voltage sinusoids. Most importantly, DFRs are much more abundant than PMUs. As of 2011, 137 PMUs were installed on the Western Interconnection (WECC) [26]. While it is unknown exactly how many DFRs are deployed, a case in point is Reliant Energy, which operates only 7% of the total generation of the WECC, but has 33 DFRs alone [27, 28, 29]. While this is in no way a proof, it should convince the reader that to develop an algorithm that is exclusive to PMUs, and cannot accept DFR data, would be unwise.

In light of this, we need to be able to convert sinusoidal data into phasor data. To do this, we use the concept of a discrete Fourier transform (DFT). For this analysis, we are interested in the fundamental component of the Fourier transform, so that we can calculate the phasor V corresponding to the discrete time series $v[t]$ [30]:

$$V = \frac{\sqrt{2}}{W} \sum_{t=1}^{W} v[t] e^{-j\frac{2\pi t}{N}} \tag{5.1}$$

Equation (5.1) calculates a steady state phasor (V has no time dependence). In order to generate a time-varying phasor $V[t]$, we use a short time Fourier transform (STFT). That is, for a sampling rate of f_s on a 60 Hz system, we define a window length $W = \frac{1}{2}\frac{f_s}{60}$, such that the STFT is over half a period (to prevent spectral leakage). Actually, the window can be any integer multiple of a half of a period, but longer windows reduce the time resolution unacceptably. One additional notation that we will introduce is the subscripts a, b, and c, which refer to the measurements of the three voltage phases:

$$V_a[n] = \frac{\sqrt{2}}{W} \sum_{t=n}^{n+W-1} v_a[t] e^{-j\frac{2\pi t}{N}} \tag{5.2}$$

Equation (5.2) converts the sinusoidal measurements of phase a into a dynamic phasor. The equations for the b and c phases are identical. In order to extract the positive sequence component of the three phase dynamic phasors, we use the concept of symmetrical components [23]:

$$V^{+}[n] = \frac{1}{3}\left(V_a[n] + V_b[n] e^{j\frac{2}{3}\pi} + V_c[n] e^{-j\frac{2}{3}\pi}\right) \tag{5.3}$$

Now that the voltages have been converted to dynamic positive sequence phasor representation, we can compare them to the simulation results from

PSS/E. Note that in the rest of this document, whenever we used $v[n]$, we were in fact using the positive sequence voltage defined in (5.3).

5.2 Filtering of Noise

One issue that we have only touched on briefly so far is measurement noise. Measurement noise can come from a multitude of sources, such as: loads turning on and off, high frequency switching from power electronics, weather effects, measurement error of the sensors themselves, and even from post-processing such as the STFT above. Fig. 5.1 shows some sample voltage data from a DFR, after it has been converted to phasor representation. The reader can see that significant noise exists. We can use filtering in order to reduce this noise.

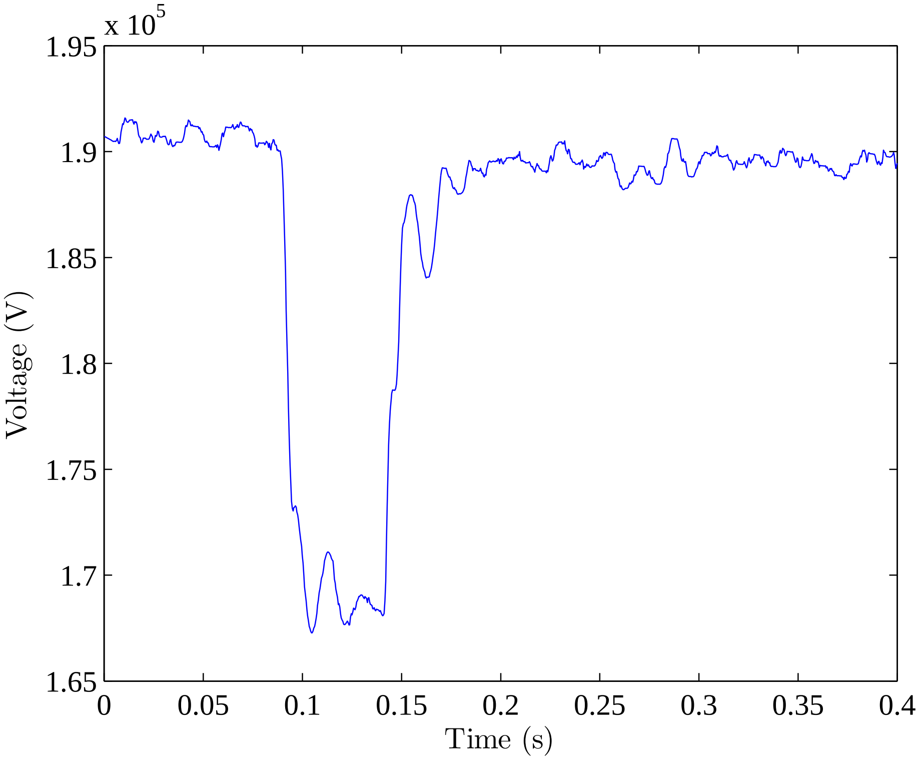

Figure 5.1: Example unfiltered voltage data after conversion to phasor representation.

There are many types of filters that we can use, such as finite impulse response (FIR), infinite impulse response (IIR), or nonlinear filters such as the median filter. Each type has its own merits and shortcomings [31]:

- **FIR filters** have linear phase, or constant group delay, so different frequency components are delayed by the same amount.
- **IIR filters** may only have somewhat constant group delay, but they are easier to implement than FIR filters.
- **Median filters** preserve edges, but their magnitude response is not well defined.

For this work, the median filter was chosen for its ability to retain edges, which is crucial because during a fault the voltage drops nearly instantaneously. A low pass FIR or IIR filter would cause the vertical edge to be distorted into a sloped transition region.

5.2.1 The Median and Pseudomedian Filters

In the median filter, we first define an odd window length of $2L+1$, where L is an integer. Then for an input $x[n]$, the output $y[n]$ is calculated as follows:

$$y[n] = Median(x[n-L], \cdots, x[n+L]) \tag{5.4}$$

The median filter is a very widely used filter for noise reduction. However, it does have some undesirable characteristics, especially when used as a low pass filter. In order to filter out high frequency noise, what we would like is a filter which has a magnitude response near 1 for frequencies below a cutoff, but then drops to 0 quickly and monotonically upon reaching the cutoff, such as the FIR filter in Fig. 5.2 (note the vertical axis is linear, not in dB). However, the median filter has a very slow drop-off, and the amplitude response oscillates between positive and negative values [32], which is also undesirable.

A similar filter called the pseudomedian filter has better frequency response characteristics [32]. Fig. 5.2 shows that the response is very similar to that of an FIR filter, but the pseudomedian filter still retains the edge preserving quality of the median filter. In fact, for the frequencies above 20 Hz in Fig. 5.2, while the FIR filter has a very small ripple, the pseudomedian filter's response is identically zero. The pseudomedian filter of length L is defined as follows [32]:

$$\begin{aligned} y[n] &= \frac{1}{2}A + \frac{1}{2}B \qquad (5.5)\\ A &= max\left\{min\left\{x[n-L], \cdots, x[n]\right\}, \cdots, min\left\{x[n], \cdots, x[n+L]\right\}\right\} \\ B &= min\left\{max\left\{x[n-L], \cdots, x[n]\right\}, \cdots, max\left\{x[n], \cdots, x[n+L]\right\}\right\} \end{aligned}$$

For the beginning and end of the signal, where there are fewer than $2L+1$ samples available, the signal is padded as necessary by the first and last samples of $x[n]$, respectively.

In (5.5), A is the maximum of a sequence of numbers of length $L+1$, where each element in the sequence is itself the minimum of a sequence of consecutive measurements of length $L+1$. B is very similar to A, but is a minimum of maxima. By the min-max theorem, $A < B$, and A and B can be thought of as "smoothed" out lower and upper bounds, respectively.

The reader should note that the "frequency response" of a median or pseudomedian filter is not a well defined concept, as it is for FIR or IIR filters. This is due to linearity. Consider an input signal $x[n]$ which is represented

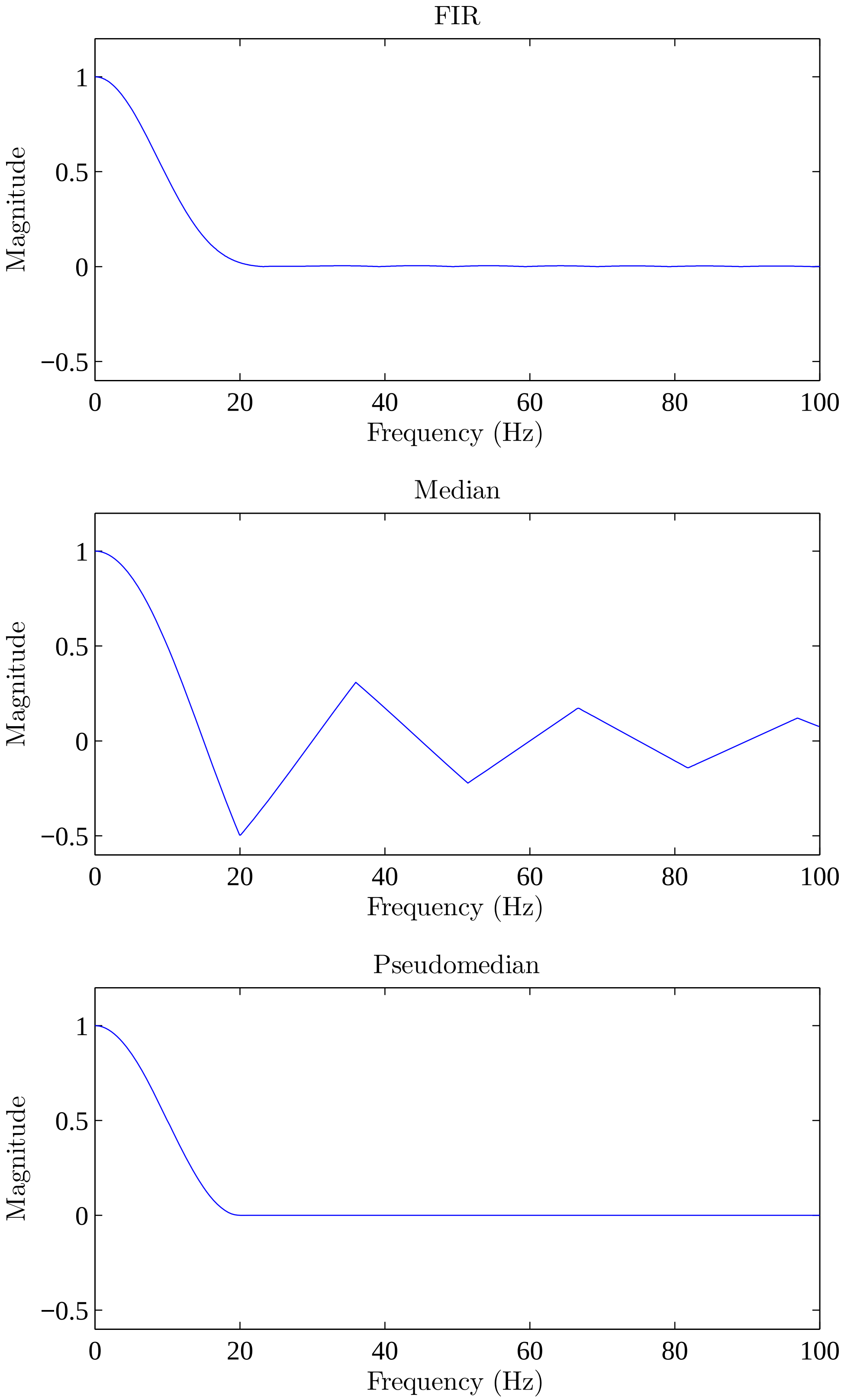

Figure 5.2: A comparison of FIR, median, and pseudomedian filters.

as a sum of sinusoids of different frequencies:

$$x[n] = \sum_{k=1}^{K} a_k \sin(w_k n) \tag{5.6}$$

The output $y[n]$ of a linear filter H_L is the same whether we apply the filter to the signal $x[n]$ as a whole, or to the individual sinusoids and then sum:

$$y[n] = H_L \left(\sum_{k=1}^{K} a_k \sin(w_k n) \right) = \sum_{k=1}^{K} H_L(a_k \sin(w_k n)) \tag{5.7}$$

However, for a non-linear filter H_{NL}, this is not the case:

$$H_{NL} \left(\sum_{k=1}^{K} a_k \sin(w_k n) \right) \neq \sum_{k=1}^{K} H_{NL}(a_k \sin(w_k n)) \tag{5.8}$$

Thus, the frequency response of the median or pseudomedian filter depends on the signal being filtered. This is precisely the reason why the median and pseudomedian filters are able to act as low pass filters, while at the same time preserving edges (which have significant high frequency content) in a signal. To generate the median and pseudomedian plots in Fig. 5.2, we calculate the frequency response of the filter to a sinusoid of a single frequency, and then varied that frequency from 0 Hz to 100 Hz to obtain the plot. Henceforth, we refer to the response of the median and pseudomedian filters as the single sinusoid frequency response (SSFR).

5.2.2 Design of a Pseudomedian Filter

Whereas FIR and IIR filters have standard design methodologies, the pseudomedian filter does not. Because the pseudomedian filter is a non-linear filter, concepts such as cutoff frequency cannot be defined in the traditional sense, since the cutoff frequency is based on the frequency response, which we know from Section 5.2.1 is not constant. Thus, as an approximation, we define the cutoff frequency of a pseudomedian filter based on its SSFR.

To calculate the cutoff frequency of a pseudomedian filter, we first write the closed form equation of its SSFR [32]:

$$H_{PM} = \begin{cases} \frac{1}{2}\left(1 + \cos\left(\pi \frac{f_N}{2}\right)\right) & 0 \leq f \leq 2 \\ 0 & f > 2 \end{cases} \tag{5.9}$$

where f_N is the normalized frequency in cycles/window, and the window length W is defined as $2L + 1$. Noting that $f_N = \frac{Wf}{f_s}$, where f is the frequency in Hz and f_s is the sampling rate, we can rearrange (5.9) to solve for the window length required to produce a desired cutoff frequency:

$$W = \frac{2}{\pi} \frac{f_s}{f_c} \cos^{-1}\left(\frac{2}{\alpha} - 1\right) \tag{5.10}$$

where α is the attenuation at the cutoff frequency f_c—typically $\sqrt{2}$ (3dB). Thus, for a desired cutoff frequency, the length of the pseudomedian filter should be:

$$L = \left\lceil \frac{1}{\pi} \frac{f_s}{f_c} \cos^{-1}\left(\sqrt{2} - 1\right) - \frac{1}{2} \right\rceil \tag{5.11}$$

Here, we also introduce the ceiling function $\lceil \cdot \rceil$, in order to return the closest integer value of L that achieves the desired cutoff frequency or better.

From 5.10 we can also calculate the beginning of the stop band. In this case, we will define the stop band as the frequency where the SSFR is zero (i.e. $\alpha = \infty$):

$$f_{stop} = f_c \frac{\cos^{-1}\left(\frac{2}{\infty} - 1\right)}{\cos^{-1}\left(\frac{2}{\sqrt{2}} - 1\right)} \approx 2.75 f_c \tag{5.12}$$

5.2.3 Applying the Pseudomedian Filter

Fig. 5.3 shows the results of the data from Fig. 5.1 filtered using pseudomedian filters with cutoffs of 480 Hz, 120 Hz, and 30 Hz. The reader can see that a 30 Hz cutoff for this application is too low; the window length is so long that important features are removed. Both the 480 Hz and 120 Hz cutoffs are acceptable, depending on the user's preference. For this work, the 120 Hz cutoff will be used.

5.3 Results

In this section, we apply the two algorithms (CaR and StC) to real measurements. The event we will study is a single line to ground fault on a 345 kV transmission line in the Eastern Interconnection. Further details have been withheld due to confidentiality. The measurements obtained were shown in Fig. 5.1, and then filtered as per Section 5.2.3.

5.3.1 Compare and Resimulate

Fig. 5.4 shows the convergence of the CaR algorithm. For the first part of the analysis, we will use an unconstrained optimization, and not restrict the values of $\boldsymbol{p}^*$ to lie between 0% and 100%, although they must still sum to 100%. The solution required 229 function evaluations, and took 48 minutes, comparable in computation time to the validation case study. However, while

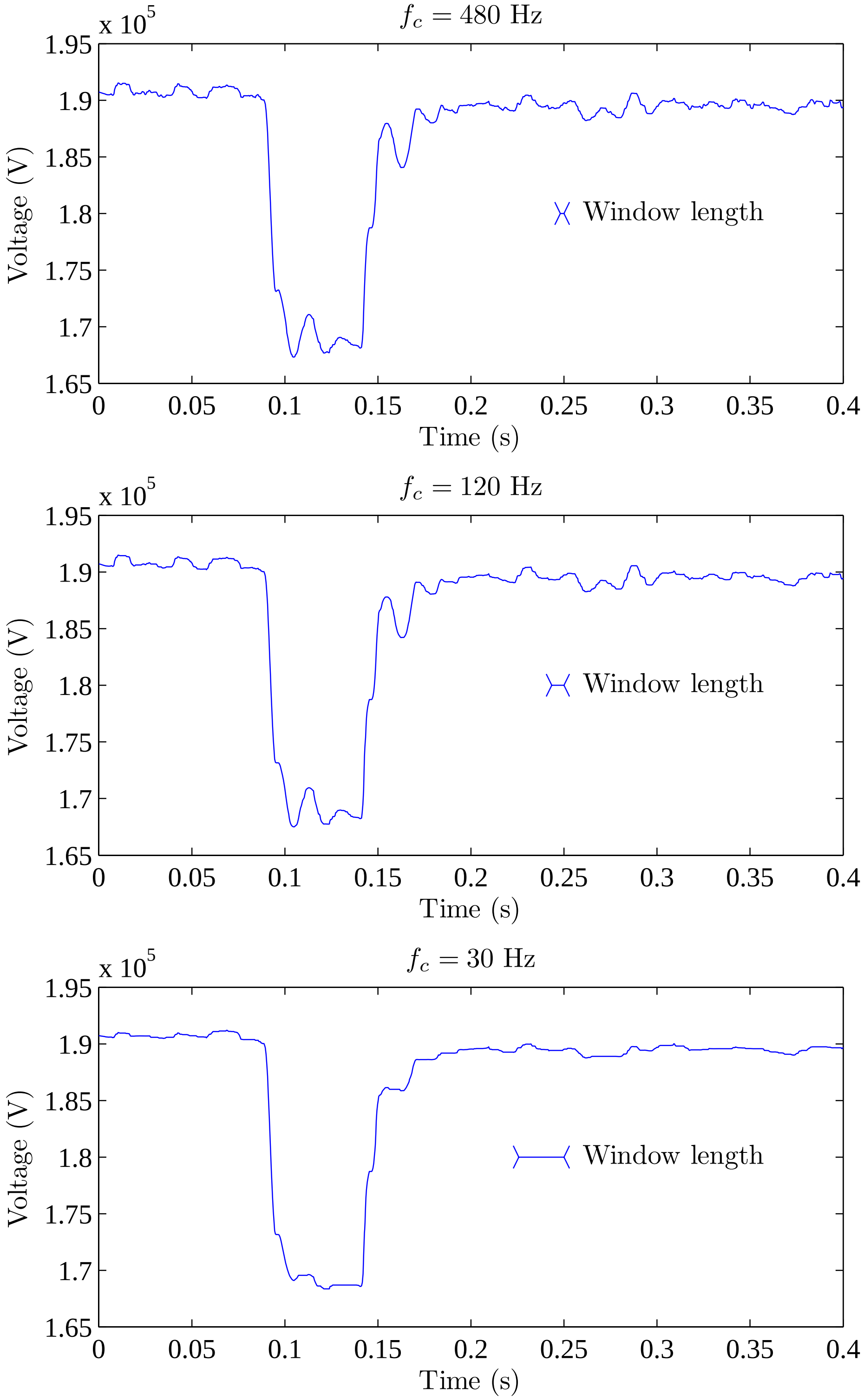

Figure 5.3: The data from Fig. 5.1 filtered using pseudomedian filters with cutoffs of 480 Hz, 120 Hz, and 30 Hz. The length of the window (W) is also shown.

the algorithm does converge, the reader can see from Table 5.1 that the values do indeed not lie between 0% and 100%, and hence do not have any physical meaning.

Table 5.1: Unconstrained CaR solution for a real disturbance.

PARAMETER	CAR SOLUTION
Large motor	4315.4%
Small motor	1092.2%
Discharge lighting	−14083.8%
Constant MVA	9852.2%
PI/QZ	−1075.9%

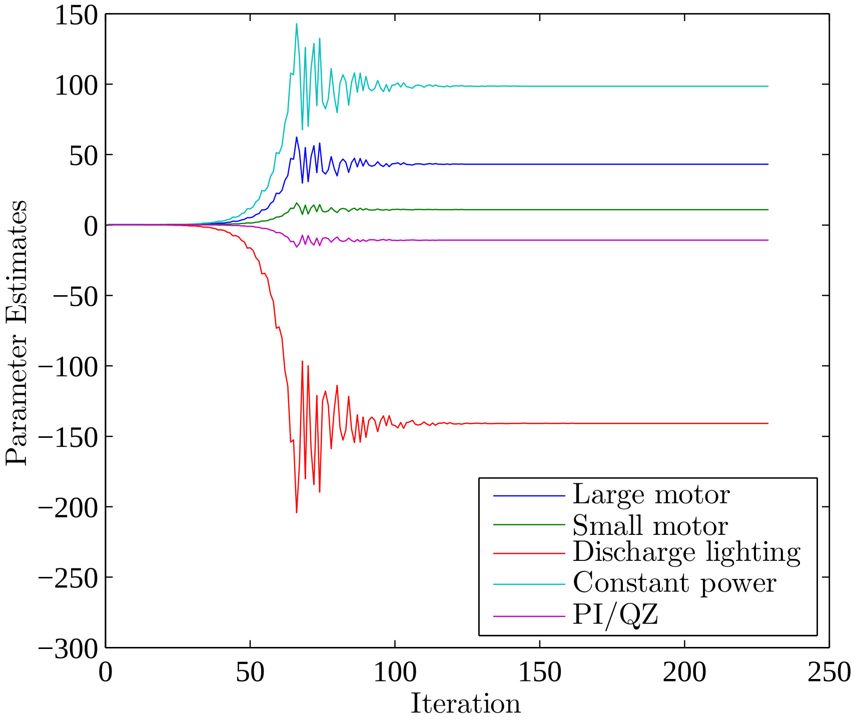

Figure 5.4: Convergence of the CaR algorithm on $\boldsymbol{p}^*$ for a real disturbance.

An absence of physical meaning does not immediately imply that the solution is useless. If we take the approach that we want a load model that responds to this fault as the real system would, then we can view the vector $\boldsymbol{p}^*$ as an abstract model parameter, which does not need to satisfy physical constraints. However, this is predicated on the condition that v^* and v_{meas} match closely. However, Fig. 5.5 shows that the simulated and measured waveforms are far from similar. Specifically, the oscillations observed could not be reproduced during simulation.

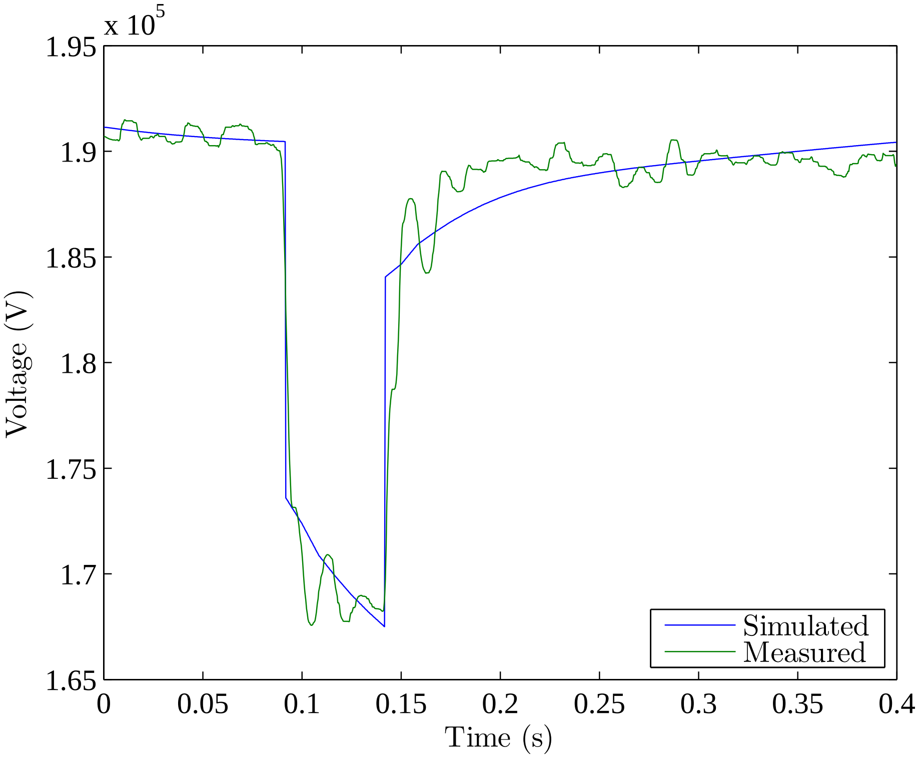

Figure 5.5: $v_{\text{meas}}[n]$ compared to $v^*[n]$, when the unconstrained CaR algorithm is applied to a real disturbance.

If we add in the constraint that $\boldsymbol{p}^*$ must lie between 0% and 100%, the solution is shown in Table 5.2, and the simulated and measured waveforms are shown in Fig. 5.6. The Constant MVA load hits the upper bound, while the other loads are at the lower bound. The waveforms in Fig. 5.6 are even more dissimilar than in the unconstrained case. Thus, we conclude that the CaR method is ill-suited for this purpose.

Table 5.2: Constrained CaR solution for a real disturbance.

PARAMETER	CAR SOLUTION
Large motor	0%
Small motor	0%
Discharge lighting	0%
Constant MVA	100%
PI/QZ	0%

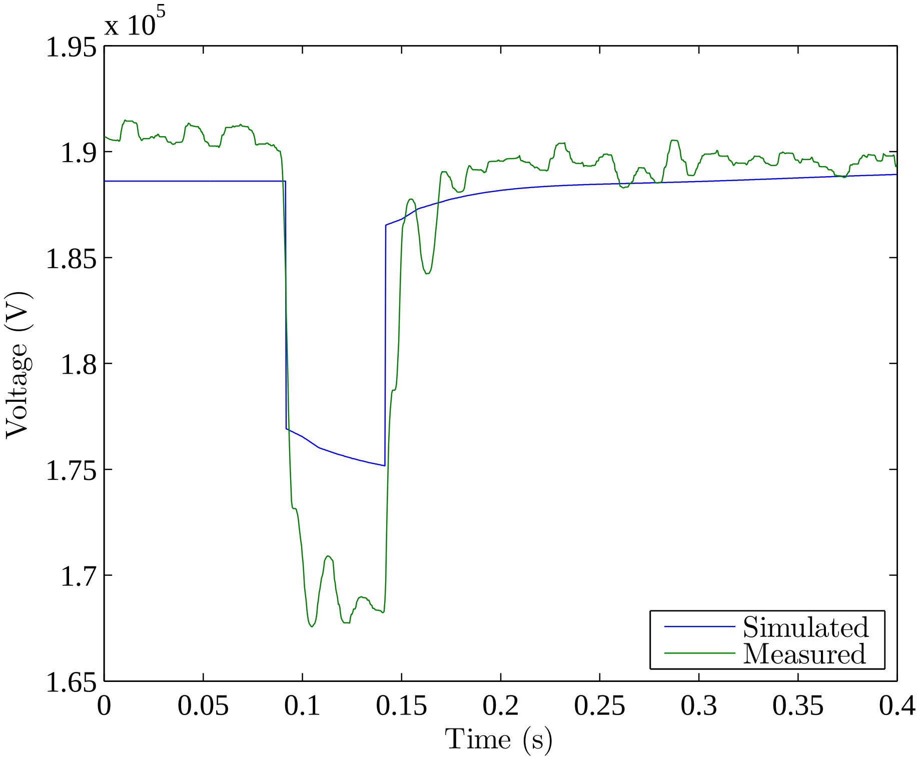

Figure 5.6: $v_{\text{meas}}[n]$ compared to $v^*[n]$, when the constrained CaR algorithm is applied to a real disturbance.

5.3.2 Simulate then Calculate

We can repeat the above analysis for the StC algorithm. 500 simulations were used and the bounds used were the same as in Section 4.2.2: $5\% \leq p(i) \leq 50\%$. The results are shown in Table 5.3, and the simulated and measured waveforms are shown in Fig. 5.7. Again, the results are nonsensical. In fact, the solution values are far outside the bounds on the parameters used for the input simulations. Thus, this solution makes an extreme extrapolation.

Table 5.3: StC solution for a real disturbance, using 500 simulations with a bounded uniform distribution.

Parameter	CaR Solution
Large motor	2.89e11%
Small motor	−5.79e10%
Discharge lighting	1.11e13%
Constant MVA	8.18e12%
PI/QZ	−1.95e13%

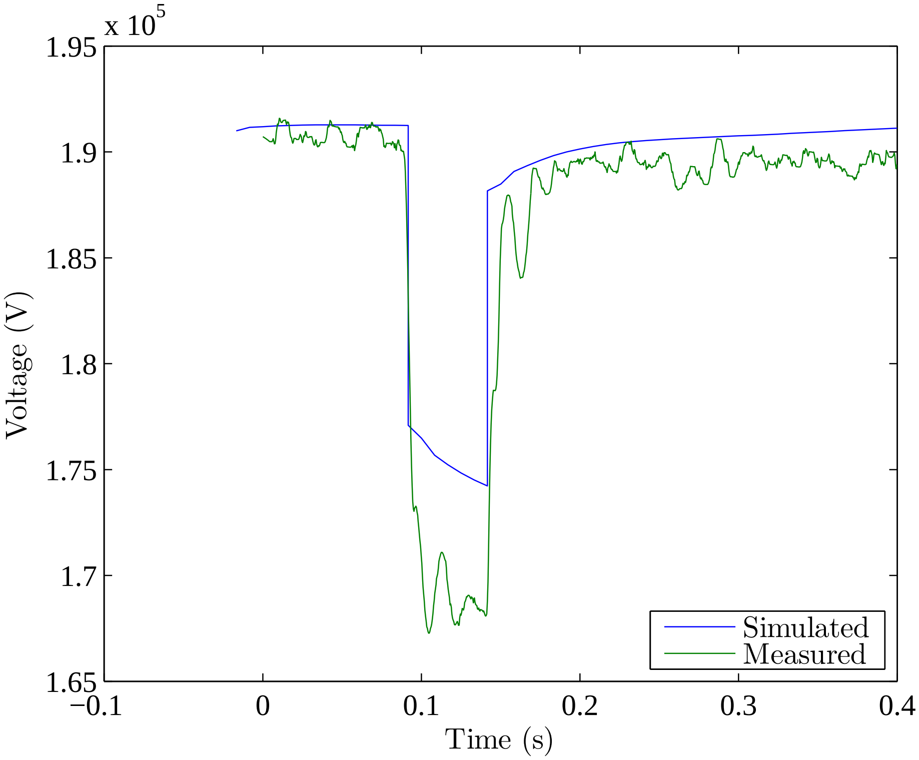

Figure 5.7: $v_{\text{meas}}[n]$ compared to $v^*[n]$, when the StC algorithm is applied to a real disturbance.

5.3.3 Discussion

From the results in this section, we can see that neither algorithm was able to find a reasonable load model, even when we relaxed the constraints. This is not an indication of errors or inadequacies in the algorithms, but instead of the fact that a simulation of the grid during a fault is not accurate enough to permit parameter estimation of a composite load model, due to the sensitivity of $\boldsymbol{p}$ to $v[n]$. In other words, if small changes in $\boldsymbol{p}$ led to small changes in $v[n]$, then the algorithm may be able to tolerate inaccuracies in simulation. However, this is not the case (see Fig. 4.2). The simulation violates the injectivity condition in Section 3.2, and thus any parameter estimation algorithm is almost guaranteed to fail.

CHAPTER 6

CONCLUSION

Simulations are used extensively in power systems, but they rely on accurate models to give us meaningful results. While most other facets of the power system have been studied extensively, there is still no consensus on what type of model to use: the model must strike a balance between simplicity and accuracy. Static load models were developed for steady state analysis, but have recently been used in transient simulations, with some success. However, they cannot match the performance of dynamic models. Dynamic models typically describe the behavior of the load using derivatives, and while some have a very generic structure, others such as the CLOD model are specific to power systems. This thesis has investigated the parameter estimation of the CLOD model, using a hybrid measurement/component based approach.

Two parameter estimation algorithms were developed: CaR and StC. CaR uses a nonlinear optimization routine with a simulation at every iteration to determine the optimal parameters. This algorithm had excellent performance in theory, but failed the validation case study when even a minute amount of noise was added to the measurements. Additionally, we identified that some of the CLOD submodels behave similarly, and thus the algorithm has a difficult time distinguishing between them. The StC algorithm uses least squares fitting, and while it was not as accurate as the CaR algorithm in the validation study, it was more robust to noise. This algorithm performs best when we have a good guess of the parameters ahead of time, since well placed bounds on the simulations were shown to provide better parameter estimates.

In order to make use of real disturbance data, especially the three-phase time domain data from DFRs, we used a DFT to convert it to the phasor domain, and then used the concept of symmetric components to extract the positive sequence data. This data, however, was extremely noisy due to both measurement noise and errors introduced in the post-processing. In order to filter out this noise, we made use of a pseudomedian filter, which has both the edge preserving qualities of a median filter, and the steep roll-off of an FIR or IIR filter.

When we applied the above processing to measurements from a real disturbance, we found that both the CaR and StC algorithms provide solutions far from expected. The reason was that the sensitivity of the parameter estimate to the simulation is too high given the accuracy that the simulation software could achieve. Additionally, the nonlinear nature of the CLOD model exacerbated the problem, as it violated the injectivity condition. Thus, we conclude that measurement-based algorithms are ill-suited for the parameter estimation of complex, nonlinear load models; a higher level of success may be achieved using component-based methods.

REFERENCES

[1] F. J. Meyer and K. Y. Lee, “Improved Dynamic Load Model for Power System Stability Studies,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS 101, no. 9, pp. 3303–3309, 1982.

[2] *PSS/E 33.0 Model Library*, Siemens Inc., 2011.

[3] J. De Leon and B. Kehrli, “The Modeling Requirements for Short-Term Voltage Stability Studies,” *2006 IEEE PES Power Systems Conference and Exposition*, pp. 582–588, 2006.

[4] M. Kent, W. Schmus, F. McCrackin, and L. Wheeler, “Dynamic Modeling of Loads in Stability Studies,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 5, pp. 756–763, May 1969.

[5] R. Bravo, R. Yinger, D. Chassin, H. Huang, N. Lu, I. Hiskens, and G. Venkataramanan, “Load Modeling Transmission Research,” CIEE, Tech. Rep., 2010.

[6] K. Keesman, *System Identification: An Introduction.* Springer, 2011, vol. 2.

[7] U.S. Energy Information Administration, “Frequently Asked Questions,” Online document, 2011. [Online]. Available: http://38.96.246.204/tools/faqs/faq.cfm?id=65\&t=2

[8] U.S. Census Bureau, “State & County QuickFacts,” Online document, 2011. [Online]. Available: http://quickfacts.census.gov/qfd/states/00000.html

[9] R. Bravo, R. Yinger, D. Chassin, H. Huang, N. Lu, I. Hiskens, and G. Venkataramanan, “Load Modeling Transmission Research Appendix H - Load Monitoring,” CIEE, Tech. Rep., 2010.

[10] J. C. Wang and H. D. Chiang, “Development of a frequency-dependent composite load model using the measurement approach,” *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1546–1556, 1994.

[11] I. R. Navarro, "Dynamic Load Models for Power Systems-Estimation of Time-Varying Parameters During Normal Operation," *Distribution*, 2002. [Online]. Available: http://swepub.kb.se/bib/swepub:oai:lup.lub.lu.se:587901?vw=full\&amp;tab2=abs

[12] P. Dandeno, H. Brown, and C. Dube, "System Load Dynamics-Simulation Effects and Determination of Load Constants," *Power Apparatus and Systems, IEEE Transactions on*, 1973.

[13] A. Capasso, W. Grattieri, R. Lamedica, and A. Prudenzi, "A bottom-up approach to residential load modeling," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 957–964, May 1994.

[14] J. Zhao and Y. Wang, "A new continuation power flow model for simulating time-domain dynamic load restoration," *Power & Energy Society General Meeting*, pp. 1–7, 2009.

[15] Y. Li, H. Chiang, B. Choi, Y. Chen, D. Huang, and M. Lauby, "Load models for modeling dynamic behaviors of reactive loads: Evaluation and comparison," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 9, pp. 497–503, Nov. 2008.

[16] H. Renmu, M. Jin, and D. Hill, "Composite Load Modeling via Measurement Approach," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 663–672, May 2006.

[17] S. G. Abhyankar and A. J. Flueck, "Simulating voltage collapse dynamics for power systems with constant power load models," *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–6, July 2008.

[18] B.-K. Choi, H.-D. Chiang, Y. Li, H. Li, Y.-T. Chen, D.-H. Huang, and M. G. Lauby, "Measurement-based dynamic load models: derivation, comparison, and validation," *Power Systems*, vol. 21, no. 3, pp. 1276–1283, 2006.

[19] C. J. Lin, A. Chen, C. Chiou, C. Huang, H. Chiang, J. Wang, and L. Fekih-Ahmed, "Dynamic load models in power systems using the measurement approach," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 309–315, 1993.

[20] B. Lesieutre and P. Sauer, "Development and comparative study of induction machine based dynamic P, Q load models," *Power Systems, IEEE*, vol. 10, no. 1, pp. 182–191, 1995. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=373941

[21] C. Taylor, *Power System Voltage Stability.* New York: McGraw-Hill, 1994.

[22] *PSS/E 31.0 Program Operation Manual Volume II*, Siemens Inc., 2007.

[23] J. Glover, M. Sarma, and T. Overbye, *Power System Analysis and Design*, 4th ed. Thomson Learning, 2008.

[24] MathWorks, "Unconstrained nonlinear optimization algorithms," Online document, 2011. [Online]. Available: http://www.mathworks.com/help/toolbox/optim/ug/brnoxr7-1.htm

[25] T. Overbye, "Load Identification Algorithm," unpublished.

[26] T. Gibson, A. Kulkarni, K. Kleese-Van Dam, and T. Critchlow, "The Feasibility of Moving PMU Data in the Future Power Grid," U.S. Department of Energy, Tech. Rep. PNNL-SA-80268, 2011.

[27] Centerpoint Energy Inc., "Securities and Exchange Commission, Form 8-K," Tech. Rep. 0001130310, 2002. [Online]. Available: http://files.shareholder.com/downloads/HOU/1945249908x0xS950123-10-17554/1130310/filing.pdf

[28] D. Sevcik, R. Lunsford, M. Kezunovic, Z. Galijasevic, S. Banu, and T. Popovic, "Automated analysis of fault records and dissemination of event reports," in *Fault and Disturbance Analysis Conference*, Atlanta, 2000.

[29] P. Smith, "Regional Capacity and New Projects," APS, Tech. Rep., 2008.

[30] P. Sauer, M. Pai, and S. Zhao, "Phasor representations of power system transients," unpublished.

[31] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River: Pearson Higher Education, 2010.

[32] M. A. Schulze, "Mathematical Properties of the Pseudomedian Filter," Ph.D. dissertation, University of Texas at Austin, 1990.