Applications of Time Synchronized Measurements in the Electric Grid

Mohini Bariya

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-196
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-196.html

August 13, 2021

Applications of Time Synchronized Measurements in the Electric Grid

by

Mohini Bariya

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kannan Ramchandran, Co-chair
Adjunct Professor Alexandra von Meier, Co-chair
Associate Professor Prabal Dutta
Professor Scott Moura

Summer 2021

Applications of Time Synchronized Measurements in the Electric Grid

Abstract

Applications of Time Synchronized Measurements in the Electric Grid

by

Mohini Bariya

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kannan Ramchandran, Co-chair

Adjunct Professor Alexandra von Meier, Co-chair

Increased real-time monitoring of the electric grid is vital to meet the burgeoning challenges posed by load growth and diversification, renewable generation integration, extreme weather events, and cyber attacks. Grid operators must have situational awareness—an understanding of the system's evolving state—if they are to respond appropriately to challenging and changing system conditions. The proliferation of measurement devices in the electric grid is critical for situational awareness, but is not sufficient: measurements need to be converted to actionable insight to be useful. Here, computational tools that ingest measurements to infer system parameters and state are critical. While many such tools have been proposed in the research literature, their real-world use is limited, resulting in a circumstance where ballooning volumes of measurements are perceived as overwhelming rather than insightful, diminishing the incentive for further sensor deployments.

This thesis argues for the creation of *usable tools* to bridge the chasm between research and deployment. Usable tools have practically realizable data input requirements and—in their forms and outputs—work in effective collaboration with human users. Such tools are well-suited to the demands of real grids, where data and prior knowledge remain scarce, and where safety critical decisions involve human participants.

The thesis goes on to describe several usable tool algorithms for the use cases of topology estimation and monitoring, event detection, and event classification. Finally, it presents broad principles for the further development of usable tools.

Throughout, the thesis emphasizes how high resolution, time synchronized measurements are particularly enabling for the creation of usable tools.

ACKNOWLEDGEMENTS

Reader: the acknowledgements are the first part you read but the final part I write of this thesis. At this very last step of my PhD, I am filled with gratitude—and a sense of immense good luck—reflecting on the people who helped me get here.

Many thanks to my wonderful advisor, Sascha von Meier, for her insight, good spirits, kindness, and unrelenting support. I have been uplifted after every meeting with her. My co-advisor Kannan Ramchandran was always generous with his time and always engaged and enthusiastic to discuss my work, no matter how distant the application from his everyday preoccupations. I ensnared my committee members Prabal Dutta and Scott Moura before my qualifying exam, and since then I have benefited from their feedback, guidance, and good nature.

Two people set me on the way when I was a young student with no idea of what to do or *why*: Michael Cohen, who introduced me to the electric grid, and Professor Jeffrey Hadler, who will always be my model of humanity, compassion, and grace.

I can confidently assert that I had the best group mates possible. I commend Sascha for collecting them; we are lifelong friends! Immense thanks to Keith Moffat, Jaimie Swartz, Gabriel Colon-Reyes, Kyle Brady, Miles Rusch and Laurel Dunn for joyous times at work and outside. Thanks also to my many wonderful colleagues, collaborators, and comrades: Jonny Lee, Aminy Ostfeld, Liz Ratnam, John Paparrizos, Deep Deka, Sean Murphy and K.S. Sajan. Their insights and assistance were crucial to my learning and this work. I must thank Betsy Mitchell for the happy summer hours working (and carpooling) together. A special thanks to Alex McEachern whose extraordinary perspicacity is matched by his enormous kindness.

Thanks to my dear Berkeley friends, who have stoically tolerated my erratic responsiveness to texts and yet are always available for me: Amit Akula, Christine Vandevoorde and Linus Kipkoech, Nicholas Gan, Kai-Sern Lim, Rebecca Pak, Nick Chang, Bala Kumaravel, Vignesh Subramanian, and Harkiran Kaur Sodhi. I have so many fond memories of days and evenings together, and look forward to our reunions in Berkeley and elsewhere. Particular thanks to the people that became my covid community: Henry Teng and my betas, Varun and Kahaan Maniar. We cooked, imbibed, traveled and, above all, laughed uncontrollably together!

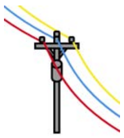
Boundless thanks to my wonderful family, near and far. Thanks to my loving grandparents: Ba —who left us too soon —Nani, Nana, and Dada. Thanks to the Bariya cousins —Amol, Aman, Mohit, and Anooj —with whom were the happiest moments of my childhood. To Monica, Madhavi, Apu, Lalana, and Sonali—how many are fortunate enough to have so many lovely and loving aunts—and my uncles. In Berkeley, thanks to Ven for great times and support, and to Danny for conversation and kindness over dinner and during my unannounced visits. And last thanks to my newest family member: Sabrina.

Perhaps strangely there are places I must thank, that kept me going and gave me meaning. India is always on my mind and in my heart. I will never forget my few days at Merced National Wildlife Refugee, watching at dawn and dusk thousands of sandhill cranes dancing and calling and flying together over the misty lake. My work is for these places.

While writing this thesis, I enjoyed the literary company of Martin Amis in *Inside Story*. Reading his passages made me mindful of my own. His advice —on picking the right word, avoiding repetition, eschewing the canned phrase and being conscious of the cadence of paragraphs —I have tried, though not fully managed, to follow. Nevertheless, I, and my reader, owe him thanks.

Finally, and most significantly: thanks to my *jigri doston*, my soul friends, Eugene Kang and Sourav Ghosh. I can not express how much happiness and joy you have brought me. I look forward to growing old together, sitting on tropical verandahs, sipping g&ts and listening to music (k-pop and qawwali) over endless conversations and laughter.

And above all and always, thanks to the *humsafar*s, the travel companions, of my life: Ilias, my sister Mallika, and my parents Anand and Suhag, who know me better than I know myself. This thesis is dedicated to them.



CONTENTS

1 INTRODUCTION 1
  1.1 The Electric Grid 1
  1.2 Motivation 4
  1.3 Contribution 7
  1.4 Notation 8
2 FOUNDATIONS 10
  2.1 Grid Measurements 11
  2.2 PMU Noise 18
  2.3 Measurement Platform 29
  2.4 Conclusion 34
3 WHAT ARE USABLE TOOLS? 35
  3.1 Tools in the Literature 40
  3.2 Tools in Industry 46
  3.3 The Gap 49
  3.4 Bridging the Gap: Usable Tools 51
  3.5 The Future Toolbox 56
4 TOPOLOGY TOOLS 60
  4.1 Towards usable tools for topology 63
  4.2 A Heuristic Approach 65
  4.3 A Physics Approach 77
  4.4 Extension to Three Phase Networks 104
  4.5 Justified Heuristics 124
5 EVENT DETECTION & CLASSIFICATION TOOLS 138
  5.1 Bayesian Event Detection 142
  5.2 Streaming Event Clustering 150
6 PRINCIPLES FOR USABLE TOOLS 164
  6.1 Linear Models 164
  6.2 Statistical Assumptions 167
  6.3 Statistical Baselines 169
  6.4 A Vision for Tools 170
BIBLIOGRAPHY 171

1 INTRODUCTION

1.1 THE ELECTRIC GRID

Electric grids deliver energy in the form of electricity from generators to consumers. Beside this unifying feature, electric grids are diverse, and any general description will likely be violated by some grid, somewhere. In size, grids range from gigantic to small. The contiguous European grid spans east to west from Portugal to Turkey and north to south from Algeria to the Netherlands. At the opposite scale, a grid may encompass just a single generator and a few small loads, supplying energy to an industrial park or to an isolated village. Most grids, and certainly all substantial ones, deal in alternating current (ac) power, with voltages and currents that vary sinusoidally in time. Yet, an increasing number of grids are direct current (dc) systems, with constant steady state voltages and currents. A mixture of ac and dc elements is also possible. Some grids may rely on a single generation source or even a single generator, but most large grids include varied generator types. The Brazilian grid incorporates enormous hydropower plants alongside gas, nuclear, and oil generators while the European grid encompasses Spanish solar farms, French nuclear plants and German offshore wind turbines. Energy user, or load, types are even more diverse than generators, ranging from households with lighting and appliances to energy-hungry factories and, today, giant data centers. In between generators and loads, a large grid includes thousands of pieces of equipment for smooth and safe operations. Lines carry current, transformers change voltage levels, switches reconfigure connections while breakers open them in response to unsafe conditions, and all the while meters measure customer consumption. Consequently, the grid and those who work on it are the elephant and the blind men in the famous parable: each participant focuses on a particular aspect of the system from the perspective of their particular priorities. A grid planner considers financing and infrastructure on time scales of years or decades, an operator thinks of markets for balanc-

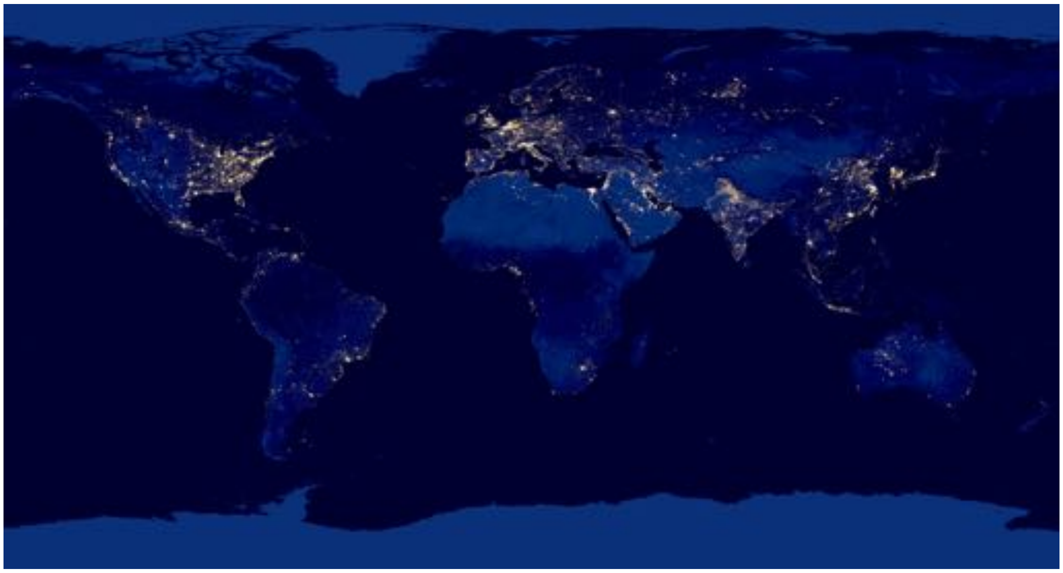


Figure 1: Satellite imagery of the Earth at night captures much about the electric grid: its scale, extent, and inequities but most vividly its fundamental importance to modern life [1].

ing demand and supply every hour, a protection engineer is preoccupied with the coordination of fuses on a particular circuit, an analyst considers the geopolitical implications of an international grid interconnection, and a politician worries about election impacts of expanding infrastructure to underserved constituents.

In a contiguous ac grid these disparate elements are bound by the single frequency of sinusoidal voltages and currents across the system. This frequency originates in the grid's heart(s)—the massive rotating turbines of generators—and is a pulse measurable everywhere, leading such grids to be termed *synchronous*. For this reason, despite the numerous elements constituting the grid, we can believably describe it as a single machine: "the world's largest machine". Historically, beneficial economies of scale have driven synchronous grids to be larger and larger, aggregating customers to be supplied by enormous, centralized power plants. From these plants, energy flows unidirectionally outward through the system to consumers, whose demand is considered inflexible, and in aggregate follows consistent patterns. This operational regularity, and the physics of synchronous generators have given synchronous grids surprising stability and robustness despite minimal sensing and meager temporal and spatial flexibility or visibility into the system state. Built out incrementally over decades into enormous, complex infrastructures, synchronous grids have worked under little real time management or even understanding, conveyed in the pithy maxim: “the electric grid works in practice but not in theory".

Synchronous grids are organized and operated in two, mostly silo-ed parts: transmission and distribution. Transmission networks carry bulk power—at high voltages to minimize losses—over the long distances from centralized generators to loads. Transmission infrastructure is large, with tall towers and hefty lines. This is the most visible portion of the electric grid both physically and operationally. Who can fail to notice transmission lines traversing the landscape between imposing towers? For operators, relatively high sensor deployments and network information provide visibility into the operating state of transmission. This enables a degree of control and optimization in transmission: for example, using network connectivity and impedances, optimal power flows can be solved to set generator outputs that minimize losses or cost. Nevertheless, even in transmission, real time visibility is surprisingly low, as illustrated by two examples. First, in many transmission networks, ac voltages at nodes are not directly measured but must be estimated through a process called state estimation, with non-negligible error. Second, the loading limits of transmission lines—which constrain the quantities of power they can transfer—change dynamically with weather conditions, but are usually known only statically, leading to overly conservative line usage.

Distribution is isolated from transmission through transformers, which reduce transmission level voltages to the lower and safer ones at which power is delivered to most loads. In total distance covered, distribution networks constitute the bulk of the electric grid. Yet it is easier to overlook the infrastructure of distribution which is smaller and sometimes even underground. Analogously, operational visibility into distribution is rudimentary, and control generally consists of automated and invariable protection devices, set to respond only to extreme events. Often, even basic parameters of distribution networks—such as their connectivity structures—are erroneous or not known. Problematically, distribution is also the more vulnerable part of the grid: highly exposed to tree falls, animal contact, storms and even unwitting or malicious human damage.

What grid management does occur takes place largely in control rooms: themselves separated into transmission and distribution. Here human grid operators monitor the system, much like in air traffic control, hands ready at phones to communicate with field workers or respond to customer reported outages. They are surrounded by system information, previously in the form of paper reports updated manually after field visits and increasingly as screens displaying measurements streaming in from field devices. The grid is so enormous, that even with paltry system visibility, control rooms are sites of information overload. Those who work with grid data know the difficulty of interpreting measurements from just a single grid sensor. In contrast, operators must synthesize multiple sources and types of information to inform critical decisions, with little automated assistance. This is already an onerous task today, though current levels of grid data and measurement fall far short of those required for complete system visibility.

Despite the existing challenges spanning transmission, distribution, and control rooms, the electric grid in the developed world was long considered a "solved problem". Grid reliability is high: for example, on average in the United States in 2013, customers endured less than three hours of interruptions per year [2]. Customers in western Europe fared even better [3]. How is this possible? As mentioned, synchronous generators and the historic regularity of loads contribute to system stability. In addition, grid infrastructure in the developed world has been oversized and overbuilt, allowing operators to abide by extra cautious system limits. The result is a reliable yet inflexible and (cumulatively) costly system which strays little from the expected operating state and therefore requires minimal real time visibility. Reliability is further enhanced through expensive infrastructure choices. In Germany, for example, extensive undergrounding of lines reduces distribution grid vulnerability and outages enormously, albeit at high cost [4].

In the years ahead, this legacy operational model for electric grids will be highly challenged and even infeasible in many contexts. In light of daunting new demands and difficulties facing the system, a new approach to grid management is needed, as are novel tools to enable it.

1.2 MOTIVATION

The looming challenges facing electric grids arise from a conjunction of new trends and old approaches. The legacy approach to running grids—

as rigid, opaque, oversized systems—-is unviable as loads and generation grow and diversify, and new security concerns emerge.

The diversification of loads and generation is due in part to transformations external to the energy sector: an example is the growth of electricity usage by computers in data centers. However, the primary driver of diversification is the accelerating global effort to decarbonize our energy systems in order to mitigate the devastating impacts of climate change. Electrifying technologies and processes that have historically depended on fossil fuels is an important first step toward curtailing carbon emissions. Currently, transportation accounts for 14% of global CO_2 emissions [5], motivating countries across the developed and developing world to set ambitious electric vehicle (EV) adoption targets [6]. Industrial production is another major carbon emitter, and the electrification of industry, though especially technically challenging, is recognized as essential for meeting global climate targets [7]. Consequently, electrified industrial plants and vehicles are increasingly being plugged into the grid. The integration of these new load types strains the electric grid in diverse ways. Novel loads may introduce unprecedented dynamics and usage trends, challenging the expertise of grid operators and engineers as they plan and run the system. For example, the operation of industrial equipment is often the cause of troublesome disturbances in transmission networks; these will only increase as more industrial plants are electrified. Another example: research suggests that long used, well established load models—vital components of power system planning and operation studies—do not adequately capture the behaviors of novel loads [8]. The location of novel loads presents additional challenges. Many novel loads will be connected to the grid through the distribution network, the most opaque portion of the system that is neither designed nor managed to handle the behaviors of such loads. Distribution power quality issues—permissibly overlooked in the past—may have serious consequences on novel loads, which could in turn amplify these local problems into system-wide ones. For example, short, significant dips in network voltage—termed "voltage sags"—are inconsequential to lightbulbs or ovens but may cause EV chargers to trip offline [9]. The coincident tripping of numbers of EV chargers can exacerbate distribution voltage problems and even destabilize the broader grid.

On the generation side, decarbonization efforts are propelling the integration of renewable energy in the grid. Renewable generators encompass a range of energy sources—wind, water, sunlight—and technologies. They differ widely in scale and character. Hydro generation is usually large scale in the form of dams, while wind and especially solar can range in size from individual household installations to farms covering many square kilometers. From the grid perspective, the challenge of integrating renewable sources has spatial and temporal dimensions. Renewable availability is spatially varied and inflexible. Places with high renewable generation potential tend to be distant from locations of significant energy demand. To utilize this potential, grid infrastructure must carry more power over longer distances. Another spatial problem arises from where renewable generation interconnects to the grid. Power generated by rooftop solar panels enters the grid in the distribution network—a system which was not designed to expect the resulting, sometimes bidirectional, power flows. Meanwhile, the temporal challenge of renewables is that their outputs, unlike those of fossil fuel power plants, vary in uncontrollable and largely unpredictable ways. In power systems—which lack significant storage yet must precisely balance generation and load—this unpredictable temporal variability is a grave challenge. Similar to novel load types, renewable generation technologies have behaviors unforeseen by grid designers which can challenge conventional approaches to safety and stability. A compelling example is the Blue Cut fire incident in California, where a fault event caused 1.2 GW of rooftop solar to suddenly trip offline, resulting in a large system disturbance [10]. The trip was incited by the controllers of the inverters which interface between rooftop solar systems and the bulk grid, and whose rapid control schemes are generally unknown to and unaccounted for by grid operators. At a wider scale, the displacement of generators with enormous turbines by power electronic based renewable technologies leads to less of the *inertia* that allows the grid to ride through small discrepancies in generation and demand and to maintain a system frequency that deviates little from nominal.

Weather poses another challenge. The grid, like us, is vulnerable to the impacts of climate change. More extreme, more volatile weather—in the form of wildfires, floods, and storms—damages grid infrastructure, leading to more frequent, longer outages. In turn, grid responses to extreme weather can exacerbate the human toll of these events. Some examples from the United States highlighting the deleterious interplay between extreme weather and the electric grid are the infrastructure damage and long-duration power outages experienced by Puerto Rico in the wake of Hurricane Maria, the California wildfires sparked by grid failures during extreme heat and ensuing public safety power shutoffs, and the freakish cold weather that forced power plants offline resulting in blackouts and surging prices in Texas [11]–[13].

A final, emerging concern for electric grids is the threat of malicious attacks by entities seeking to disable this vital infrastructure. The proliferation of networked devices and automation in electric grids can increase their vulnerability to cyber attacks. Given their importance to most aspects of modern life, electric grids are relatively unprotected from attackers. Grid cyber attacks have already occurred: a 2015 attack in Ukraine led to loss of power for over $200,000$ people [14]. The risk of such attacks will only grow in the years ahead.

1.3 CONTRIBUTION

The first, critical step to address the increased threats, faster changing conditions, and greater complexity emerging in all parts of the electric grid, is improved real-time visibility across the system. This is especially necessary in distribution networks, which are traditionally managed as passive and therefore opaque systems. Before grid engineers and operators can begin to concern themselves with how to respond to challenging conditions, they must be aware of the conditions, and of the broader system context. They must have what is termed *situational awareness*: a sufficient understanding of the system's status to inform an appropriate response. However, obtaining situational awareness is complicated by two realities: the lack of comprehensive data availability and the challenge for a human to understand even the limited data coming from such a massive and complex system.

This thesis argues for the creation of computational *tools* that provide awareness of important system parameters and occurrences to human users. The tools must derive operational insight from measurements, especially by leveraging the beneficial traits of high-resolution, time synchronized grid measurements. The tools must transform overwhelming volumes of data to a scale and form suitable for human comprehension. Grid monitoring tools have been developed in the past for various use cases, but most have failed to percolate from research to application. This thesis argues that for tools to be operationally useful, they must meet the criteria of *usability*, encompassing practical deployment requirements and human interpretability.

The thesis then presents work on usable tool development for several use cases.
The rest of the thesis is organized as follows.

- Chapter 2 presents the technical foundations for the creation of grid monitoring tools. These foundations consist of high-resolution, time synchronized measurements from novel grid sensors and the computational platforms that enable the performant storage and access of this data.
- Chapter 3 surveys the literature of grid monitoring tools to understand broad, common features and postulate why most tools fail to transfer to application. Goaded by this survey, the chapter then proposes a succinct set of criteria for *usable tools*, arguing that tools which meet these criteria are well-suited to the needs of real-world deployment.
- Chapter 4 presents several tools for the use case of topology monitoring, highlighting how the tools represent a progress toward increasing usability.
- Chapter 5 presents usable tools for event detection and classification.
- Chapter 6 draws together the lessons from prior chapters to present fundamental principles and strategies for designing usable grid monitoring tools. The chapter emphasizes how high-resolution, time synchronized measurements are especially empowering for the creation of usable tools.

1.4 NOTATION

To the best of our ability, we try to maintain consistent notation throughout this work. Scalar quantities are denoted in lowercase, while vector and matrix quantities are denoted in uppercase. Boldface indicates complex valued quantities, while real quantities are non-bold. Uppercase, calligraphic letters, such as $\mathcal{N}$, denote sets. We use j for the imaginary unit to avoid confusion with electric current: $j = \sqrt{-1}$. In the electric grid, quantities are often expressed in *per-unit* (pu), in which the raw value is standardized by nominal level. Therefore, a voltage of 1200 V on a line with a nominal voltage of 1 kV is expressed as 1.2 pu. There are a smattering of per-unit values in this thesis.

The table below contains a basic demonstration of the notation used. Rest assured that notation will be defined and reiterated in each section.

x	real-valued scalar
$\mathbf{x}$	complex-valued scalar
X	real-valued matrix
$\mathbf{X}$	complex-valued matrix
$\mathbf{X}_{ij}$	i,j element of matrix $\mathbf{X}$

2 FOUNDATIONS

Includes work from [15]–[17]

Novel tools for grid monitoring and management are built on a foundation of grid measurements and the computing platforms that hold them. The features of both the measurements themselves and the platforms they are stored in are instrumental in enabling *and* precluding the types, forms, and scope of tools. Both must be carefully considered and leveraged in tool design.

This chapter summarizes grid measurement types, focusing on the features of measurements that enable the design and development of particular tools. I especially emphasize phasor measurements—made by eponymous phasor measurement unit (PMU) sensors—as they are the inputs to the tools described in following chapters. I highlight the arguably serendipitous features of phasor measurements that make them extraordinarily enabling for novel tool design.

Then, I describe the high performance Berkeley Tree Database (BTrDB). Broadly useful for storing and accessing time series data, BTrDB was originally designed with PMU measurements in mind. The efficacy of several tools in later chapters relies on the attributes of BTrDB. The relevant features of BTrDB are presented through a simple but exemplary electric grid use case.

2.1 GRID MEASUREMENTS

Though the electric grid, especially at its periphery, is a relatively opaque system, it is not totally bereft of sensors and measurements[1]. With burgeoning demands and challenges on electric grids, sensor deployments and measurement volumes have been increasing. There has been a parallel advancement in measurement types, and consequently in the types of grid analyses and visibility that are possible.

In early electric grids, the quantities necessary for daily planning and operation were often estimated rather than measured. The primary requisite quantity was electric demand. In Edison's day, the only load was electric lighting, so utilities could crudely forecast demand by counting the number of light bulbs in each customer's home and assuming they would all be switched on after dark. As electric appliances diversified, load could not simply be enumerated, and more sophisticated forecasts, correlated with weather and other factors, were developed. Nevertheless these were still computed manually, using tables of data, and presumably with significant error margins [18]. In real time, the precise balance of demand and generation necessary for grid stability was achieved through automated generator controls. These responded to changes in the ac frequency of the grid: a direct proxy of system-wide power balance. Frequency could be measured locally at each generator by looking at the rotation rate of the generator turbine [19]. Consequently, with sufficiently large turbines (or equivalently sufficiently small load changes), power balance and grid stability could be maintained without networked measurements and communications. This wonderful feature of the physics of ac electric networks allowed the electric grid to predate computing.

Mathematical advances led to the introduction of more sophisticated grid management techniques. Economic dispatch and later optimal power flow allowed power output to be allocated across generators to meet some objective: generally cost minimization [20]. Astoundingly, these computations were also done by hand, sometimes taking hours to complete. Correspondingly, the data needed for these analyses was far from real time: it was produced through forecasts (in the case of load) or static estimates of sys-

---

[1] This section outlines the transformation of measurement and communication in the electric grid over the last century. While the narrative captures technology development, it does not, and can not, provide a unified global account of technology deployment, which differs enormously from grid to grid.

tem parameters (in the case of connectivity and impedances). Together, data and computation limitations constrained the analyses to be carried out on slow time scales, rather than in rapid response to changing conditions.

Grid sensors are as old as the grid itself. In the early days, a variety of analog sensors were developed for grid applications, including a menagerie of electric meters. Edison developed an early electric meter which intriguingly didn't measure energy but ampere-hours (it was rather inaccurate and upset customers) [21]. With only rudimentary communication technologies, analog grid sensors measurements had restricted, local availability. These sensors were either directly connected to mechanical equipment which automatically responded to their measurements—as in the case of some controllers—or were accessed manually and occasionally—as in the case of electric meters which had to be regularly visited by a meter reader. Some sensors were connected over short distances to analog controllers in the earliest "control rooms" [22]. Most early sensors lacked significant recording capacity, and there was no reporting of time series data. Without computers to process it, there was little use for such data anyway.

Digitization transformed the types and capabilities of grid sensors. As digital sensors and communication infrastructure were deployed—enabling measurements from a wide area to be aggregated at a single location—finally a degree of relatively real time, expansive visibility was achieved. Transmission visibility was (and largely remains) the priority. In the United States, large blackouts spurred the need to not only monitor power at each generator, but to monitor voltages and currents throughout the system. Thus state estimation (SE) was born [23]. Early SE used power measurements from remote terminal units (RTUs) deployed at substations throughout the network and collected by the supervisory control and data acquisition (SCADA) system [24]. These measurements were made and reported at low time resolutions on the order of several seconds or minutes and were not time synchronized. Grid state, consisting of voltage magnitudes and angles at each node (from which current flows on each line could be inferred given impedance values), was not directly measured but inferred from this data through nonlinear SE methods.

Let us briefly pause to consider the implications of the low resolution, non-synchronized data delivered by SCADA. A typical SCADA system reports measurements every two to fifteen seconds [25]. In between, there is no visibility into the system, which is troubling given that we know many much faster processes occur in the grid. Even when SCADA reports, the data is not time synchronized, meaning, for example, that two power values reported simultaneously from two locations may not actually be coincident. Unsurprisingly, this data must therefore be used with care to get even a low time resolution snapshot of the system state. Nevertheless, SCADA measurements were a significant step toward increased spatial and temporal granularity in grid visibility.

Digital technologies permeated distribution networks as well. Most visible and well known are smart meters, which record and report customer consumption at intervals of several minutes. Smart meter deployments were expensive and often contentious [26], but the meters ended the need for manual meter visits every billing cycle, saving utilities enormously [27]. Smart meter types are varied: simpler ones may suffer from time synchronization issues [28], while more sophisticated models have a range of capabilities for control and measurement that are not yet widely used [29].

Less conspicuous are the often advanced digital sensors integrated into distribution network equipment. For example, digital fault recorders can make high time resolution measurements, but only do so over short periods following the detection of a fault event in the system [30]. This data must often be manually retrieved from the recorder. Similarly, many relays make extremely high resolution measurements to trigger protection equipment, but generally do not save or communicate this data [31].

Overall, roll out of communication networks lags behind digital sensor deployment in most electric grids, especially in distribution systems. Therefore—as with relays or digital fault recorders—many devices capable of making sophisticated measurements do not communicate them. Devices that are networked and do report data regularly, such as smart meters, often have low sampling frequencies or time synchronization issues, stymieing efforts to obtain real-time, wide area visibility through their data.

Phasor Measurement Units (PMUs)—boasting high measurement resolutions and accurate time synchronization—promise to mitigate these issues. They embody a nascent vision for a data rich, comprehensively monitored electric grid (popularly called the "smart grid"). For this reason, and because they are the inspiration and data source for much of the work in future chapters, the next sections are devoted to a brief but thorough introduction to PMU data, starting with a description of the quantity reported by PMUs: phasors.

Interlude: What are phasors?

In ac electrical networks, voltages and currents alternate: continually changing in size and direction. Under idealized steady-state conditions, they are perfect sinusoids, oscillating with a fixed amplitude, frequency, and phase shift. In reality, they are imperfect and time varying, often with multiple frequency components. Nevertheless, the idealized model generally holds well, and ac voltages and currents are conventionally represented as perfect sinusoids at the fundamental, or system, frequency.

$$v(t) = v\cos(2\pi f t + \theta) \text{ where } f = 50 \text{ or } 60 \text{ Hz} \tag{1}$$

Since frequency is assumed fixed, the explicit time dimension can be discarded altogether, resulting in a compressed "phasor" representation which still captures everything there is to know about the original, two variable sinusoid: the root mean square (rms) magnitude, and the phase angle shift, or exact timing of the zero crossing:

$$\mathbf{v} = \frac{v}{\sqrt{2}}\angle\theta = \frac{v}{\sqrt{2}}e^{j\theta} = \frac{v}{\sqrt{2}}(\cos\theta + j\sin\theta) \tag{2}$$

Notice the three equivalent denotations of the phasor representation of time domain voltage $v(t)$. The first explicitly indicates the two free parameters of the ac voltage signal: amplitude and phase. The second and third represent the phasor as a complex number and are equated by Euler's formula. The magnitude and angle of the complex number respectively capture the amplitude and phase of the time domain quantity. The $\sqrt{2}$s in (2) arise from the convention of using the root mean square rather than the amplitude of the sinusoid in the phasor notation.

There are numerous methods, such as [32], [33], for estimating the magnitude and angle parameters—v and θ—from real-world, time domain voltage measurements. It is likely that different software and sensors have differing, and often proprietary, estimation algorithms. A unifying and convenient conceptualization, elucidated by Kirkham in several works [34]–[36], is to think of phasor "measurement" as a mathematical fitting process: we are trying to fit the idealized model of (1) to a real-world signal which will differ from it to varying extents (Fig. 2). In this framework, different phasor estimation algorithms fit the parameters of the phasor model by different techniques in order to optimize different objectives. This conceptualization—of the phasor as a model rather than a true

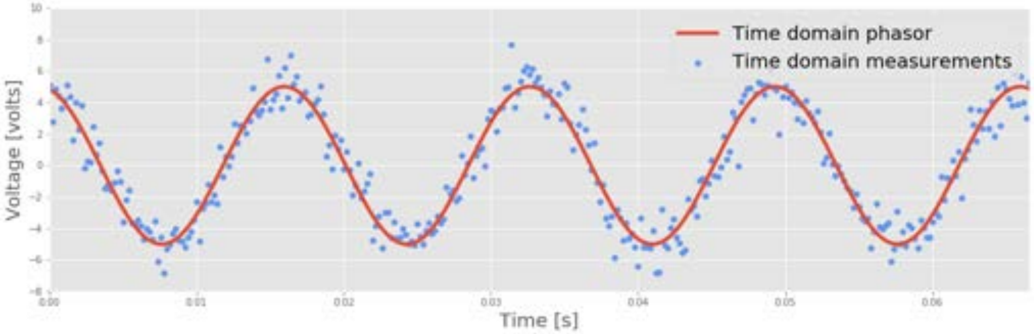


Figure 2: A fitted phasor model visualized in the time domain overlaid on raw measurements of the ac voltage waveform. Notice the discrepancy between model and measurements.

measurement—also illuminates the hazards of phasor data. What should v and θ be when the voltage measurements differ considerably from (1)? How should we use and interpret these values? I do not attempt to (and perhaps it is impossible) definitively answer these questions here. However, when working with phasor data, it is important to keep these quandaries in mind, remembering that phasor measurements are really model parameters, and only as good as the model itself.

Modeling of ac electrical networks is greatly simplified in the phasor domain. The phasor version of Ohm's law has the familiar, linear dc form, but now all parameters are complex valued, capturing the phasor voltages

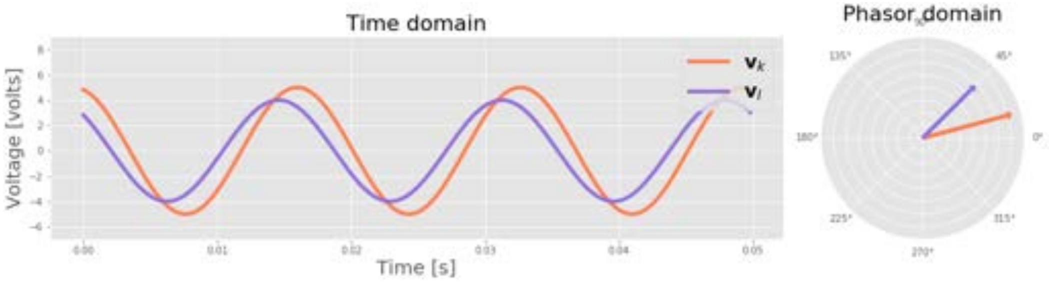
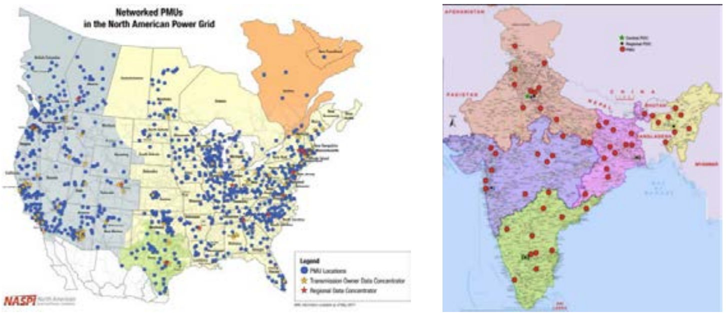


Figure 3: Voltages at two ends of a line visualized in the time and phasor domains. Angle and magnitude differences have been unrealistically exaggerated for illustration.



(a) United States **(b)** India

Figure 4: Transmission PMU deployments in the United States and India. Maps from (a) [37] and (b) [38].

and currents and the resistance and reactance of an electric line. The ac voltage-current relationship across a line is described by:

$$\begin{aligned} \mathbf{v}_k - \mathbf{v}_l &= \mathbf{z}_{kl}\mathbf{i}_{kl} \\ \mathbf{z}_{kl} &= r_{kl} + jx_{kl} \end{aligned}$$

where $\mathbf{z}_{kl}$ is the complex-valued impedance of the line connecting points k and l. We will come back to versions of this equation in later chapters. While ac quantities are visualized in the time domain as oscillating signals, phasors are represented as vectors in the complex plane. These two representations are applied to the voltages $\mathbf{v}_k$ and $\mathbf{v}_l$ and visualized in Fig. 3. Notice how the angle and magnitude differences between the $\mathbf{v}_k$ and $\mathbf{v}_l$ phasors manifest as time delays and amplitude differences respectively in the time domain.

2.1.1 Phasor Measurement Units (PMUs)

Phasor Measurement Units (PMUs) are devices that report the phasor parameters of ac quantities. To do so, PMUs take measurements of ac waveforms—termed point on wave (pow) measurements—at high frequencies up to 1 MHz [25]. A computation then determines the phasor magnitude and angle corresponding to the pow data. It is impossible to obtain a meaningful phasor from a single pow measurement so a phasor is estimated from a window of pow measurements. Generally, PMUs report phasors at lower frequencies than those at which they sample the ac waveforms, and

Figure 5: The μPMU sensor holds the promise of increased distribution network visibility. The white "mushroom" on top of the device box is an antenna for receiving the GPS clock signal.

in this light, PMUs can be thought of as performing a filtering and compression operation on raw pow data. PMUs use a GPS clock to precisely synchronize the sampling of ac waveforms across devices, enabling accurate time alignment of the final reported phasors.

PMU development and deployment was spurred by the urgent need for increased, higher quality visibility into transmission grid state. PMUs directly report the phasors that constitute the grid state, simplifying state estimation to a linear problem. PMUs' accurate time synchronization and considerably higher reporting rates enable data from multiple devices to be collated to obtain expansive snapshots of the system at high time resolution [39]. These benefits have led to extensive PMU deployments on transmission networks globally, for example in the United States and India (Fig. 4), enabling greater transmission visibility. PMUs scattered over large areas in transmission have especially been used to monitor wide area quantities—such as grid frequency and voltage angle differences—for improved understanding of system wide events in projects such as FNET [40].

In 2014, a pilot project sought to extend the visibility enabled by transmission PMUs into distribution networks [41]. Compared to transmission, distribution is characterized by shorter, lower impedance lines and smaller power flows, resulting in angle differences between voltage phasors that are up to two orders of magnitude smaller. In this context, a specialized distribution PMU, termed the micro-PMU (μPMU), with high angular and magnitude resolution, was created (Fig. 5). Sampling the underlying ac waveforms at 512 Hz, μPMUs report phasor quantities at 120 Hz and have reliably discerned angle differences up to 0.01 deg and voltage magnitudes to within 10^{-4} per-unit [42]. As part of the pilot project, a number of μPMUs were deployed on operational distribution networks across the United States. As soon as these μPMUs came online, it became clear that the data they were reporting was extraordinarily rich. This was bolstered by some manual, one-off analyses using the new data streams [43], [44]. However, much work was needed to enable more systematic use of this novel data to increase situational awareness in distribution networks. Certain fundamental properties of the μPMU data needed to be better understood, and while a platform that allowed easy, efficient access to high resolution data was already developed as part of the pilot, fundamental algorithms that could use the platform to sift through and analyze large volumes of data were still needed. This was the context in which I began my PhD in 2016. In the remaining sections of this chapter, I describe my work in addressing these foundational gaps. Section 2.2 describes the development of a workable model for the sensor noise present in data from field installed μPMUs. Section 2.3 describes the structure of the high performance time series database developed to ingest and store μPMU data. A simple use case is presented that highlights how algorithms can leverage the database structure to rapidly run across large volumes of data.

2.2 PMU NOISE

A model for the noise present in μPMU—and more generally PMU—data has fundamental value both for research and field applications of PMU data. For researchers, gaining access to real-world PMU data for algorithm testing and validation is still onerous, due to the legal and privacy constraints on using utility data sets and the relative paucity of device deployments, especially in distribution networks. Consequently, many algorithms utilizing PMU data proposed in the literature are validated in simulation alone. In this context, the effect of sensor noise is either overlooked entirely, or white noise with arbitrary variance is added to simulated data to represent real noise [45]. The noise model and signal-to-noise ratios developed here may be used to better incorporate the effect of noise in simulation studies through the creation of more realistic PMU noise.

A PMU noise model is also useful for field applications, such as state estimation. At present, most state estimators use weighted least squares (WLS) and power flow models to combine PMU data with other network measurement types from which they compute a maximum likelihood estimate of the true grid state [46], [47]. A PMU noise model would inform the selection of the weights in such a state estimation algorithm, which could improve performance over the heuristics currently used. Dynamic state estimation, which incorporates the system's historical state into future estimates, remains an active research topic [48], [49]. Dynamic state estimators use a Bayesian update step to generate a state estimate that balances confidence in the expected state against that in the measured state. Again, a PMU noise model could inform this trade off in a more principled manner.

PMUs are promising for use in control applications. For example, an inverter might be controlled through a feedback loop incorporating PMU data which adjusts the inverter's output to maintain a target nodal voltage. In this case, the noise in the PMU data can substantially impact the inverter's output. Especially in distribution feeders, where line impedances are low, small errors in the voltage magnitude reported by a PMU can lead to large changes in the commanded level of actuation. For example, consider a standard line from the IEEE 13-bus model with impedance $\mathbf{z} = 0.0756 + j0.0423$ (p.u.) and voltage magnitude $v = 2.4\text{kV}$ [50]. The (per unit) current error induced by a 0.5% error in the voltage measurement is:

$$|\mathbf{i}_{\text{err}}| = \left|\frac{v_{\text{err}}}{\mathbf{z}}\right| = \left|\frac{2.4 \times 0.005}{0.0756 + j0.0423}\right| \approx 0.140$$

At a 2.4kV voltage level, this 140A translates to approximately 300 kVA of power. This large amount of erroneous actuation may be caused by a seemingly low, 0.5% level of sensor noise.

Quantifying the noise level in PMU data is similarly vital in the context of estimation. Consider the problem of estimating a line impedance from phasor voltage and current measurements. For this purely illustrative example, assume a dc, real-valued model in which the current measurement i are perfect, while the measured voltage, $\tilde{v}$, is the true voltage, v, contaminated by additive white noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$: $\tilde{v} = v + \varepsilon$. With voltage measurements at two ends of a line, $\tilde{v}_k$ and $\tilde{v}_l$, and a noiseless measurement of the current flowing along the line, i_{kl}, the line impedance $\tilde{z}$ can be estimated as follows:

$$\tilde{z} = \frac{\tilde{v}_k - \tilde{v}_l}{i_{kl}} = z + \frac{\varepsilon_k - \varepsilon_l}{i_{kl}}$$

where z is the true line impedance, and ε_k and ε_l are the noise in the voltage measurements on each end of the line. Assuming the noise in the two voltages is independent but identically distributed, the estimated impedance can be modeled as the true line impedance contaminated by white noise with distribution $\mathcal{N}(0, \frac{2\sigma^2}{i^2})$. In the case of additive white noise, one way to combat the effects of noise is to average multiple estimates. Let $\tilde{z}(1), ..., \tilde{z}(n)$ denote multiple estimates of z computed over time from different noisy voltage measurements but under identical line current conditions. A new impedance estimate, denoted $\tilde{z}_{avg}$, can be obtained by averaging the n individual estimates:

$$\tilde{z}_{avg} = \frac{1}{n}\sum_{t=1}^{n} \tilde{z}(t) = \frac{1}{n}\sum_{t=1}^{n}\left(z + \frac{\varepsilon_k(t) - \varepsilon_l(t)}{i}\right) \triangleq z + \varepsilon_{avg}$$

$$\varepsilon_{avg} = \frac{1}{n \cdot i}\sum_{t=1}^{n}(\varepsilon_k(t) - \varepsilon_l(t)) \sim \mathcal{N}(0, \frac{2\sigma^2}{n \bullet i})$$

Therefore, the estimation accuracy is parameterized by $\frac{2\sigma^2}{N \bullet I}$. To achieve a desired estimation accuracy, n can be chosen appropriately, but first it is essential to know the underlying noise level, that is σ^2.

The following sections detail the efforts of my collaborators and I to validate a realistic PMU noise model and level from three days of data from μPMU devices deployed on an operational distribution feeder. These results are valuable to all those working with PMU data, and will also be relevant to the applications described in later chapters.

2.2.1 PMU Noise Background

There are two components to a complete description of PMU noise: the noise model and the noise level. The *noise level* defines the amount of noise present in the signal. Noise level is usually quantified by the variance of the noise random variable, or as the ratio between the signal and noise variance (termed the signal-to-noise ratio, or SNR). The noise level conveys a sense of how much an individual measurement can be trusted. The *noise model* parametrizes the noise in the signal, defining how the noise gets added to the "true" (and unknown) phasor to produce the reported noisy data. Understanding the noise model is fundamental for designing noise robust algorithms, as techniques to handle noise differ based on the noise model at play. For example, the techniques to combat additive noise are different from those to deal with multiplicative noise. There is further complexity when working in the phasor domain, where noise is present in both magnitude and angle measurements.

Prior Work

To our knowledge, the only prior empirical study of PMU noise is [45]. The authors assess noise using three different data sets from PMUs deployed at three voltage levels: low voltage (120V), medium voltage (20kV), and high voltage (345kV). They attempt to estimate the noise level using measurements from a *single* PMU with no external information about the true value being measured. Naturally, this is a difficult task, and to make it tractable, the authors choose a window length m and assume that the *median* over the window is the true phasor value while all variation in the window from the median is noise. The selected m differs between the three data sets, whose PMUs have different reporting frequencies. For the low, medium, and high voltage data sets, the chosen m corresponds to a window length of 0.8s, 0.5s, and 8.3s respectively. This approach to noise estimation seems inadequate, especially for high precision μPMUs. It is inaccurate to assume that all variation from the median over the m-length window corresponds to noise, as illustrated by Fig. 6 which plots the correlation between the voltage magnitudes reported by two PMUs monitoring the same voltage. The correlation between two n-length measurement time series $\tilde{v}_k(1), ..., \tilde{v}_k(n)$ and $\tilde{v}_l(1), ..., \tilde{v}_l(n)$ is defined as:

$$\mathrm{corr}(\tilde{v}_k, \tilde{v}_l) = \frac{1}{n}\sum_{t=1}^{n} \frac{(\tilde{v}_k(t) - \mu_k)(\tilde{v}_l(t) - \mu_l)}{\sigma_k \sigma_l}$$

where μ_k, σ_k are the sample mean and variance of the voltage at end k of the line. This quantity will be bounded between -1 and 1. The correlations are plotted for increasing sample *aggregation*, which refers to further down-sampling the signal using the mean or median. This process replaces m data points with a single point that is either the mean or the median of

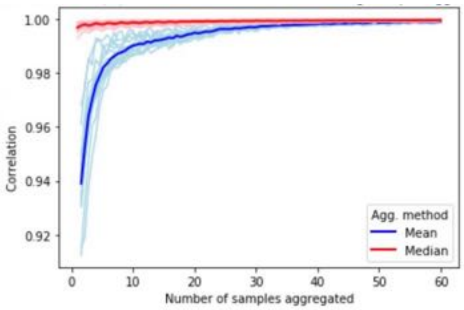


Figure 6: The correlation of voltage magnitude streams from two PMUs monitoring the same voltage with increasing aggregation (down-sampling) of the data. PMUs report at 120Hz and the highest level of aggregation corresponds to 0.5 seconds

the original m points. As the signal is down-sampled, the noise is reduced through aggregation, and consequently the correlation between the two voltage time-series should increase. This is the premise of the work in [45] and is indeed what happens in Fig. 6: note the lines for both aggregation methods converging toward 1 as m grows. However, Fig. 6 shows that even at low levels of aggregation, the correlation between the two voltage streams is very high. This observation strongly suggests that the variation in the PMU data even over short duration can not be dismissed as noise—which would produce low correlation—as the same variation is visible across two PMUs. Therefore the method and results of [45] may be too pessimistic, motivating another approach to noise estimation.

2.2.2 Experimental Setup

Our attempt at PMU noise estimation uses a setup consisting of two, identical µPMUs—labeled k and l—measuring a *single voltage*. The PMUs are deployed on a distribution feeder in Northern California and are plugged directly into a wall socket—that is on the secondary side of the service transformer—at which they measure a 120V line-to-neutral or 208V line-to-line point voltage. One known source of noise in distribution PMU data is the voltage and current transformers that mediate between distribution lines and PMU devices [51]. This transformer noise is believed to add a bias to the measurements that changes gradually and minimally over durations on the order of weeks. With adequate sensor deployment, this bias can be calibrated for with techniques such as that proposed in [52]. For these reasons, transformer noise is generally referred to as *systematic*. In contrast, we are focused on the *random* noise in PMU measurements. This noise is important because, in general, it must be handled in an online fashion, at the time when the noisy data is used. Therefore, understanding and quantifying the level and type of this noise is vital to enable PMU data users to integrate techniques for noise robustness into their PMU use cases. We consider the noise in voltage phasors, and therefore use both magnitude and angle data. Let $\tilde{v}_k(t), \tilde{v}_l(t)$ denote the voltage magnitudes reported by each PMU at some time point t. Then, $v(t)$ denotes the true voltage magnitude at that time. Similarly $\tilde{\theta}_k(t), \tilde{\theta}_l(t)$ denote the reported voltage angles, while $\theta(t)$ is the true voltage angle. The reported phasors can be expressed in rectangular coordinates by Euler's formula:

$$\tilde{v}_k(t)e^{j\tilde{\theta}_k(t)} = \tilde{v}_k(t)\cos\tilde{\theta}_k(t) + j\tilde{v}_k(t)\sin\tilde{\theta}_k(t) \triangleq \tilde{v}_k^{re}(t) + j\tilde{v}_k^{im}(t)$$

Note that the phasor angle θ in this context is not related to the power factor angle, and in general *will* vary from 0 to 2π radians.

2.2.3 Determining the Noise Model

A first principles approach to determining the noise model would entail expressing mathematically each step of the PMU measurement process—from transformer physics to phasor model fitting—while carefully accounting for each source of noise along the way. This is extremely difficult, if not practically impossible. Instead, we propose a data-driven approach, hypothesizing two simple noise models and then using reported PMU data to determine which one better expresses the empirical noise.

The first model is a *multiplicative phasor noise* (MP) model in which the reported and true phasors are related as:

$$\tilde{v}_k(t)e^{j\tilde{\theta}_k(t)} = \varepsilon_k(t)e^{j\phi_k(t)} \cdot v(t)e^{j\theta(t)} = (\varepsilon_k(t)v(t))e^{j(\theta(t)+\phi_k(t))} \tag{3}$$

An equivalent equation can be written for $v_l(t)$. ε and ϕ are respectively the magnitude and phase angle of the multiplicative noise. ϵ is centered around 1 and ϕ is centered around 0. The MP model leads to multiplicative noise in the voltage magnitude and additive noise in the angle.

The second model is an *additive phasor noise* (AP) model in which the reported and true phasors are related in rectangular form as:

$$\tilde{v}_k^{re}(t) + j\tilde{v}_k^{im}(t) = \left(v^{re}(t) + \varepsilon_k^{re}(t)\right) + j\left(v^{im}(t) + \varepsilon_k^{im}(t)\right) \tag{4}$$

Noise terms ε_k^{re} and ε_k^{im} are centered around 0. The MP model is more intelligible in the polar domain, considering noise in magnitudes and angles independently. The opposite is true of the AP model, where a rectangular formulation allows independent consideration of the real and imaginary noise components.

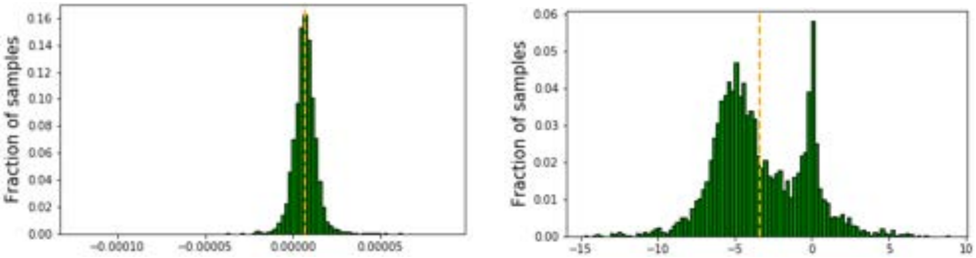
The PMU reports phasor quantities $\tilde{v}_k(t), \tilde{\theta}_k(t)$ from which $\tilde{v}_k^{re}(t), \tilde{v}_k^{im}(t)$ are computed. The true voltage quantities —$v, \theta, v^{re}, v^{im}$ —and noise quantities —$\varepsilon_k, \phi_k, \varepsilon_k^{re}, \varepsilon_k^{im}$ at PMU k—are treated as random variables of unknown, static distribution. Furthermore, the noise random variables are assumed to be unbiased (zero-mean for additive components, and mean one for multiplicative components) and independent of the true quantities as well as each other. The noise distributions are assumed to be fixed over time so $v(t)$ is an *instance* of the random variable v. Finally the noise is assumed to be symmetric, or identically distributed, across PMUs.

Distinguishing between an additive and multiplicative noise model is essential for developing applications of PMU data. Consider the simple scenario of trying to obtain an accurate voltage magnitude value from multiple readings of a PMU monitoring a fixed voltage with magnitude v and angle 0° under purely real noise. If noise is multiplicative, the reported magnitude is $\tilde{v} = \varepsilon v$, whereas if noise is additive, it is $\tilde{v} = v + \varepsilon$. Under additive noise, simply averaging multiple data points will reduce the noise variance. However, if the noise is multiplicative, the noise variance will be scaled by the signal magnitude v, even after averaging. Instead, the log of the measurements should be averaged, illustrating the point that multiplicative and additive noise must be treated differently in data applications. To determine which is the appropriate model for μPMU data, we propose two tests: a multiplicative model test and an additive model test.

Multiplicative Model Test

We formulate a covariance test to assess if the noise follows the MP model. Under the MP model, PMU k reports magnitude $\tilde{v}_k(t) = \varepsilon_k(t)v(t)$ and angle $\tilde{\theta}_k(t) = \theta(t) + \phi_k(t)$ and PMU l reports magnitude $\tilde{v}_l(t) = \varepsilon_l(t)v(t)$ and angle $\tilde{\theta}_l(t) = \theta(t) + \phi_l(t)$. Define a new random variable to be the difference in reported magnitudes: $d = \tilde{v}_k - \tilde{v}_l = (\varepsilon_k - \varepsilon_l)v$. The covariance between d and v_k is:

$$\begin{aligned} cov(d, \tilde{v}_k) &= \mathbb{E}[d\tilde{v}_1] - \mathbb{E}[d]\mathbb{E}[\tilde{v}_k] \\ = \mathbb{E}[v^2]var(\varepsilon_k) - \mathbb{E}[v]^2\mathbb{E}[\varepsilon_k - \varepsilon_l]\mathbb{E}[\varepsilon_1] &= \mathbb{E}[v^2]var(\varepsilon_k) \end{aligned} \tag{5}$$



(a) $cov(d, \tilde{v}_k)$: MP model test **(b)** $cov(d, \tilde{v}_k^{re})$: AP model test

Figure 7: Distributions of the sample covariances computed for the MP and AP model tests, with yellow lines marking means across all samples. Note that the definition of d differs between the two tests.

The manufacturer reported magnitude error is ±0.05%. Letting this error equal the noise standard deviation, an optimistic value for $var(\varepsilon_k)$ is $\approx (5 \times 10^{-4})^2 = 2.5 \times 10^{-7}$. At 120V nominal, $\mathbb{E}[v^2] \approx 14400$, resulting in $cov(d, \tilde{v}_k) \approx 3.6 \times 10^{-3}$—a lower bound on $cov(d, \tilde{v}_k)$ under the MP model. Intuitively, the MP test asserts that, in the case of multiplicative noise, the amount the measured quantity differs from the true quantity (captured in random variable d) is correlated with the true quantity itself (v in our case). This is because the noise *scales* the true signal.

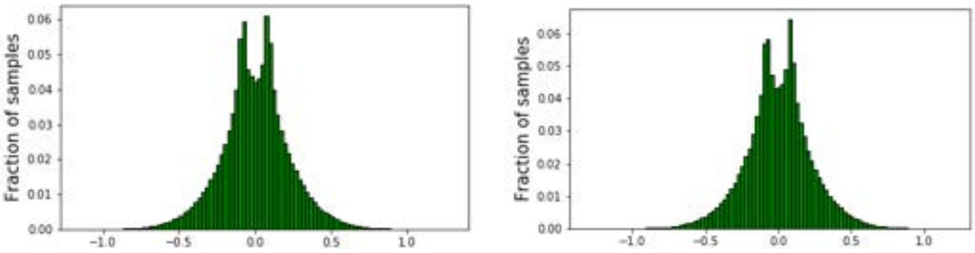
We compute $cov(d, \tilde{v}_k)$ over one second (120 samples) windows of data from multiple hours during the day. Fig. 7a shows the resulting distribution of sample covariances. Across many trials, $cov(d, \tilde{v}_k)$ is very low, with an average of:

$$cov(d, \tilde{v}_k) = 6.59 \times 10^{-6}$$

This would imply an unrealistically low noise level with $var(\varepsilon_1) \approx 4.58 \times 10^{-10}$. Therefore, the MP model is inappropriate to describe the observed PMU noise.

Additive Model Test

We formulate a covariance test to assess if the noise follows the AP model. Under the AP model, PMU k reports complex voltage $\tilde{v}_k^{re}(t) + j\tilde{v}_k^{im}(t) = (\tilde{v}^{re}(t) + \varepsilon_k^{re}(t)) + j(\tilde{v}^{im}(t) + \varepsilon_k^{im}(t))$ while PMU l reports complex voltage $\tilde{v}_l^{re}(t) + j\tilde{v}_l^{im}(t) = (\tilde{v}^{re}(t) + \varepsilon_l^{re}(t)) + j(\tilde{v}^{im}(t) + \varepsilon_l^{im}(t))$. Define a new random



(a) $\tilde{v}_k^{re} - \tilde{v}_l^{re}$

(b) $\tilde{v}_k^{im} - \tilde{v}_l^{im}$

Figure 8: Distributions of differences in real and imaginary parts of voltage reported by PMUs k and l.

variable to be the difference in reported real voltage components: $d = \tilde{v}_k^{re} - \tilde{v}_l^{re} = \varepsilon_k^{re} - \varepsilon_l^{re}$. The covariance between d and $\tilde{v}_k^{re}$ is:

$$cov(d, \tilde{v}_k^{re}) = \mathbb{E}[d\tilde{v}_k^{re}] - \mathbb{E}[d]\mathbb{E}[\tilde{v}_k^{re}] = var(\varepsilon_k^{re}) \tag{6}$$

If the AP model holds, sample estimates of $cov(d, \tilde{v}_k^{re})$ should be small and independent of the true voltage v. Intuitively, the AP model test asserts that under additive noise, the difference between the measured and true quantity (captured in random variable d) is uncorrelated with the true quantity v^{re}.

As the PMUs report voltages in polar form, voltages must be translated to rectangular form for the AP test. Fig. 7b shows the distribution of $cov(d, \tilde{v}_k^{re})$ samples computed over 1 second of data, with an average value of:

$$cov(d, \tilde{v}_k^{re}) = -3.41$$

At 120V nominal this corresponds to approximately 1.5% error for noise within one standard deviation. These results suggest that the AP model is reasonably accurate for describing noise in PMU measurements. Nevertheless, the estimate of $cov(d, \tilde{v}_k^{re})$ is slightly higher than expected. Notice the samples of $cov(d, \tilde{v}_k^{re})$ are bimodally distributed (Fig. 7b). Indeed, one of the peaks is close to 0, bolstering the validitiy of the AP model. The other peak likely arises from our simplifying assumptions: a slight bias in d and/or $\tilde{v}_k^{re}$ will cause $cov(d, \tilde{v}_k^{re})$ to differ from the value we derived.

2.2.4 Determining the Noise Level

Having validated the AP noise model, we can use it to derive an estimate of the noise level present in the PMU data. The noise level is captured in the variance quantities $var(\varepsilon_k^{re})$ and $var(\varepsilon_k^{im})$ (which, under our assumptions, are equal to $var(\varepsilon_l^{re})$ and $var(\varepsilon_l^{im})$ respectively). Estimates of these variances can be obtained from the following equations:

$$var(\tilde{v}_k^{re} - \tilde{v}_l^{re}) = var(\varepsilon_k^{re} - \varepsilon_l^{re}) = 2var(\varepsilon_k^{re})$$
$$var(\tilde{v}_k^{im} - \tilde{v}_l^{im}) = var(\varepsilon_k^{im} - \varepsilon_l^{im}) = 2var(\varepsilon_k^{im})$$

Fig. 8 plots the distribution of $\tilde{v}_k^{re} - \tilde{v}_l^{re}$ and $\tilde{v}_k^{im} - \tilde{v}_l^{im}$. From the variances of these distributions, we obtain the estimates: $var(\varepsilon^{re}) = 0.024$ and $var(\varepsilon^{im}) = 0.024$, assumed equal across PMUs k and l. The signal-to-noise ratio (SNR) of the data stream $\tilde{v}^{re}$ is defined as $snr^{re} = \frac{\mathbb{E}[(v^{re})^2]}{var(\varepsilon^{re})}$, with snr^{im} defined equivalently. The numerator $\mathbb{E}[(v^{re})^2]$ can not be computed exactly, and is estimated as $\mathbb{E}[(\tilde{v}_k^{re})^2]$. The final SNR estimates are:

$$\text{snr}^{re} = 3.09 \times 10^5, \ \text{snr}^{im} = 3.08 \times 10^5$$

which is approximately 55 dB.

The AP model test covariance in (6) could have been used to estimate the noise level, but directly using $var(\tilde{v}_k^{re} - \tilde{v}_l^{re})$ and $var(\tilde{v}_k^{im} - \tilde{v}_l^{im})$ reduces errors introduced by our simplifying assumptions. Deviations in the true statistics of d will be scaled by $\mathbb{E}[\tilde{v}_k^{re}]$ in (6), which is not the case when we use the variance of the differences directly.

* * *

To summarize: in this section, we used µPMU data from sensors deployed on an operational distribution feeder to validate a tractable additive phasor noise model and estimate the noise level for PMU data. The noise model and noise level form an important part of the foundational understanding on which algorithms and tools using PMU measurements can be built.

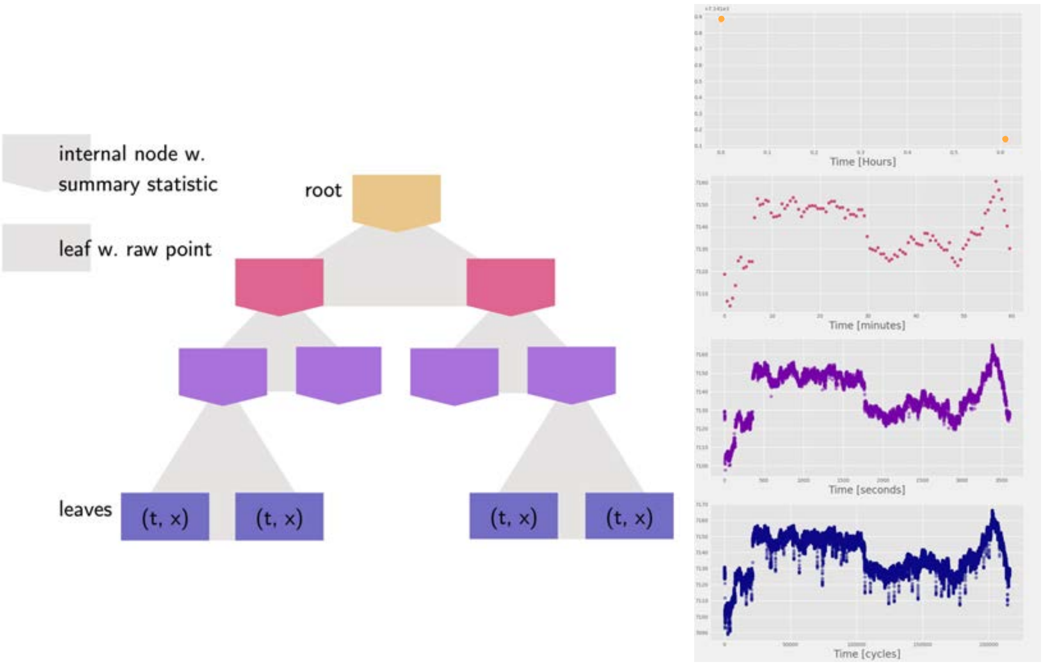


Figure 9: The structure of the Berkeley Tree Database (BTrDB). Plots at right show different time resolutions of data, with full resolution data at the bottom, and mean downsampled data at lower resolutions above. The lower resolution means are stored at internal nodes in the database tree.

2.3 MEASUREMENT PLATFORM

The Berkeley Tree Database (BTrDB)[2] was developed as part of the μPMU pilot project to store the large volume of time series data produced by μPMUs as they report at 120 Hz [53]. Though developed for this purpose, BTrDB can handle generic time series data, in which each data point is a tuple of a timestamp and value: (t, x). Thanks to its special architecture, the database is extraordinarily swift at both writing and reading data, outperforming prior phasor databases by several orders of magnitude. Consequently, BTrDB not only handles the ingress of 120 Hz data from numerous μPMUs (each of which typically generate 12 data streams: magnitudes and angles of voltages and currents on three phases), but also supports rapid computation on and interactive visualization of data streams.

For the purposes of this thesis and, more generally, the design of algorithms using μPMU data, it is the structure of data storage in BTrDB that is pertinent. As the name suggests, BTrDB is *tree-structured* (Fig. 9). The nodes at the bottom of the tree are the *leaves*. Leaves store the raw data points—tuples of time stamps and values—reported by sensors. Moving up through the tree from the leaves towards the *root*, internal nodes are encountered. Each internal node corresponds to a time range, defined by the time stamps of the left and right-most leaves lying below it. An internal node stores the statistical summaries (minimum, mean, maximum, standard deviation) over all the data in its time range. Taken together, the data in the leaves is the full resolution time series data reported by the sensor. The data in internal nodes at a particular level in the tree corresponds to a lower resolution version of this time series, down sampled using either the minimum, mean, maximum, or standard deviation (depending on which summary statistic is considered). In the extreme, the tree root stores the minimum, mean, maximum, and standard deviation of the entire stream: a single point summary of all the data.

Accessing data at internal nodes is significantly faster than reaching into the full resolution data in the leaves. Therefore, algorithms which leverage the summary statistics in the database to avoid unnecessarily querying raw data points can run rapidly over long periods of high resolution data. Similarly, a long duration of data can be quickly visualized by querying and plotting the appropriate lower resolution summary statistics. Then, as a

---

[2] http://btrdb.io/

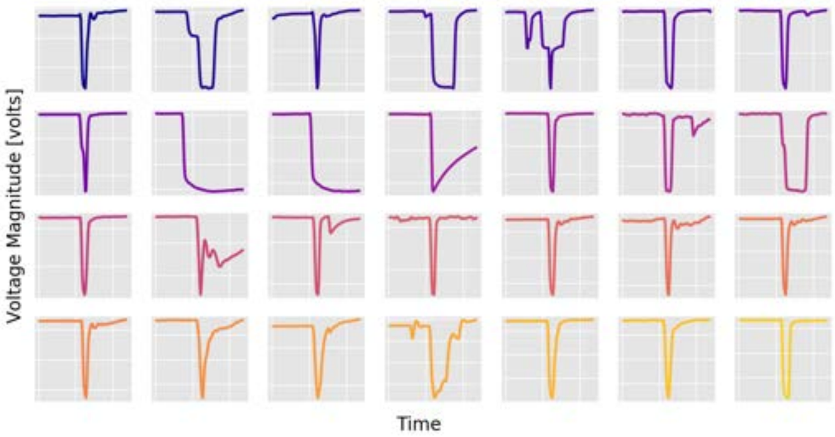


Figure 10: A set of events retrieved by depth-first search sag detection applied to a µPMU voltage magnitude data streams. The aggressive choice of threshold has led to some events being retrieved that do not resemble sags, though most of the events are some type of voltage sag.

particular period is zoomed into, only the necessary, shorter duration but higher resolution data need be queried from the database.

It is important to recognize the alternatives to the BTrDB platform [15], especially those currently in use in the industry. Many utilities store PMU and other grid sensor data in data historian applications which were built, not to enable algorithmic processing and analysis of data, but to store large data volumes to meet regulatory requirements on maintaining data history. Therefore, these data historians prioritize efficient data archiving, often using downsampling or lossy compression schemes that permanently destroy the high-resolution information in the raw measurements. For algorithmic data analysis, utilities depend on tools such as MATLAB or Excel, which make it highly challenging, if not impossible, to work with such enormous data sets at scale. Incredibly, at many utilities, high frequency PMU data is downloaded and shared in comma-separated-value (csv) files, which is an extremely inconvenient and unscalable method for data access. Early in my PhD, I worked with 120 Hz µPMU data shared in csv files, and I can attest to the near impossibility of effectively visualizing, exploring, and deploying algorithms on data in this form. In contrast to these options, the BTrDB platform is liberating.

2.3.1 The Example of Voltage Sag Detection

The enabling capabilities of the BTrDB platform for algorithm development are vividly illustrated by the use case of voltage sag detection [16].

Voltage sags are significant transient dips in voltage magnitude in an electric network that can persist from less than a cycle up to several seconds. They are relatively common events in transmission and distribution networks, with varied causes including equipment misoperation, faults, motor starts or the rapid reclosing operation of circuit breakers. Large, long, or frequent voltage sags can be problematic for utilities, causing sensitive loads to turn off, motors to stall, or solar photovoltaic inverters to trip offline. Many devices are pre-programmed to disconnect from the grid if they measure a significant excursion from the nominal system voltage, as occurs during a large voltage sag. The consequences of disconnection can be significant. Load trips can be a serious nuisance, with substantial economic losses particularly for large commercial customers. A large number of simultaneous inverter trips can lead to broader system instability, as was the case in the Blue Cut Fire Incident in California [54]. Altogether, knowing if and when voltage sags occur in their system can be useful to transmission and distribution operators. The high resolution of µPMU measurements means that many more, short duration voltage sags are visible in this data. The authors of [55] detail the manual study of voltage sag data collected during the µPMU pilot.

Automating voltage sag detection with the BTrDB platform is straightforward. As a voltage sag consists of a significant, temporally localized drop in the mostly flat voltage magnitude profile, it is easily found by looking through the summary statistics—specifically the minimum—stored in BTrDB. An efficient depth-first search algorithm for localizing voltage sags is presented in Listing 1, where `tau` is the user specified threshold below which a voltage deviation qualifies as a sag. Depth-first refers to the algorithm's approach of searching first at low time resolutions, and then proceeding deeper to high resolution data only when necessary. The algorithm begins by scanning through summary statistics at a low time resolution: these statistics correspond to long time windows of raw data. If the minimum within such a window is less than threshold `tau`, the algorithm traverses deeper and deeper down the tree until the minimum point is localized in the full resolution, raw µPMU data.

By leveraging BTrDB's structure, this algorithmic approach is highly efficient. For example, over a three month period of 120 Hz µPMU data, with an aggressive threshold of 0.99 times nominal voltage, the search finishes in 51 seconds, or approximately $(1.5 \cdot 10^5)\times$ real time. The algorithm finds 24 sags, visualized in Fig. 10.

This use case conveys the power of the BTrDB architecture for finding periods of interest in vast volumes of raw measurement data.

Listing 1: Depth-first search for voltage sags in BTrDB

```
def find_vsags_dfs(stream, tau, start, end, rez):
# Inputs
# stream - measurement stream in which to find sags
# tau - voltage threshold for sags
# start, end - times to search between
# rez - time resolution at which to search

    # Query summary statistics of windows
    # Time resolution rez specifies window width
    windows = stream.aligned_windows(start, end, rez)

    # Traverse left to right over windows
    for window in windows:
        # Check if window contains possible sag
        if window.min <= tau:
            # Get time range of window
            wstart = window.time
            wend = window.time + rez.nanoseconds

            if pw <= 30:
                # If window length <= 1 sec, get raw values
                points = stream.values(wstart, wend)
            else:
                # Otherwise, recurse deeper into tree
                points = find_vsags_dfs(stream, tau, wstart,
                    wend, rez-1)

            # Return only sag points
            for point in points:
                if point.value <= tau:
                    yield point
```

2.4 CONCLUSION

Let us take stock. This chapter laid the foundation on which algorithmic tools for grid monitoring and management are built. These tools take grid measurements as inputs, and Section 2.1 introduced several measurement types, with a particular focus on the high resolution, time synchronized phasor measurements from PMU devices. Extracting insights from measurements requires an understanding of the nature of the data itself. The type and level of noise present in the data is of particular importance if algorithms are to convert measurements into reliable and accurate information. Section 2.2 proposed a tractable noise model and estimated a noise level for μPMU data. Finally, especially when working with the large volumes of high resolution data generated by PMUs, an efficient platform for data storage and access is vital. Section 2.3 introduced the Berkeley Tree Database and illustrated how the structure of the database enables efficient measurement search, analysis and processing.

With the foundation laid, we can begin—not yet to build, but to envisage and plan the edifice of tools. The next chapter takes time to define and determine the kinds of tools we desire, before we start constructing them.

3 WHAT ARE USABLE TOOLS?

The academic literature teems with proposed algorithms for grid monitoring and management. The electric power industry is also increasingly adopting computational methods to improve *situational awareness*: the real time cognizance of grid state. I collectively term these computational methods—proposed in the literature and deployed in industry—*tools*, because it explicitly captures their intended purpose and mode of use, clarifying their desired design. The word tool instantly evokes an image of a helpful, physical object: a fork, a chisel, a wrench. We see the tool in the hand of a person, permitting them to effortlessly complete a task which would otherwise be difficult if not impossible. While this picture is slightly archaic for the present context, it is not altogether irrelevant (Fig. 11). Computational tools for grid management are virtual, housed in computers and not easily visualized in hand. Yet, like traditional tools, they aim to make a daunting task—that of understanding complex electric networks—tractable. They share another crucial feature with traditional tools: both are used by a human.

The word *tool* is evocative not just of use but also quality: we easily recognize the difference between good and bad tools. A good, or usable, tool is simple to use: intuitive and transparent. Transparency also makes the tool trustworthy and reliable. A tool lacking these features becomes unattractive and feels unhelpful. These criteria apply to grid tools as much as traditional ones. Transparency, trust, and reliability are especially vital in computational tools. The computing black box, however prescient, has a nightmarish quality; like the literal and figurative black box computer HAL in Clarke's Space Odyssey.

Yet, development of grid algorithms that satisfy the criteria of good tools is unfortunately meager. Indeed, many proposed grid algorithms, while technically ingenious, do not prioritize being good tools, giving minimal consideration to the needs of their human users. I believe this lacuna plays a large part in the limited transfer of algorithms from academia to industry. In this chapter, I survey tools for situational awareness in industry and literature, using the comparison to concretely define criteria for usable grid tools. I then propose a model for a grid monitoring *toolbox*. Creating grid tools that meet the criteria defined in this chapter is the aim of this thesis. More generally, it can ease the transfer of academic work into the real world, where tools can address the emerging challenges for situational awareness in the grid.

To demarcate a tractable scope, this chapter focuses on a subset of all tools: those that use data from PMU devices for the broad application category of situational awareness. Some of the tools may have very specific, narrow uses, while others are broadly useful. The focus here is not on the use case of a tool, but its design: what kind of result does it produce, how does it produce it, and how is the result delivered? I compare tools based on two qualitative dimensions. The first dimension relates to how a tool produces its result, while the second relates both to how a result is produced and the nature of the result. These dimensions are:

1. **Tool input information requirements.** How much input data (in the form of measurements or external system information) does the tool require?
2. **Tool Transparency.** How interpretable to a human user is the algorithmic approach and output of the tool?

Both dimensions are challenging to quantify precisely. Differences in the types of inputs to various tools makes information requirements difficult to compare across tools, while tool transparency can be a nebulous and contentious concept. Here, I aim instead for fair qualitative comparison in both dimensions.

When considering the amount of input information a tool requires, I consider the volume of information demanded. For example, a tool which requires measurements from sensors at every load connection has a high input information requirement. Similarly, a tool which requires an impedance value for every network line also has a high input information requirement. Two caveats are required here. First, some tools have flexible input data needs, in which case I consider the *minimum* data they require to produce a meaningful output. Second, some tools require large volumes of data during an offline training phase, but little data during online operation. Quantifying training data needs is highly challenging and often overlooked or concealed in the literature. Therefore, while noting the potential training data demands of tools, I compare them based on online data needs.

Tool transparency encompasses two features. A highly transparent tool has both algorithmic and output transparency. Algorithmic transparency refers to an algorithmic design for which the designer can define performance guarantees. Output transparency is even more stringent: the output must be interpretable to a typical user in the electric power industry—someone with a general understanding of power systems concepts and quantities but without extensive expertise in statistics or computer science. The disparity in the requirements on algorithmic and output transparency are motivated by the manner in which a tool is used. A tool user needn't understand the complete inner workings of a tool, but for them to trust it, the tool must provide clear guarantees—not that it will always work, but instead when it will work and when not. On the other hand, tool output(s) must be completely understood by the user if they are to enhance situational awareness and inform further actions. Transparent tool designs are especially urgent in the electric grid context where there is a need to bridge the cultural divide between engineers and operators, who have very different conceptualizations of and priorities for the system [60]. Another reason for urgency is the retirement of experienced grid operators who have built deep intuition for the system over long careers. They are replaced by young operators who need to accumulate their own empirical observations and experience of the system. Transparent tools can both bridge the operator-engineer divide and help new operators build understanding and intuition for the system.

In the next sections, I describe a range of tools from industry and academia, locating them in the two dimensional space of information requirements and transparency. The tool positions are visualized in Fig. 12.

(a) Sweetmeats [56] **(b)** Spinning [57] **(c)** Sculpture [58]

(d) Situational Awareness [59]

Figure 11: Tools for various purposes. Thinking of grid tools for situational awareness in the context of traditional tools elucidates how they will be used and therefore how they should be designed. Images adapted from cited sources.

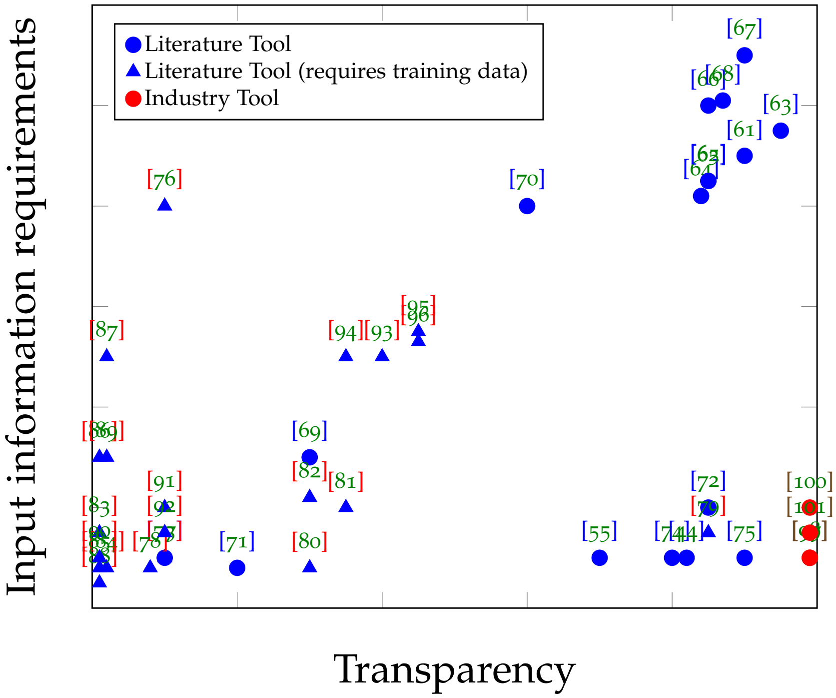


Figure 12: Visualizing tools from literature and industry in the two dimensional space of transparency and input information requirements.

3.1 TOOLS IN THE LITERATURE

This section surveys tools proposed in the literature, starting with those having high information requirements.

Model Estimation & Model Comparison

Several tools in the literature estimate the entire network structure, consisting of network lines and their impedances (this topic is the focus of Chapter 4 where it is covered in far greater depth). Monitoring the connections and impedances of a network is important both for successful management and control of the system, and for detecting structural changes that could indicate significant issues such as faults, equipment changes, or network reconfigurations. In [61], the proposed tool uses PMU voltage and current measurements from *every* network node to estimate the complete system impedance model. The tool of [62], with only slightly lower information demands, uses voltage data from every node to recover only network connections. [63] describes a fundamentally similar but more targeted approach, monitoring a single line using PMU measurements made at both ends. Extended to an entire network, this tool too demands data from every node. While these tools require almost no prior information on the system, they still have high information requirements in the form of sensor data with extensive spatial coverage. Algorithmically, these tools are based on physics equations relating network structure to measurements, which are inverted to estimate network structure from measurements alone. As long as the underlying physics equations hold, the tools are guaranteed to work. Further, the outputs of these tools are direct, physical quantities (electrical connections and impedances) with concrete meaning easily understood by a human. Using the tools' outputs to understand system changes —to the extent of determining the location and nature of the change —is essentially inbuilt. Consequently, these tools score highly in transparency.

Across use cases, many tool algorithms are based on physics equations. Often, the physics equations are approximations of the true system physics, and therefore the efficacy of the tool depends on the efficacy of the approximation which can vary across contexts. A class of approximations commonly used in tool designs are linearizations of the nonlinear, nonconvex power flow equations which relate nodal power injections to nodal voltage phasors. The tool in [64], based on a linearization, uses nodal voltage angle measurements to detect line outages. [65] proposes a linearization-based tool for fault detection from nodal voltage phasors. While the use of linearizations does not significantly reduce the information requirements of tools ([64] and [65] both require extensive sensor data), it does produce tool outputs that emerge from physics equations and are therefore physically meaningful and interpretable to the user. Indeed, linearizations often preserve fundamental system features while being more easily interpreted by a human than the precise nonlinear system equations, which explains their widespread use in analysis. In general, linearizations are based on well-defined assumptions and approximations. Tools which use them can therefore precisely define performance guarantees accordingly. This, along with the interpretability of outputs, makes such tools highly transparent.

Another class of tools compare observed measurements to those predicted by system models. The models could encompass an entire transmission network or a single generator [102]. Discrepancies between the predicted and observed data may signal the occurrence of an event in the system. Such a data disparity for a generator, for example, could indicate anomalous equipment behavior within the generator or a fault in the sensor monitoring the generator. The tool in [66] monitors discrepancies between measurements and models to identify problematic generators. The tool in [67] detects and classifies structural changes in a network by comparing observed measurements to those simulated from a large library of models comprised of all possible operating configurations. Similarly, the tool in [68] detects and localizes events on a radial distribution line by comparing measured voltage magnitudes to those predicted using a network model. While this class of tools tends to have less stringent sensor coverage requirements, the input information demanded is still considerable, consisting of accurate, detailed system models. Creating and maintaining such models is always burdensome and sometimes impossible—for example, it is rare to have access to detailed, accurate models for distribution networks. Yet, by linking system data to rich, virtual models, the tools can produce outputs with physical specificity —for example, localizing a change to a specific line or piece of equipment —leading to high transparency.

Note that the tools described so far, which estimate quantities from physics equations or compare measured and modeled quantities, may be particularly sensitive to measurement noise. Noise must be considered in the design of such tools if they are to be suitable for real world use.

Thresholds: Simple to Elaborate

Numerous tools lack any connection to a model, whether in the form of a physics equation or a simulation. Such tools have varying levels of transparency, but generally lower information requirements. The simplest of these tools apply constant thresholds—often based on utility standards or definitions—to measurements to flag significant system events. The tool in [55] detects voltage sags by thresholding voltage magnitude data according to the utility definition of a sag event: “the decrease in RMS voltage between 10 and 90 percent of nominal for one-half cycle to one minute” [103]. When multiple measurement points are available, this approach can localize events to the extent of identifying their origin as either in the transmission or distribution system. The technique requires no prior information and needs measurements from limited locations. While it flags events, it provides almost no further insight for diagnosis or analysis. Yet, because the threshold is applied to raw data and set in a simple manner based on an industry standard, the algorithm and output of the tool are easily understood by the user. Therefore, within its limited but well-defined scope, the tool has high transparency.

Thresholds can also be applied to computed quantities rather than raw measurements. In this case, tool transparency depends on the interpretability of the computed quantity. Several tools detect events by applying thresholds to differences between voltage phase angles measured at multiple locations across a network, as described in [72]. Through a power flow linearization, angle differences have intuitive physical meaning in terms of power flows across impedances. Therefore, these tools have surprisingly high transparency—more so than simple voltage thresholding—and enable detection, localization, and diagnosis of physical system events. The input information requirements are flexible and generally low, though greater measurement coverage can enable more granular visibility. In contrast, the tool in [73] detects events by applying thresholds to wavelet coefficients computed from raw measurement time series. This tool’s transparency is low due to the limited transparency of wavelets. It is unclear how the user—especially a grid operator without signal processing background—is to understand what it means when an event has been detected based on an abnormal value of the n^{th} wavelet coefficient. Similarly, [69]–[71] have low transparency due to thresholding of less intuitive, computed quantities.

Additional complexity is added to threshold-based tools by defining more elaborate thresholds. Historical data can be used to determine the bounds of normal behavior with varying complexity (and proportionally varying volumes of historical data). For example, baselines can be computed with hourly and seasonal dependence, as discussed in [77]. The high impedance fault detection tool in [76] thresholds measurements based on a Gaussian distribution fitted to historical data. The transparency of these tools hinges on the transparency of the threshold choice and its presentation to the user. Many tools present "atypicality" scores [77], [78] which are generally opaque. On the other hand, a probability score based on a learned distribution is more interpretable to a user, with the added benefit of being effectively visualized. Such *statistical baselines* promise greater transparency than hard thresholds because they can inform the user not just that data is abnormal but how abnormal it is, expressed quantitatively and intuitively (for example, as a probability or the number of standard deviations from mean behavior).

Machine Learning Methods

A large set of tools employ supervised and unsupervised machine learning techniques on raw measurements or features computed from them. In general, these tools are applied to segments of data that have been extracted from measurement streams based on anomaly detectors such as those just described. Supervised techniques require *labeled* training data: datasets that include both the algorithm input and the desired, associated output (such as events captured in measurements and their associated labels). In contrast, unsupervised techniques may require training data, but do not require information external to the data. Therefore, they are particularly advantageous in the grid context where privacy concerns and data paucity compound to severely restrict the availability of labeled data.

Clustering is one unsupervised technique commonly used in event classification tools (a demonstration is visualized in Fig. 13). These tools take event data samples as inputs and return a "classification"—a cluster membership—for each, grouping them by similarity. No explicit label is produced, but recurring events can be identified in this manner, which can then assist in external analysis and diagnosis. Such tools tend to have low information input requirements in terms of spatial sensor coverage or prior network information, though some volume of data is needed to form meaningful clusters. While clustering itself is an intuitive operation, the

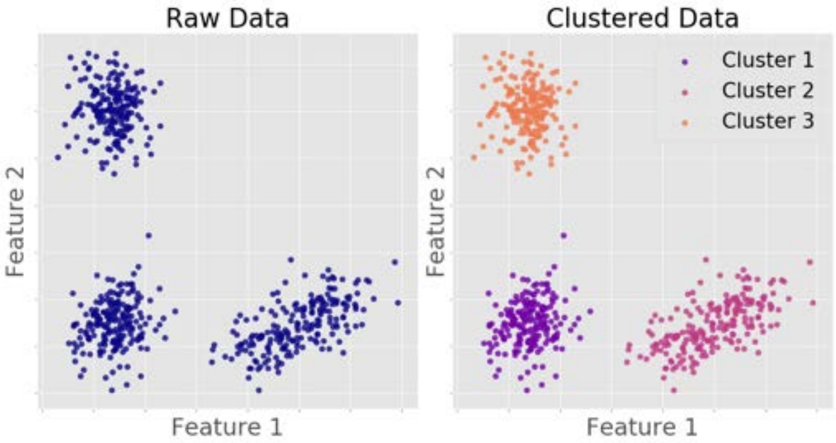


Figure 13: A toy demonstration of clustering. When a clustering algorithm is applied to the two-dimensional raw data visualized on the left, it groups the data points by similarity / proximity into the clusters on the right.

transparency of such tools depends on the transparency of the quantity being clustered and the distance metric used to determine clusters. The tool in [79] clusters three phase μPMU time series measurements with k-Means using Euclidean distance as the distance metric. This technique has high transparency as it clusters raw measurements with an intuitive distance metric. The identified clusters are easily visualized and clearly consist of similar events. Clustering of less intuitive or higher dimensional data diminishes transparency and trust in the results. This is the case for [80]–[82], all of which cluster larger sets of non-intuitive, derived features.

An emerging class of unsupervised tools are based on generative models, which learn a statistical distribution of the data from a training data set. The architecture choice of the model enables it to capture complex, nonlinear statistical dependencies. The transparency of these approaches benefits from the fact that the result can be conveyed as an interpretable probability quantity, however complex choices of architecture and model inputs compromise transparency. [91], [92] present such unsupervised techniques for event detection. While they may succeed at flagging events, these tools provide little further insight into the nature (location, type, etc.) of the event, which can make their outputs difficult to use effectively.

Despite the difficulties of obtaining training data in the real world, a variety of tools based on supervised machine learning have been proposed in the literature. Consider the motley of classifier tools: for protection fault detection [83], for event detection [84], for line fault detection [85], for cyber attack detection [86], [89], for event identification [87], for stability monitoring [88], for equipment malfunction detection [90], among many others. These tool techniques are black boxes with minimal transparency. While their online data demands may be low, they generally require significant volumes of training data and can be difficult to train: a point which is often glossed over in the literature. Even when trained successfully in one context, it is unclear if the tools will generalize to another context (a different feeder or interconnection, etc). Methods to address poor generalizability—for example by replacing a single classifier with several—further reduce transparency [104].

Physical Proxies

With a few exceptions, the tools discussed so far lie at the two extremes of transparency. While model estimation and model comparison tools have high transparency, machine learning tools tend to be black boxes with paltry transparency. This polarity hinges on the use of physically meaningful quantities in the approaches and outputs of these tools—which the model tools posses but machine learning tools lack. I now share a miscellany of tools which deal in quantities that are not explicitly physical but are related to physical quantities and can be termed physical proxies. Physical proxies carry the intuition and transparency of explicit physical quantities insomuch that, for example, a change in a physical proxy indicates the location and nature of a system event.

Some tools do deal in quantities that can be understood as physical proxies for increased transparency, but still fail to interpret the quantities as such. The algorithms underlying these tools lack mathematical derivations that contextualize their outputs as physical proxies by connecting them to physics equations of the system. These derivations are necessary to concretely establish implicit algorithmic assumptions and guarantees—vital both for algorithmic transparency and the correct interpretation of the physical proxy output. The absence of such derivations results in tools that miss an opportunity for increased transparency. Consider, for example, tools which monitor the principal components of PMU data to detect events [93], [94]. While principal components of voltages can have physical meaning under certain assumptions on the system, the tools in [93], [94] do not derive or interpret them in this manner, diminishing them from physical proxies to purely heuristic features, and compromising tool transparency. Similarly, several tools monitor measurement correlations. Though voltage correlations also have physical meaning under specific assumptions, the correlation-based tools in [95], [96] use correlations as purely heuristic, statistical quantities.

There are some tools in the literature which do better in deriving and using physical proxies. The tool in [74] estimates the *ZIP* parameters of loads from PMU data. These parameters do not represent a complete load model, but nevertheless carry physical intuition and are associated with a specific system element, bolstering transparency. The tool in [44] detects the switching operation of capacitor banks to infer the control parameters of these devices. These control parameters have clear intuitive meaning, and can be monitored to detect equipment malfunction. The tool in [75] estimates the quantity of distributed solar generation in a distribution feeder from PMU data. The algorithm is derived starting from a simplified solar generation model, under clear and explicit assumptions. The result consequently has wonderful transparency.

3.2 TOOLS IN INDUSTRY

As PMUs proliferate in operational electric grids, industry use of tools that leverage PMU data is expanding concurrently. Yet, the range of tools used in industry is narrow, with just a few tool types gaining widespread popularity. This is a stark testament of the impediments to tool transfer from academia—where numerous new tool algorithms are proposed each year—to industry. The differences between widely used industry tools and tools in the literature is telling of different attitudes towards usability. Here, I describe industry tools for system monitoring as well as some monitoring workflows which illustrate how tools are used.

Detecting and localizing power oscillations—which can manifest in voltages, currents, and frequency—is of particular interest to utilities at the transmission level, and is an important application of transmission PMUs. Several types of proprietary and open-source oscillation monitoring tools are employed in industry. Most of these compute oscillation mode and magnitude at different system locations and visualize these quantities directly to the user in an interface, some of which are shown in Fig. 14. These tools have low information requirements: while better source localization is enabled by increasing measurement coverage, the algorithms can be applied even to a single measurement stream. Transparency is high due to the presentation of intuitive, physically meaningful quantities directly to the user. From the presented oscillation data, the user can often infer the root cause as a particular generator [105].

Steady-state voltage angles across a network indicate power flows and electrical distance. They have intuitive physical meaning–especially in transmission networks, where mostly reactive lines result in more significant differences in voltage angles than magnitudes. Angle difference monitoring can therefore be used to detect system stress due to congestion caused by large power flows or loss of lines [100]. Several industry tools, some of which are shown in Fig. 15, compute and display voltage angles to the user. Similar to oscillation monitoring tools, angle monitoring tools have low information requirements—though increasing sensor coverage improves visibility—and high transparency due to their direct presentation of intuitive angle data.

The quantity with perhaps the longest history of monitoring in electric grids is frequency. For all those who deal with ac power systems, frequency has lucid meaning: it is uniform across a synchronous network and indicates the balance of energy generated and consumed. Therefore, monitoring frequency at multiple locations in a system can reveal the presence of electric "islands"—regions that are energized but isolated from the rest of the system. Fig. 16 shows the interfaces of several industry tools for island detection. These tools measure frequency at different points in the network, visualizing the results directly to the user who can easily identify an island by noticing areas with significantly different frequencies. These tools require few measurement inputs—though increasing coverage improves island detection—and have high transparency. Together, island detection and angle monitoring tools can also be used to successfully reconnect an island to the bulk system.

Tools for ad hoc analysis

Many applications of sensor data in the electric industry occur, not through established, targeted tools, but as ad hoc analyses for particular, often transient applications. While not typical tool examples, these analytic vignettes illuminate the needs of utilities and convey how tools will likely be used

Figure 14: Interfaces of industry tools for oscillation monitoring. Clockwise from top left: SEL, Space-Time Insight [99], ERCOT [97], BPA [98]

Figure 15: Interfaces of industry tools for angle difference monitoring. Clockwise from top left: ERCOT, Dominion, Columbia WAMS, and EPG [100]

Figure 16: Interfaces of industry tools for island detection. From left: EPG, OSIsoft, and Alstom [101]

and useful.
The authors of [106] relate several instances where sensor measurements, under investigation by analysts, revealed and diagnosed issues in operational networks. On one occasion, analysts noticed significant voltage imbalance across the three phases and pin-pointed the cause to the malfunctioning of one phase of a three phase transformer. Such asymmetrical behavior across phases could have been symptomatic of varied issues. It was the analysts expertise and experience of the system that enabled them to swiftly and accurately diagnose the issue manifested in the data. On other occasions detailed in [106], analysts used phasor data to identify capacitor bank switching problems, disconnections and unbalanced currents in circuit breakers, and arrester failures.
These incidents share common features: many of the issues detected are both critical and unusual, manifesting in complex ways across several data streams. Their diagnosis is far from fully automated, but instead occurs through rich interactions between human users with knowledge of the system and data from the system. The user's experience plays a critical role in ultimately identifying the root cause.

3.3 THE GAP

Altogether, the prior sections and Fig. 12 highlight a gap between the majority of tools in literature and those in industry, both in their fundamental aims and realized forms. In their forms, literature tools generally tradeoff transparency and input information requirements. While tools with high transparency mandate extensive input information, tools requiring little input information have low transparency. On the other hand, tools in industry prioritize low input information requirements while maintaining high transparency through explicable algorithms and physically meaningful output quantities.
This divergence in forms stems from a divergence in aims between literature and industry tools. Many literature tools either circumvent or neglect the human user, aiming to provide finalized, conclusive results. Once the human user is disregarded, the need for establishing trust and understanding—in short the need for transparency—fades. Indeed, this is why many literature tools are designed as end-to-end black boxes which

ingest measurements to generate complete diagnoses. Industry tools, in contrast, are definitively tools in the hands of human users. They generate intuitive outputs always intended for delivery to a human user. This arises from a recognition that the identification of many important events, such as those in [106], is highly challenging to automate completely, and that the experience and insight of the human user is crucial to correct diagnosis. Successful automation generally requires consistent characteristics or—in the case of machine learning approaches—numerous samples of a particular occurrence, both of which are lacking for many system issues, and especially for unusual ones. Humans, however, can draw on their intuition and knowledge of system equipment and physics to diagnose even rare, previously unseen issues.

A consequence of these divergent attitudes towards the human user is another stark difference between literature and industry tools: visualization. Industry tools include significant visualization components, while few literature tools discuss visualization at all. A broad survey of PMU tools found visualization to be a sliver of the full research pie (Fig. 17) [107].

This is not to suggest that industry tools are the grail for all tools. Rather, they are often siloed into particular use cases, and provide the bare minimum of insight, so that the brunt of analytical burden remains on the human user. Nevertheless, the fundamental aim of industry tools—to be useful to a *human user* even in *low information* scenarios—reflects a more realistic and persuasive vision of how computational tools will be used in the electric grid context. If literature tools are to translate to industry use, they must better heed this vision.

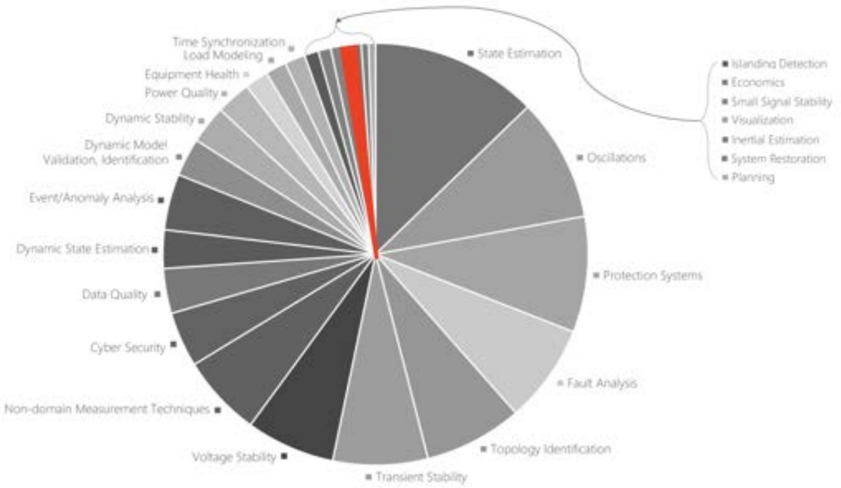


Figure 17: The proportion of research works dedicated to visualization in a large survey of PMU applications. Modified from [107].

3.4 BRIDGING THE GAP: USABLE TOOLS

In this section, I recommend three broad features that should be incorporated into tools to make them usable and bridge the gap between literature tools and industry application. To be usable, tools should:

- **have low and flexible input information requirements.** Algorithms with high information requirements—whether in the form of exhaustive sensor measurements or extensive system information—are unlikely to be workable in real-world contexts. While grid sensor deployments are growing, comprehensive measurement of distribution networks, with their highly branched topologies and multitudes of loads, is still remote. Similarly, accurate and detailed system information, such as line impedances or equipment characteristics, is scarce and challenging to obtain. Information in the form of training data is also limited. Building sufficiently large training data sets is expensive, stymied by privacy concerns, and essentially impossible for rare grid events. The usefulness of these training data sets is also suspect, as many event signatures may not generalize across networks and scenarios. Tools which require training on large data sets before they can be at all effectual are unlikely to be viable in real application. Flexibility of input information requirements allows tools to be used *and* useful across contexts with varying levels of information availability. Further, as information availability increases—for example, sensor deployments grow—such tools can accommodate new information to produce results with higher resolution and quality.

- **use physically meaningful quantities, ideally through basis in an underlying physics model.** This is an important feature for transparency and user trust of tools. The intelligibility of physically meaningful outputs to human users means that, rather than standing alone, the outputs augment users' situational awareness and can be used in further analysis. An algorithm built on a physics model not only produces meaningful outputs, but has structural transparency, with easier formulation of guarantees, including well-defined assumptions and failure modes.
- **consider the limitations of real measurements.** Few literature tools assess the impacts of real measurement characteristics on tool performance. In the case of several tools, this partly stems from inherent algorithmic opacity which makes it challenging to determine the impacts of noise or to articulate and quantify the importance of implicit assumptions. Many literature tools are only validated on simulated data, which differs strikingly from real grid data. It is often unclear and even dubious if and how these tools will translate to real data. Therefore, for literature tools to be usable in industry, it is essential that their very forms include features for real world robustness. Basis in physics models aid robustness as assumptions and noise impacts can be precisely defined.

These bridging features are summarized in Fig. 18, with the space of highly usable tools in the center of the Venn diagram.

3.4.1 Design Recommendations

This section describes a few concrete design recommendations to meet the criteria of usable tools.

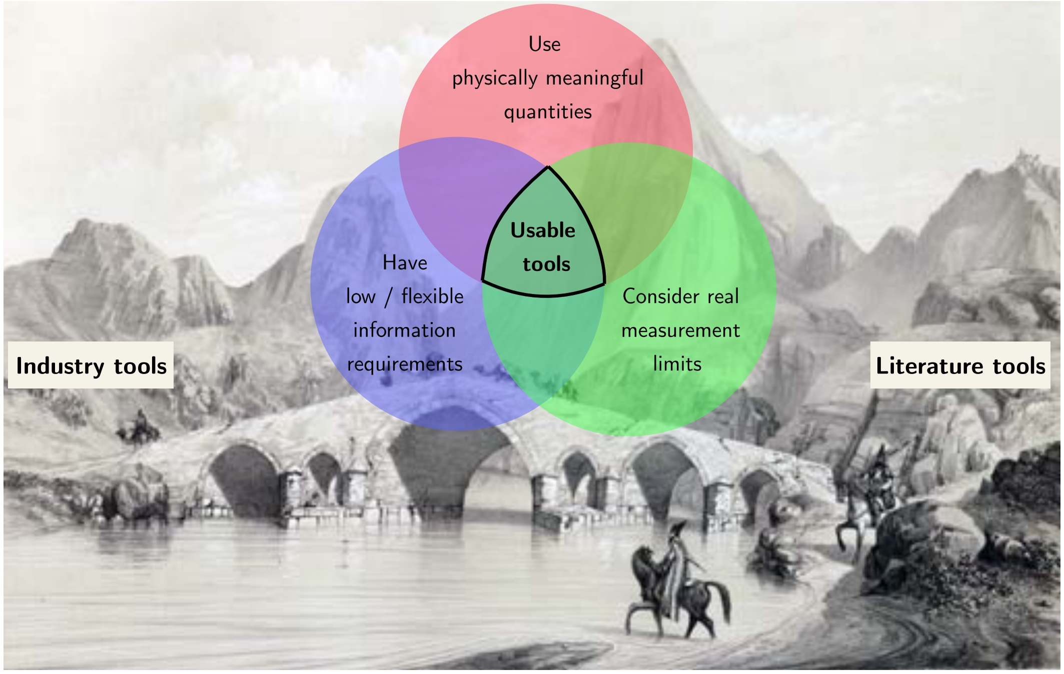


Figure 18: Three broad features that lead to tool usability and can bridge the gap between literature and industry tools. Background modified from [108].

Expose the Features

Many literature tools are black boxes that take in measurements and produce outputs. In between, they may compute features—insightful derived quantities—but these are obscured from the user; algorithmic ephemera. Instead, computing insightful, interpretable, physically meaningful features and *exposing* these directly to the user is a step towards usable tools. Exposing features co-opts—rather than bypasses—the human user in the decision-making and analysis process, leveraging their intuition, experience, and insight of the system. These features contribute to building the user's mental model of the system, based on which they can understand and diagnose system issues. Exposing the features often leaves the final inductive step to the user (a compromise is presented later in Section 3.5). In simple or frequent scenarios, the algorithm could automate this step, but relinquishing it to the user comes with benefits of transparency and trust. In complex and rare scenarios, the human may be essential to successfully conducting this final step. Examples of features to expose are: the measured voltage magnitudes at a bus, a simulated nodal voltage based on a utility model, the correlation between two nodal voltages, an estimated impedance quantity such as a source impedance, or an important angle difference.

Exposed features lead to modularity and flexibility. As algorithms are developed or sensors are deployed, new features can be presented to the user. Together, a diverse set of features create an increasingly complete and insightful picture of the system state. Changes in this feature set can indicate a range of issues and events, increasing flexibility and applicability compared to highly targeted, black-box tools. Another benefit of exposed features is resilience. Many existing tools tend to synthesize numerous data streams into a single output. A single erroneous input can completely contaminate the output of such tools. In contrast, presenting features computed from distinct raw measurement sets creates redundancy across features that can increase resilience to cyber attacks or sensor failures.

Statistical Baselines

Statistical baselines are statistical models of quantitative features. They capture both the typical range of a feature as well as the bounds of abnormality, expressed in intuitive probability terms. Therefore, statistical baselines can be used to flag unusual feature values, while also indicating their *level* of abnormality, reported as probabilities. Statistical baselines are also easily visualized for individual features. Altogether, this gives statistical baselines a transparency that is absent in algorithms which simply label data as normal or abnormal. For more binary needs, thresholds can be added atop statistical baselines to trigger event alarms. These thresholds can be intuitively adjusted, or formulated through Neyman-Pearson style tests (here is another opportunity for operator participation) which balance desired risks of false positives and false negatives [109]. The use of statistical baselines could address real measurement limitations by making algorithms more robust to noise. The normal variation in raw measurements or computed features due to noise can be encapsulated in the statistical baseline model, averting excessive and unnecessary alarms. Statistical baselines also aid system modularity because they can be separated from feature computation.

Time-Aligned Visualization

Time alignment is a powerful feature of modern grid sensors, notably PMUs. I propose leveraging this feature in the *outputs* of tools as well, by presenting the user with a time-aligned visualization—in which time series are vertically aligned by time—of the exposed feature streams (Fig. 20). This visualization addresses the broad deficiency of visualization capabilities in literature tools: a major impediment to usability. Presenting time-aligned features to the user builds their situational awareness and facilitates event diagnosis as they notice coinciding, anomalous events, all in a transparent manner. Given the high resolution of novel grid sensors, it is highly likely that perfectly coincident events, even at different locations, share a common cause. Many existing tools do not present time-aligned visualizations, aggravating user cognitive load as they struggle to determine and understand simultaneous events [110]. Another potential benefit of time-aligned visualization is pedagogical: it can train users to interpret and utilize novel data types or features. PMU data, for example, is newer than SCADA data, and due to their differing time resolutions, PMU and SCADA measurements can look significantly different over the same time. As PMU deployments grow, grid data users must familiarize themselves with these new measurements. Time-aligned visualizations of PMU data alongside SCADA data can build users' understanding of PMU streams by enabling easy comparison with the more familiar SCADA reference. The authors of [111] realized this too, in the context of a different application and system.

3.5 THE FUTURE TOOLBOX

Figs. 19-20 visualize the structure of a future grid *toolbox* incorporating the features and design recommendations of Section 3.4. This high-level proposed design takes inspiration from the system described in [112] created for the computer network monitoring domain, which perhaps surprisingly shares several defining characteristics with the grid management space.
Fig. 19 visualizes the information flows for the future grid toolbox. The inputs to the toolbox are data from a diversity of sources which, critically, are time synchronized. The quality of time synchronization required will vary across tools and applications, but some level of dependable time synchronization is essential for collating diverse and disparate data sources. Input data could include measurements from PMUs and smart meters, weather measurements and forecasts, outputs of simulation models, and customer phone calls, among others. Input data types and volumes will continue to grow, and new data can be easily incorporated in this framework. The input data is ingested by an extensible set of tools which meet the criteria of usability. Each tool may take in one or more data streams to output one or more interpretable features. Examples of intuitive and meaningful tool outputs are impedance estimates, angle differences, and solar output estimates among innumerable others. These output features, generally in the form of time-series, are then passed to statistical baselining algorithms, which learn the features' distributions. The baselining algorithms are independent of the tools, so an extensible library of methods can be created and used. This structure gives the system modularity and flexibility. Finally, the outputs are presented to the user in a time-aligned visualization.
Fig. 20 presents a mock-up of the visualization component of the future grid toolbox. Tool outputs—i.e. interpretable features—are presented to the human user vertically aligned by time. This allows the user to observe coincident changes in multiple features and thereby surmise the source of the change. The full set of features is too large to be visualized at once, so features are ordered based on their abnormality, under the assumption

that more anomalous values deserve greater attention. The statistical baselines enable this ranking and, for transparency, are also visualized with the feature time series. In Fig. 20, the learned feature distribution is visualized as a color gradient, while a likelihood quantity is displayed alongside each time series. Notice the least likely, or most anomalous, feature is presented first. User collaboration is enabled and solicited through like or dislike buttons attached to each feature; votes can be incorporated into future decisions to display a particular feature. This visualization is highly transparent, enabling and even relying on user analysis to make the final diagnostic conclusions. However, the future toolbox framework also allows for further automation without compromising transparency. Automated diagnoses or explanations can be suggested to the user alongside feature plots. In Fig.20, this is visualized for an event detected in Feature n. An automated event classification and diagnosis is indicated by a dashed hypothesis box, with an arrow that will display further details.

* * *

In this chapter, a comparison of tools widespread in industry and tools proposed in the literature motivated the definition of three broad features that beget usable tools. A few concrete design recommendations were presented, and ultimately an architecture for a versatile toolbox of several algorithmic tools was envisioned. This chapter lit the beacon for usable and useful computational tools; the next chapters relate one journey getting there.

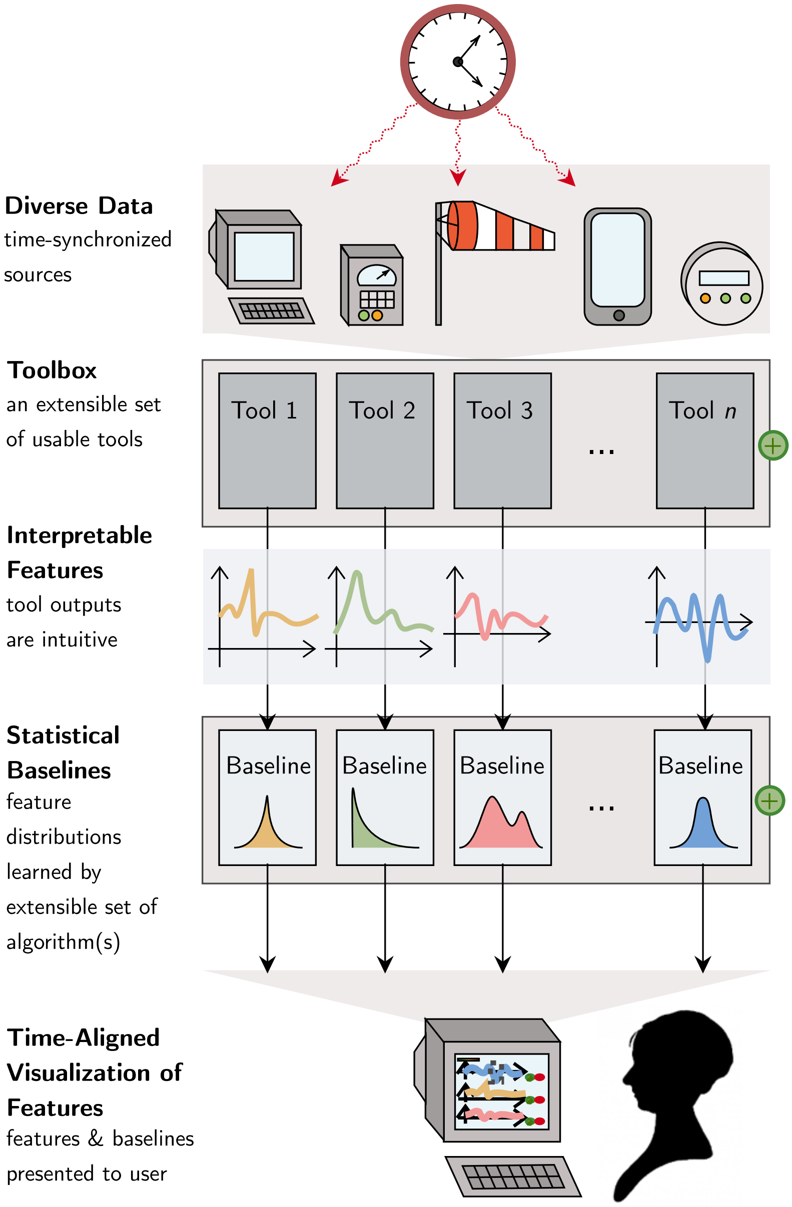


Figure 19: Information flows for the future grid toolbox. Data from varied, time synchronized sources is the input to computational tools. The tool outputs are interpretable features, mostly in the form of time series. These time series are passed through algorithms that statistically baseline the features. Baseline and tool algorithms are independent and extensible. Results are presented to a human user in the visualization of Fig. 20.

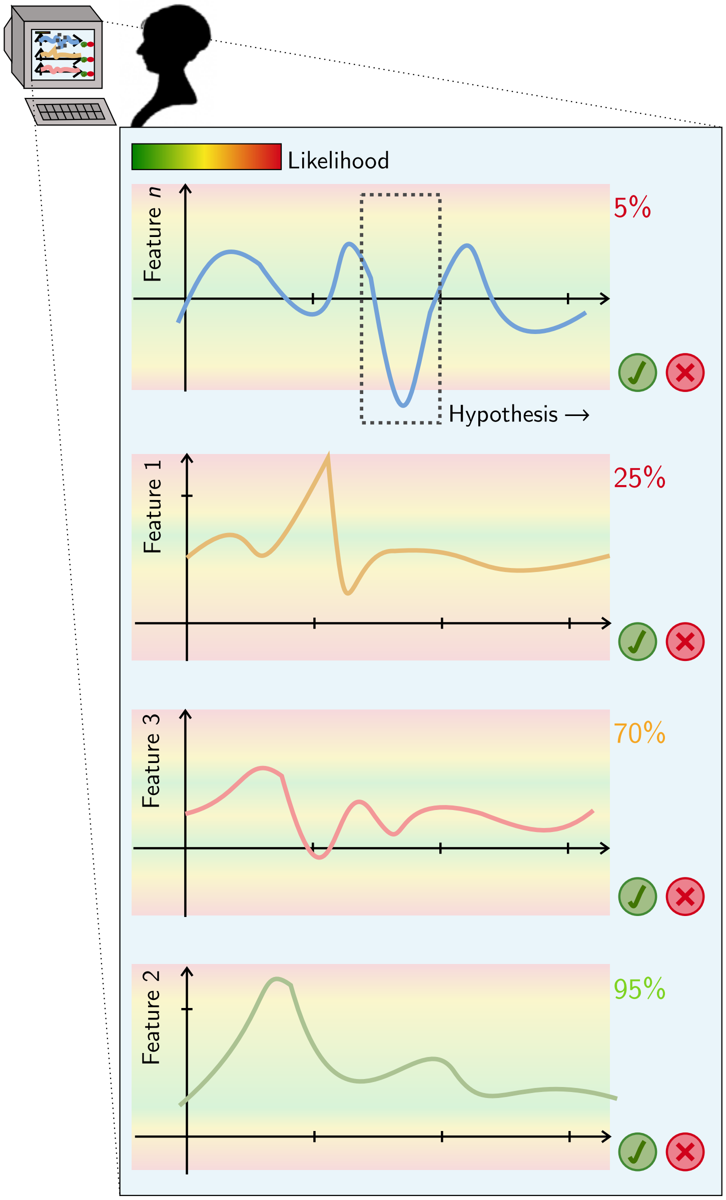


Figure 20: A time-aligned visualization presents the outputs of the future grid toolbox to the human user. Statistical baselines indicate the expected distribution of interpretable features, and the likelihood of the latest values. Users can indicate features which are helpful or unhelpful. Additional automation can be unobtrusively incorporated in the form of suggested diagnoses.

4 TOPOLOGY TOOLS

Includes work from [113]–[117]

At the level of abstraction inhabited by this thesis, an electric network consists solely of lines and buses. Lines are electric conductors of some finite impedance along which power flows. Buses are juncture points where lines, loads, generators, and other components connect—though for our purposes these individual components are generally abstracted away: encapsulated within buses and represented by net flows. Buses provide convenient reference points at which to specify system voltages, and are also the only points where current can enter and leave the network. Power, on the other hand, transits at buses but is also dissipated on lines in the form of losses.

Topology is the fundamental property of all electric networks. It defines the network structure by describing exactly how buses are connected by lines. A complete network topology indicates the bus endpoints and impedances of every line, as well as the phase label of each line (see Interlude below). Topology is essential for understanding network behavior. For example, by specifying current injections at every network bus, applying Ohm's and Kirchhoff's Laws to the topology allows for the derivation of bus voltages, line flows, and losses. Similarly, when bus power injections are specified, optimal power flow uses the topology to determine voltages, flows, and losses. All together, knowledge of network topology is vital for understanding or intervening in a system. Many control and monitoring methods—such as the canonical procedure of state estimation—assume knowledge of topology as a prerequisite.

Grid topologies come in two broad types: radial and meshed. Radial networks have no loops, meaning that there is only one path—defined by a set of lines—connecting any two buses. Meshed networks contain one or more loops, leading to parallel paths between some pairs of buses. Transmission systems, which deliver power at high voltage over long distances,

are organized as meshed networks. The parallel paths create valuable redundancy in case of system failures, endowing resilience to this especially critical portion of the grid. Distribution systems, which deliver power to customers at lower voltages and over shorter distances, are generally operated as radial networks. This simplifies protection design and network management. Through the opening and closing—or switching—of lines, many distribution networks can be shifted between multiple topologies.
Electric network topologies have a natural connection to the mathematical structures called graphs; consequently much theoretical work on topology uses the terminology and definitions of graph theory. In this framework, buses are *nodes* and lines become *edges*. The graph representation of a topology is generally *undirected*, as electric lines have no particular orientation. Graphs of radial topologies are termed *trees*. Mooring electric network topologies to mathematical graphs enables the co-option of a vast, venerable repository of results and techniques from the field of graph theory. These results can be illuminating and empowering in the design of novel

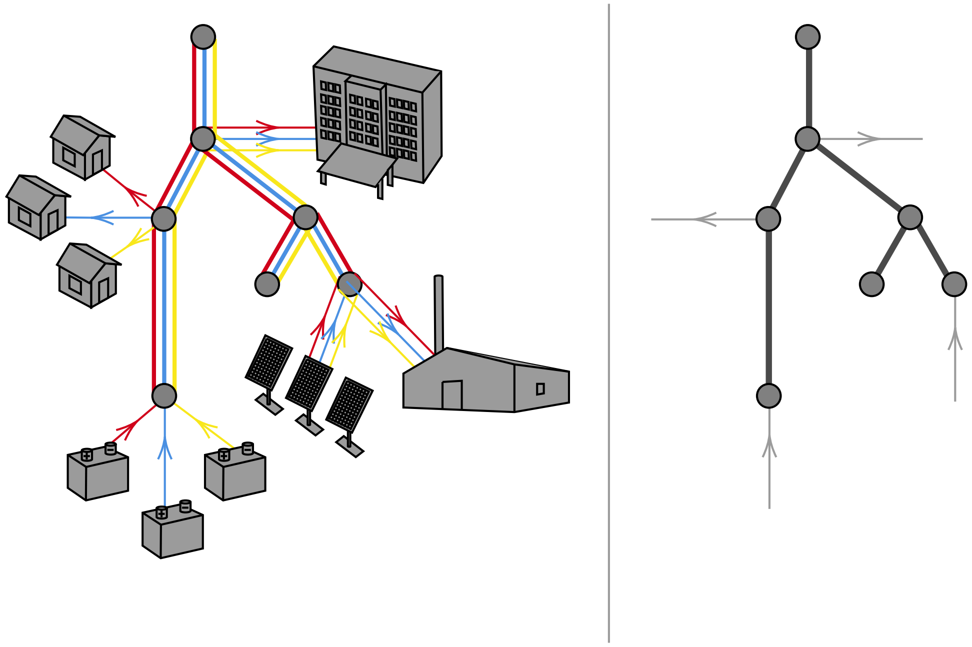
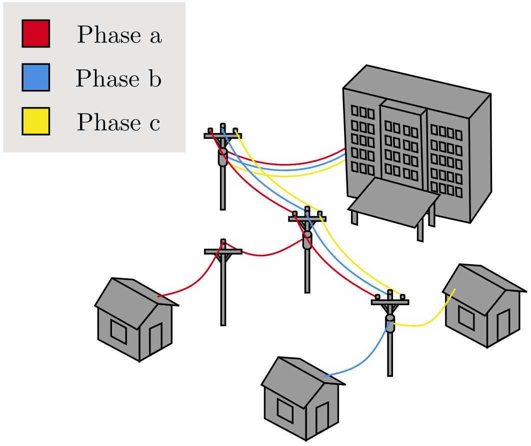
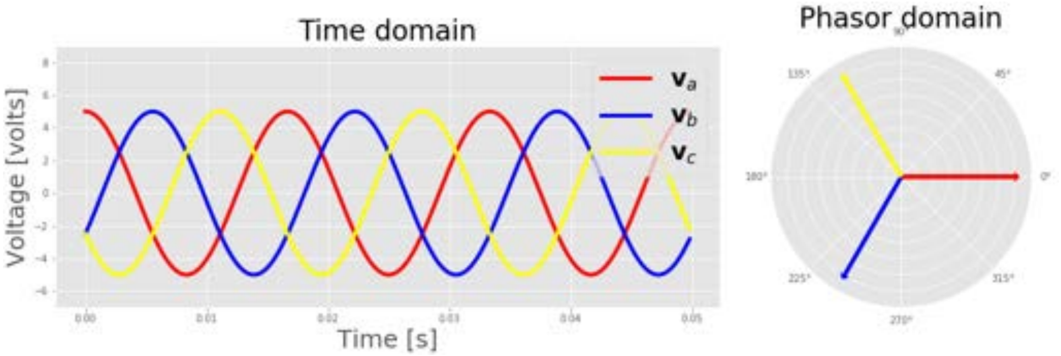


Figure 21: A complete three phase network converted to a single phase graph representation with net flows at each node.



(a) Distribution network with three & one phase loads.



(b) Time domain & phasor voltages on each phase in the perfectly balanced case.

Figure 22: Three phase electric systems

tools for topology estimation and monitoring, as the following sections will highlight.

Interlude: Three Phases

Alternating current electric systems are also termed three phase systems because power is transmitted on three lines—generally named phase a, b, and c—by currents (and correspondingly voltages) that are 120° apart, as visualized in Fig. 22. If the three phase currents are balanced, their sum is zero, eliminating the need for a return wire. Maintaining near balance such that voltages and currents across phases remain symmetrically 120°

apart and have equal magnitudes is an important priority for grid operators. Transmission systems tend to be highly balanced, and here all lines are three phase, coming in sets of three. At the distribution level, the three phases of transmission may be separated to reach different loads, which are generally connected to a single phase only. Therefore, distribution network lines can be one, two, or three phase. Balance is achieved by evenly distributing loads across the three phases.

Although nearly all grids contain multiple phases, systems are often assumed to be balanced with identical impedances and loads on all phases. A balanced three phase system can be reduced to and analyzed as a single phase system, with symmetric behavior (albeit shifted by 120°) on the neglected phases. Many real distribution networks, which contain lines with a subset of all phases, can not be reduced in this way.

A complete network topology should not only specify the connectivity of each line, but also its phase label. However, when treated as balanced and reduced to a single phase equivalent, a network's line phase labels are not handled; the labels are either assumed known or must be determined separately.

Incorrect phase labeling cam be highly problematic, especially if intervention is desired on a particular phase. For example, a fault on phase a may be wrongly localized to phase b due to incorrect labels, causing confusion, time wastage, and potentially safety issues for repair crews. Incorrect phase labels are more common in distribution networks than transmission, due to generally poorer information and visibility on this part of the system.

4.1 TOWARDS USABLE TOOLS FOR TOPOLOGY

This chapter describes several algorithms for topology monitoring. Though they differ in context, data availability, and results, they cohere in their aim to improve real-time topology awareness. Ordered here almost chronologically, they reflect an intellectual progression and academic journey towards *usable* topology tools. Usable topology tools—per the discussion in Chapter 3—possess three features: they deal in physically meaningful quantities, have low or flexible information requirements, and consider the limitations of real measurement data. In the narrative of this chapter, the tools detailed tend toward usability as a preordained destination, but in truth the destination and tool designs coevolved, with the criteria of usability coalescing through the process of tool creation and experimentation.

There is a sizeable literature on algorithms for grid topology estimation. Already extensive when I began working in this area, it has only continued to expand. Algorithms in this literature are highly diverse, using varied measurements and input information and applying assorted techniques to extricate topology. While some methods appear to aim for the usability criteria I define, many have different aims or seem unconcerned about usability altogether. Consequently, of this large literature, only a minority of tools are workable.

The tool features that contribute to poor usability also vary. Many approaches have practically prohibitive demands for prior system information. For example, [67] and [118] require access to simulation models of every possible network configuration; simulated measurements are then compared to real PMU measurements to identify the operational topology. [119] describes a similar approach, using non-PMU measurements. Also requiring models are a class of methods that combine the process of state estimation—estimating voltage phasors at every bus from a set of available measurements—with topology monitoring [46], [120]–[122]. Unfortunately, especially in distribution, network models are often missing or erroneous, and running simulations is time-consuming, thwarting such approaches. Even more inaccessible than methods requiring accurate system models are those which require physical intervention, for example by injecting voltage perturbations into the system [123], [124].

Another set of topology tools—including all those I have worked on and present here—are purely measurement driven, requiring no extra system information or intervention. However, their usability is often hampered by cumbersome measurement requirements or their failure to consider and handle the detrimental impact of real measurement imperfections such as noise. For example, [62], [125]–[128] require PMU measurements at almost every network node, still a remote possibility in most electric grids. On the other hand, several of these approaches are built atop a theoretically justified physics model, which lends certain advantages: success, failure, and sensitivity can be precisely defined and handled. Radial topology estimation forms a special subset of all topology tools, as the constraints of radial structures greatly improve the tractability of this challenging estimation task. [129]–[132] form a single body of work on this problem, with slightly varying approaches. [133], [134] extend these techniques to explicitly handle multiple phases, rather than making the balanced, single phase reduction implicit in most other works.

Taking a different tack are heuristic tools for topology monitoring, which surveil quantities computed from raw measurements that are indicative of underlying network structure. Popular quantities include the rank [135]–[137], pairwise correlations [95], and principal components [93], [94], [138]–[140] of nodal voltage measurements. The usability of heuristic tools benefits from low and flexible measurement coverage requirements. This comes at the cost of fine-grained results, but nevertheless these tools can provide much practically useful system visibility. On the other hand, their usability suffers from the absence of a theoretical basis: obscuring both why they work and any implicit assumptions they make. This in turn hinders precise specification of when and why these tools succeed and fail, limiting user trust.

Section 4.2 describes one such heuristic topology estimation algorithm, vividly illustrating both the benefits and drawbacks of the type. While the algorithm's low measurement coverage demands enable it to be effectively applied to a real distribution network, the absence of a theoretical foundation limits understanding of algorithm failures. Section 4.3 goes on to detail a tool with an underlying physics model and rigorously analyzed noise sensitivity, but exorbitant sensor coverage requirements that make real world application unviable. The tool of 4.4 is an extension to three phase, unbalanced networks; again built on a meticulously defined physics model but with impractical sensor demands. Nevertheless, the theoretical results of these works lay an important foundation for the creation of usable topology tools. Finally, Section 4.5 melds the benefits of the previous tools, using the physics models to derive simple statistical quantities that indicate topology. The resulting *justified heuristic* tools have low or flexible sensor requirements but also a well-defined rationale and transparent assumptions.

4.2 A HEURISTIC APPROACH

This section describes *corr-top*, a heuristic tool for topology identification using μPMU measurements. Very practical, the tool has low, flexible requirements of sensor coverage and demands no prior knowledge of network characteristics. It returns a part of the full network topology; specifically the relative connectivity between measured nodes, which preserves the ordering of sensors along the feeder. Corr-top uses correlations in voltage magnitude measurements to infer topology and works with data on a single phase (though it is likely effective for phase identification as well). To ensure consistency, the algorithm selects and uses specific data segments that are most informative of topology. The alternative, of using arbitrary data segments, produces inconsistent and variable results, the unpredictability of which is an inherent drawback of heuristics. Corr-top relies on signal shapes and trends rather than values, making it resilient to noise, missing data segments, and the persistent transducer error that plagues μPMU data [141]. Corr-top is demonstrated on measurements from an operational distribution feeder.

Though corr-top is not based on an underlying physics model and therefore does not deal in explicit physical quantities, it's simple algorithmic approach preserves a degree of transparency and intuition for the user. In addition to providing a topology estimate, it returns a sample of data that supports the recovered topology. Continuous, high resolution raw voltage magnitude streams contain vast quantities of data that are overwhelmingly difficult for a human to process and interpret. By returning a succinct data example, corr-top distills an interpretable explanation from an overwhelming data set, increasing transparency and trust. This improves the usability of the tool in the context of mixed human-computer control for grid management, which is the broader objective of this thesis. In general, this thesis advocates for building transparency features into grid tools in this manner, especially in the case of algorithms such as heuristics where the lack of an underlying physics model and explicit physical quantities can diminish transparency.

First, some notation for what follows. The voltage magnitude measurement at node i is denoted v_i, referring either to the full, continuous measurement stream or a portion of the stream at this node. A single measurement from this stream is denoted $v_i(k)$ while $v_i(t_1, t_2)$ is a vector of measurements covering the period from time t_1 to t_2. The data streams are discrete, but for simplicity we index them using a time point t or an index k. A matrix is denoted in capitals and its $(i, j)^{th}$ element is denoted with a subscript, for example matrix P and element P_{ij}. $\mathbb{R}$ denotes the set of real numbers.

4.2.1 The Algorithm

The corr-top algorithm includes four steps: data selection, correlation-based proximity computation, spanning tree construction, and data validation. In the first data selection step, a search algorithm selects informative data segments from continuous voltage magnitude measurement streams. Next, correlations are computed between simultaneous segments at spatially distributed sensors to build a *proximity matrix* between sensors. A greedy method uses the proximity matrix to construct an estimate of the feeder connectivity, which is assumed to be radial. In the final step, a subset of the selected data segments is returned as validation of the estimated topology.

Corr-top ingests 120Hz data from μPMU sensors. Synchrophasors are prominent for their ability to report the phase angle of voltages and currents in ac systems. Discussions of their advantages typically revolve around the use of phase angles in various applications, from basic complex power computations to more sophisticated analytics. In contrast, corr-top uses only root mean square (rms) voltage magnitude data from μPMUs, disregarding angles altogether. The compelling, critical value of μPMUs to the algorithm is the measurement resolution and synchronicity across locations. Precise synchronization means events seen at different sensors can be reliably matched in time and confidently claimed to have the same source. Corr-top's topology inference depends on correlation, a common technique for signal matching. Correlations between signals deteriorate rapidly with time misalignment making the precise synchronization of μPMU data vital for this tool. Fig. 23 visualizes an event as measured at two μPMUs, where the similar signatures and time alignment compellingly suggest a common cause. The bottom plot shows how the measurement correlation between sensors for this event rapidly drops as time misalignment grows. The GPS time-stamping used by μPMUs has an accuracy on the order of 10 ns, at which correlations are highly reliable.

One challenge for any high-precision sensor connecting to high- or medium-voltage systems is that instrument transformers—which mediate between the sensor and the system by reducing current and voltages to safe-to-measure levels—can introduce an error that is orders of magnitude greater than the sensor error itself. This limits the field feasibility of analytics requiring precise computation of steady-state quantities such as impedance. Fortunately, transducer errors tend to be stable over timescales of seconds

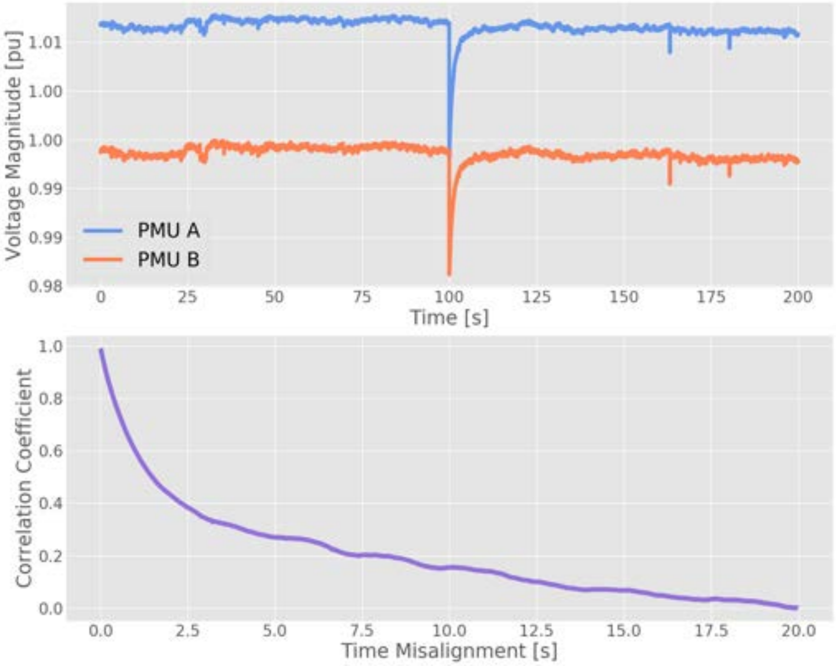


Figure 23: The importance of accurate time synchronization is illustrated by the rapid drop in signal correlation with growing misalignment for an event signature measured at two PMUs.

or minutes. As corr-top uses signal *shapes* rather than raw signal values, it is uncompromised by the nearly constant additive or multiplicative errors in the signal that must be expected.

The signal shape segments used in corr-top are found through an event search algorithm, described next.

4.2.2 Data Selection with Event Search

Due to the high frequency of µPMU measurements, it is infeasible to use an entire stream of µPMU data at full resolution over considerable time periods such as multiple days. Instead, selected windows of data must be used, raising the question of what kind of data segments to choose. Voltage magnitude signals in the distribution grid are distorted by network happenings and load fluctuations throughout the electric grid, near and far. Therefore, though voltage magnitudes are carefully regulated to stay within a small band around the nominal voltage level, there is essentially

constant ambient variation within this band, which may or may not correlate across locations. Experimentally, I found that short, high resolution data windows at arbitrary time points show uniformly low correlations across sensors. In contrast, longer, lower resolution windows show uniformly high correlations across sensors, since average voltage magnitude profiles track each other closely across a single distribution network. This phenomenon is known as the low-rank nature of synchrophasor measurements and is discussed in [142]. Overall, using arbitrary measurement segments of high or low resolution leads to inconsistent topology estimates. Instead, corr-top uses data during periods where voltage magnitudes deviate rapidly and significantly from the nominal, which were experimentally found to be most informative of topology. I refer to these periods as *event points*. Typical event points observed in µPMU voltage magnitude data include transformer tap changes, capacitor bank switching, and voltage sags. The source or nature of the event is not important for the purpose of topology identification (event classification is discussed further in Chapter 5).
As described in Section 2.3.1, many events can be rapidly localized by leveraging the summary statistics stored in the multi-resolution structure of the Berkeley Tree Database (BTrDB). Corr-top makes use of a simple event search algorithm presented in [79]. The algorithm scans through non-overlapping, fixed duration time windows of voltage magnitude data, computing a metric on the summary statistics of each window. If the metric exceeds a preset threshold—denoted τ—the window is considered to contain an event and its raw data is queried and returned. The metric used is given in Eq. (7).

$$\frac{\max[v(t_k, t_k + \delta)] - \min[v(t_k, t_k + \delta)]}{\mathrm{mean}[v(t_k, t_k + delta)]} > \tau \tag{7}$$

Here, $t_k, t_k + \delta$ are respectively the start and end time points of the k^{th} window with duration δ. By accessing only summary statistics to localize events, the event search algorithm runs in seconds across an entire day of µPMU data on my Intel i7 Core processor.

4.2.3 Proximity Matrix Construction

Consider a network with n sensors, each monitoring a common phase at separate network nodes. The voltage magnitude time series for these sensors are $v_1, v_2, ..., v_n$. Corr-top runs event search on each of these n time

A 1: $[E]$ = event-extraction($pmus$, δ)

```
Input  : $pmus$ is a length $n$ list of pmus
         $\delta$ is a constant event duration
Output : $E$ is a matrix of event data.
$n \leftarrow$ len($pmus$) // Number of pmus
for $i \leftarrow 0$ to $m$ do
    $t_{starts} \leftarrow$ find-events($pmus[i]$) // event start times
    $e_i \leftarrow$ len($t_{starts}$) // number of events found at pmu $i$
    // Iterate over events
    for $j \leftarrow 0$ to $n$ do
        for $k \leftarrow 0$ to $e_i$ do
            // Get data at pmu $j$ for event $k$
            $E[i][j][k] \leftarrow$ get-data($pmus[j]$, $t_{start}[k]$, $t_{start}[k] + \delta$)
        end
    end
end
```

series. For every event found in sensor i's data stream v_i, data is extracted for the corresponding time period from the measurement stream of every other sensor. Assuming e_1 events are found in v_1, e_2 in v_2 and so on, the result is a total of $e = n \sum_{i=1}^{n} e_i$ extracted time segments. Usually, there is a great deal of overlap between the events found at each sensor. However, it is necessary to run event search on each time series as some events are detected at one sensor but fall below the detection threshold at another sensor. It is precisely this variation in event visibility across sensors that contains important topology information. Algorithm 1 defines the event extraction process.

The extracted event data is used to build up a proximity matrix $P \in \mathbb{R}^{n \times n}$. The value P_{ij} is set to the average Pearson correlation coefficient between events detected at sensor i and the corresponding measurements at sensor j. The Pearson correlation coefficient between two t-length voltage magnitude time series $v_i \in \mathbb{R}^t$ and $v_j \in \mathbb{R}^t$, denoted $\rho(v_i, v_j)$, is defined in Eq. (8).

$$\rho(v_i, v_j) = \sum_{k=1}^{t} \bar{v}_i(k)\bar{v}_j(k)||\bar{v}_i||_2||\bar{v}_j||_2 \tag{8}$$

$$\bar{v}_i = \frac{v_i - \mathbb{E}[v_i]}{||v_i - \mathbb{E}[v_i]||_2}, \ \mathbb{E}[v_i] = \frac{1}{t}\sum_{k=1}^{t} v_i(k)$$

Note the normalization in the definition of $\rho(v_i, v_j)$. Normalization of the voltage data removes the effect of the nominal voltage magnitude, which may vary across buses within a single feeder. More generally, normalization means the correlation coefficient reflects how closely the shapes, rather than the raw values, of signals v_i and v_j align. Normalization also makes the metric ρ more interpretable than a simple dot product, because $\rho(v_i, v_j)$ is bounded between -1 and 1, with 1 indicating perfectly matched signal shapes.

Denote the start and end time stamps of the e_i events found at sensor i by $(t_1^{(i)}, t_1^{(i)} + \delta), (t_2^{(i)}, t_2^{(i)} + \delta), ..., (t_{e_i}^{(i)}, t_{e_i}^{(i)} + \delta)$. Then, P_{ij} is computed as follows:

$$p_{ij} \triangleq \frac{1}{e_i}\sum_{k=1}^{e_i} \rho\Big(v_i(t_k^{(i)}, t_k^{(i)} + \delta), v_j(t_k^{(i)}, t_k^{(i)} + \delta)\Big). \tag{9}$$

$$P_{ij} = \min(p_{ij}, p_{ji}) \tag{10}$$

Since P_{ij} is the average of correlation coefficients, it lies in the range $[-1, 1]$. p_{ij} indicates the proximity of sensors i and j based on the events detected at i while p_{ji} indicates the proximity of sensors i and j based on the events detected at j. A higher score indicates greater proximity. By setting P_{ij} to the minimum of p_{ij} and p_{ji}, corr-top uses the more pessimistic estimate of proximity. This tends to emphasize local events which contain the most useful topology information. The final proximity matrix is symmetric: $P = P^T$. Fig. 24 visualizes the construction of P.

4.2.4 Topology Estimate

Corr-top is targeted for application in distribution networks and assumes that the topology to be recovered is radial—in graph terms, tree structured.

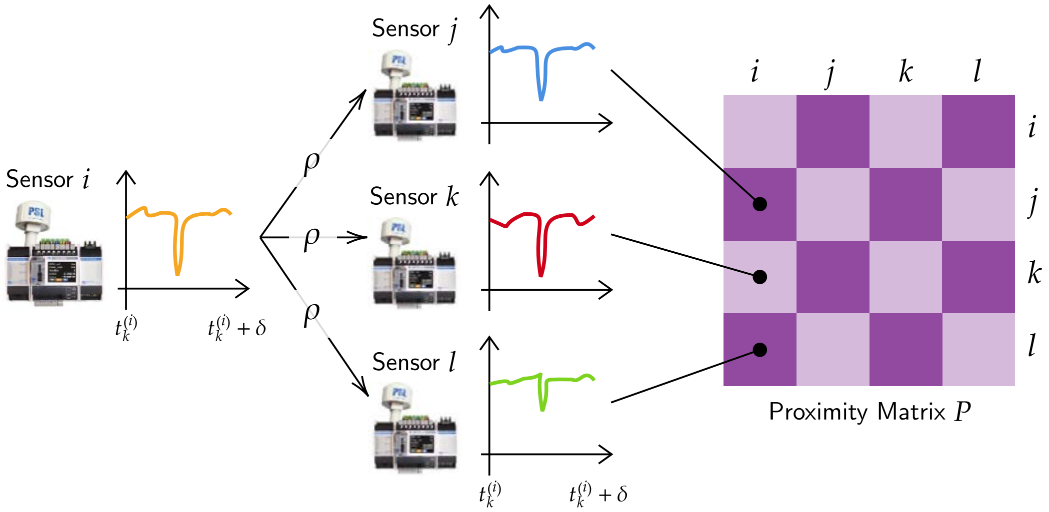
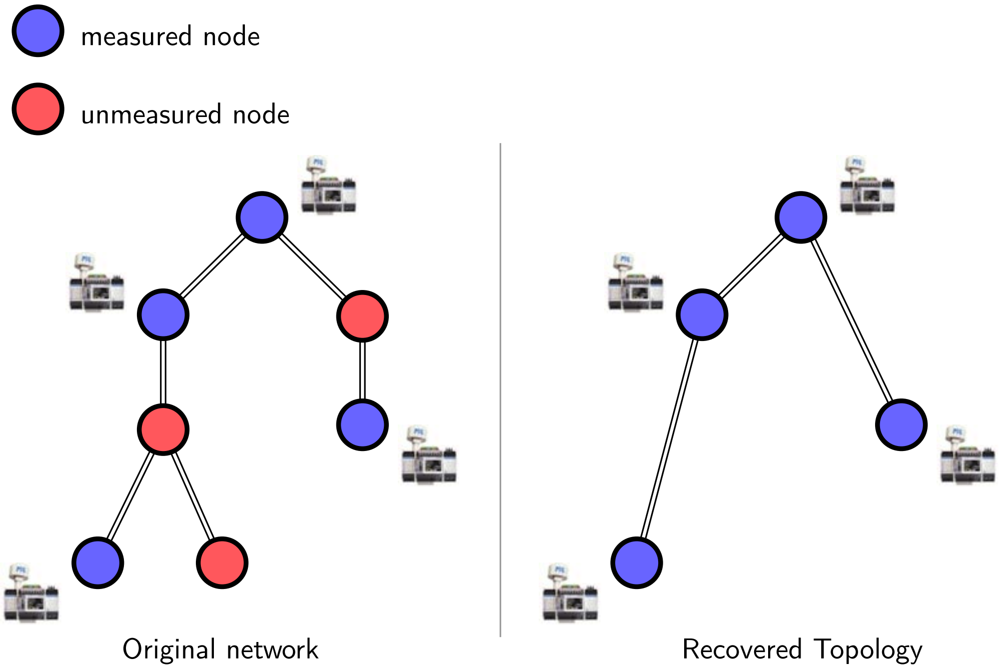


Figure 24: Adding values to the $n \times n$ proximity matrix P for single event k found at sensor i.

To estimate the tree connectivity of the measured network nodes from the computed proximity matrix, P is passed to a *maximal spanning tree* (mst) algorithm. The mst algorithm, homologous to the minimal spanning tree algorithm, is a classic greedy algorithm [143]. The mst algorithm interprets P as the weighted adjacency matrix of a graph, where the n sensors are the n nodes and P_{ij} is the weight of the edge connecting node i to j. From the full set of $n \cdot (n-1)$ edges, the algorithm selects a subset of $n-1$ edges that connect the n graph nodes together in a tree while having the maximum sum of edge weights. This maximal spanning tree is returned as the estimated topology.

Corr-top only handles measured nodes. Therefore, edges in the estimated topology are relative, reflecting how the monitored nodes are connected to each other within the full, real network (Fig. 25). Unmeasured nodes are not considered in the algorithm and do not appear in the final solution. Therefore, there are combined topology and sensor deployment scenarios where different topologies result in the same relative sensor connectivity and will therefore be indistinguishable to corr-top.

While corr-top aims to work under various sensor placement schemes, some placements produce better results than others. For example, uniform spacing of sensors along the feeder length is better than “clumped” placement. Consider the case where we have three sensors s_1, s_2, s_3 such that



Original network

Recovered Topology

Figure 25: A comparison of the true network topology and the estimate returned by corr-top showing recovery of the relative connectivity of measured nodes.

the distance from s_1 to s_2, denoted $\mathtt{dist}(s_1, s_2)$, is much less than that from s_1 to s_3: $\mathtt{dist}(s_1, s_2) << \mathtt{dist}(s_1, s_3)$ and s_3 lies down the network from s_1 and s_2. Measurements at s_1 and s_2 are highly correlated, so $v_{s_1} \approx v_{s_2}$. Therefore $\rho(v_{s_1}, v_{s_3}) \approx \rho(v_{s_2}, v_{s_3})$, and the data provides little indication if s_3 is connected directly to s_1 or s_2. Note that if the network has specific switches or breakers whose open/closed status is of interest, it should generally be straightforward to identify suitable μPMU placements for reliable detection.

With slight modification, corr-top can be extended to cases where the n sensors are split between two or more radial feeders by modifying the algorithm to return more than one tree when edge weights fall short of a threshold. In such a case, the sensors on different feeders will be separated by a single, low weight edge. This scenario is detailed in Section 4.2.6.

4.2.5 Data Validation

Along with the topology estimate, corr-top returns selected data to validate the estimate for the user. This data quantity must be kept small to ensure it is interpretable to a human user. Corr-top returns the voltage magnitude time series for one event at each sensor–a total of n time series segments. Returning a single data snippet for every sensor is highly informative, enabling the user to "see" the topology from the vantage of each sensor node. This is especially important when the network is configured as multiple separate radial feeders.

At a single sensor, for the selected event, a $n \times 1$ vector of correlation coefficients is also returned, the computation of which was described earlier in Section 4.2.3. More concretely, suppose at sensor s there are a total of e_s event points extracted from the voltage measurement stream v_s and used to construct proximity matrix P. The e_s event points correspond to the time segments $(t_1^{(s)}, t_1^{(s)} + \delta), ..., (t_{e_s}^{(s)}, t_{e_s}^{(s)} + \delta)$. Only one of these events is chosen to be part of the returned validation data. The selected event is the one which maximizes the following objective, for $k \in \{1, ..., e_s\}$:

$$k_s^* = \max_k \operatorname{var}_i \left[\rho \left(v_s(t_k^{(s)}, t_k^{(s)} + \delta), v_i(t_k^{(s)}, t_k^{(s)} + \delta) \right) \right]. \tag{11}$$

Maximizing this objective finds the event with the largest variance in correlations across all other sensors $i \in \{1, ..., N\}$. This selected event, which is maximally different across sensors, tends to be most explanatory of the topology. By capturing the network topology and validating the topology estimate, the selected event makes the topology estimation algorithm more transparent, trustworthy and hence useful to human users.

4.2.6 Results

Corr-top was tested on data from two operational distribution feeders. The topology estimates returned by the algorithm were validated against ground truth information provided by the utility. The first network, termed network a, operated under a fixed topology and the algorithm was tested for consistency at this location. The second, network b, underwent reconfiguration multiple times. Table 1 contains the details of each test case. The size and complexity of the test cases presented here were constrained by the networks and reconfiguration operations for which data was available.

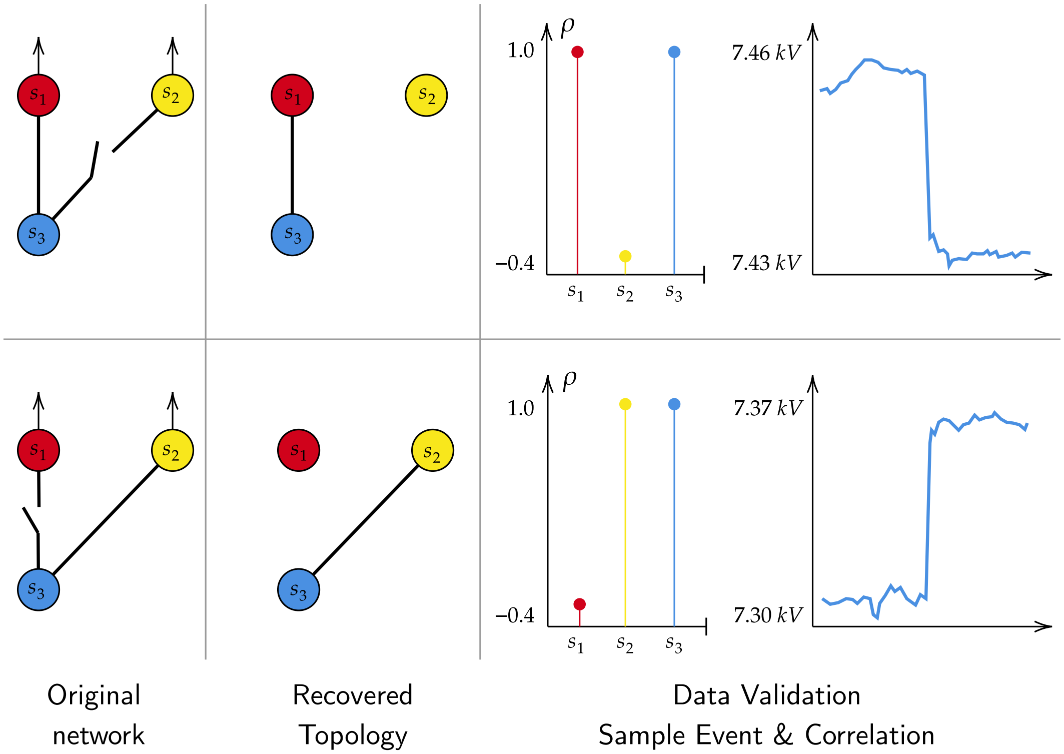


Figure 26: The true topology, topology estimate and data explanation from s_3 for test cases 4 (top) and 5 (bottom)

The number of sensors used varies across test cases on network b as data was available for only a subset of all sensors during each test period. The test case duration must be long enough to generate a sufficient number of events for topology estimation. On network a, we used a longer 48 hour duration since the network topology was fixed. On network b we generally used a 24 hour duration. Test case 7 is unusually short and corresponds to a brief period during which network b operated under an alternate topology. In this instance the time duration proved adequate for correctly recovering the topology.

To provide intuition for corr-top's algorithmic performance, this section describes in detail two of the test cases on network b: cases 4 and 5 in Table 1. Network b generally operates as two separate radial feeders. Tie switches allow nodes to be switched from one feeder to the other. Consequently, for network b test cases, the topology estimation algorithm was allowed to return a solution consisting of multiple trees. A threshold of 70% was used to split the topology estimate. That is, if the minimum weight edge in the mst solution had a weight less than 70% of the next lightest edge, the minimum weight edge was removed, splitting the spanning tree solution into two spanning trees. This threshold was chosen empirically.

Cases 4 and 5 use the same set of three sensors, denoted s_1, s_2, and s_3. The true connectivities of these sensors is shown in the left panel of Fig. 26. Measurement nodes s_1 and s_2 are on different feeders. Node s_3 can be connected to one substation through node s_1, as in test case 4, or to the other substation through node s_2, as in test case 5. The middle panel of Fig. 26 presents the sensor connectivity estimated by corr-top. In both cases the estimated connectivity is correct.

The right panel in Figure 26 contains the returned validation data for sensor s_3. In both cases, the returned validation measurement is a tap change, which manifests as a step change in voltage magnitude. In case 4 the tap change is clearly visible at s_1 and s_3 but invisible at s_2. In case 5, the tap change is visible at s_2 and s_3 but invisible at s_1. This returned data snippet is a compelling validation of the topology estimate.

* * *

Corr-top is a heuristic approach to topology estimation and benefits from low and flexible sensor coverage requirements as well as a level of algorithmic transparency and interpretability. However, like many heuristic approaches, the absence of a theoretical foundation for the algorithm makes it difficult to understand and predict algorithm failure or inconsistency. In

Table 1: Corr-top test case summary

Test Case	Network	Number of sensors	Duration	Correct?
1	a	4	48 hr	Yes
2	a	4	48 hr	Yes
3	a	4	48 hr	Yes
4	b	3	24 hr	Yes
5	b	3	24 hr	Yes
6	b	4	24 hr	Yes
7	b	4	5 hr 48 min	Yes

the case of corr-top, this was mitigated through the averaging of many data samples when constructing proximity matrix P; however this limits the real-time applicability of the approach. In the search for usable tools for topology monitoring, we need to build a tractable, underlying physics model, which is the focus of the next section.

4.3 A PHYSICS APPROACH

This section approaches the problem of topology estimation starting with fundamentals: the physics equations that relate phasor measurements and topology. It describes an algorithm for estimating the complete network topology—encompassing both lines and their impedances—from nodal voltage and current phasors. Perhaps more importantly, it explores the theoretical foundations of topology estimation, establishing informational limits on how much of the topology can be inferred from data and the noise sensitivity of estimation.

The proposed estimation method requires no apriori information about the network structure or user guidance—making it easier to use, widely applicable, and resilient to human error. It makes no assumptions on load behavior or network parameters. It addresses the network as a whole and specifically improves noise robustness over prior work by formulating the problem to be better conditioned. Here, impedances are estimated via ordinary least squares (ols) regression but the method is agnostic to the optimization method used for regression.

The topology estimation method requires nodal voltage and current injec-

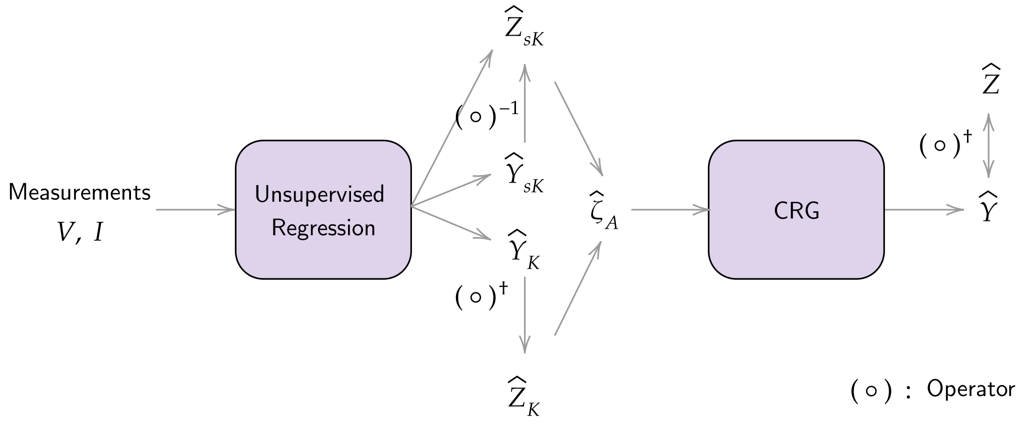


Figure 27: Impedance and topology estimation algorithm schematic. Measurements V and I are fed into estimation algorithms to recover various network representations.

tion phasor measurements at all active network nodes—anywhere where current enters or leaves the network—to estimate what is called a Kron reduced network impedance or admittance model. Next, effective impedances quantities between active nodes—which capture cumulative system impedance—are extracted from the Kron reduced model. We develop Complex Recursive Grouping (crg) to recover network topology from the complex-valued effective impedance estimates, in the case where the network is radial. The crg algorithm is an extension of the recursive grouping algorithm, originating in the field of evolutionary biology [144], and previously applied to topology estimation in the case of real-valued impedances [145]. Fig. 27 shows a schematic representation of the proposed topology estimation method; it is a useful reference for tracking the processes, quantities, and notations that will be introduced throughout this section.

For the toy network in Fig. 28 the proposed method proceeds as follows. First, unsupervised regression is used to estimate the admittance or impedance matrix of the Kron reduced network, visualized on the right of Fig. 28. The specific regression method can be ols—as done here (e.g. Eq. (19))—or another optimization formulation. Next, effective impedance estimates between active nodes are calculated using Eq. (13). Finally, the full network is reconstructed using the crg algorithm applied to the set of effective impedances, as illustrated in Fig. 30. The four primary contributions of this section are:

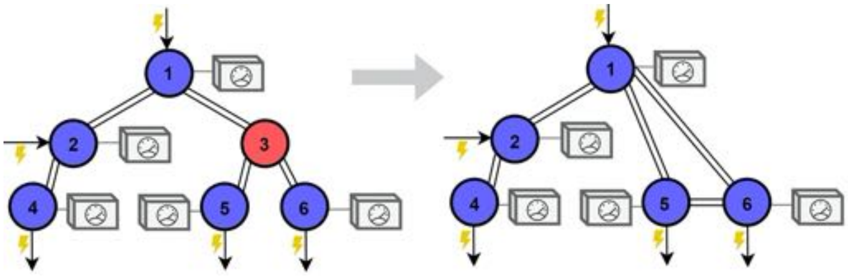


Figure 28: Kron reduction of a toy network with 5 active nodes (blue), including the substation, and 1 passive node (red) which does not have any current injection.

1. Defining the fundamental information limits of (phasor) measurements made on electrical networks.
2. Proving the Kron reduced admittance matrix to be the best achievable outcome of unsupervised learning on electrical network measurements.
3. Proposing estimation of the reduced Laplacian (*subKron*) matrix network representation to improve impedance estimation noise-robustness.
4. Generalization of the Recursive Grouping algorithm to complex-valued distances to recover radial topologies from effective impedances.

The subsequent section parts are organized as follows. Part 4.3.1 provides necessary background information. Section 4.3.2 establishes the challenges of network estimation. Section 4.3.3 introduces the subKron network model and section 4.3.4 explains its physical meaning. Section 4.3.5 discusses noise-robust unsupervised impedance estimation. Section 4.3.6 presents the crg algorithm. Finally, section 4.2.6 evaluates the proposed methods in simulation.

4.3.1 Background

Notation and Definitions

First, let us take care of vital preliminaries: the notation and mathematical definitions used in this section, which are mostly in line with those introduced in Section 1.4. $\mathbf{M} \in \mathbb{C}^{p \times q}$ is a complex-valued p-by-q matrix with

i^{th} row $\mathbf{M}_i^T$ and element $\mathbf{M}(l, m)$ in row l, column m. $\mathbf{M}^T$ and $\mathbf{M}^H$ are its transpose and conjugate transpose respectively. $^{-1}$ and † denote the matrix inverse and pseudoinverse respectively. $\mathcal{Q}$ is a set with cardinality $q = |\mathcal{Q}|$. $\mathbf{I}$ denotes complex current measurements—*not* the identity matrix. $\mathbb{1}$ is the 1's vector and e_i is the i^{th} standard basis vector. Operators $\mathrm{Re}(\bullet)$ and $\mathrm{Im}(\bullet)$ respectively return the real and imaginary parts of their arguments. $\ker(\mathbf{M})$ is the span of vectors $\{v\}$ such that $\mathbf{M}v = 0$. $1j = \sqrt{-1}$. Matrices A and B are *informationally equivalent* if we can exactly compute A from B and vice versa.

A graph $\mathcal{G}$ has nodes $\mathcal{N}$ and edges $\mathcal{E}$. If $\epsilon_{ij} \in \mathcal{E}$, nodes i and j are connected by an edge with weight $\mathbf{w}_{ij}$. The degree of node i in $\mathcal{G}$ is the number of nodes to which it is directly connected. A node with degree $= 1$ is a *leaf*.

A connected graph has a path (a sequences of edges) between every pair of nodes. In an acyclic graph, this path is unique. As mentioned before, a *tree* is a connected and acyclic graph, termed *radial* in power systems. To orient a tree, one node is chosen as the *root*. Node i and j are respectively *parent* and *child* if $\epsilon_{ij} \in \mathcal{E}$ and i is closer to the root. Nodes k and k' are *siblings* if they share a parent.

We consider an electrical network with $n = a + p$ nodes and e edges, where n, a, p and e are respectively the cardinality of the set of network nodes $\mathcal{N}$, the set of active nodes, $\mathcal{A}$, the set of passive nodes, $\mathcal{P}$, and the set of edges $\mathcal{E}$. Active nodes are the subset of network nodes where current enters or exits the network. In distribution networks, the substation is an active node that is also a convenient choice for the root. Passive nodes are the subset of network nodes where three or more lines come together and current is rerouted, but does not enter or exit the network. We assume voltage and current phasor measurements are available at *at least* all active nodes. These definitions are visualized in Fig. 28.

Complex-valued, synchronized voltage and current phasor measurements from all n nodes at t time points can be collected into matrices $\mathbf{V} \in \mathbb{C}^{n \times t}$ and $\mathbf{I} \in \mathbb{C}^{n \times t}$. Every row of $\mathbf{V}$ and $\mathbf{I}$ is a measurement time series at one node. Ohm's law defines a linear relationship between $\mathbf{V}$ and $\mathbf{I}$:

$$\text{[Admittance] } \mathbf{I} = \mathbf{YV} \leftrightarrow \mathbf{V} = \mathbf{ZI} \text{ [Impedance]} \tag{12}$$

The two forms are parametrized by either the network *admittance* matrix, $\mathbf{Y} \in \mathbb{C}^{n \times n}$, or the network *impedance* matrix, $\mathbf{Z} \in \mathbb{C}^{n \times n}$.The forms are informationally equivalent, however $\mathbf{Y}$ has a simpler, more intuitive structure and is generally used in practice. In graph theory terms, $\mathbf{Y}$ is the *weighted*

Laplacian of the graph describing the electrical network, with edges corresponding to physical lines and weights corresponding to line admittances. For a network with no shunt admittances—lines connecting nodes directly to ground—$\mathbf{Y}$ is a *loopless* weighted Laplacian [146]. Physically, $\mathbf{Z}$ is the inverse of $\mathbf{Y}$, however it cannot be computed as $\mathbf{Y}^{-1}$ due to the following well-known lemma [147]:

Lemma 1. *Loopless Laplacian* $\mathbf{Y}$ *of a connected graph is rank* $(n-1)$ *with* $\mathbf{1} \in \ker(\mathbf{Y})$

However, we can calculate $\mathbf{Z}$ from $\mathbf{Y}$ with the matrix pseudoinverse: $\mathbf{Z} = \mathbf{Y}^{\dagger}$. The pseudoinverse is defined via the singular value decomposition, and inverts all non-zero eigenvalues of the original matrix while preserving its null space.

We neglect shunt admittances in this work. At active nodes, shunt admittances are irrelevant as both voltage and current measurements are available.

Effective Impedance

In [146], the effective resistance r_{ij} of a resistive network is defined as the voltage drop between nodes i and j when 1 amp of current is injected at i and extracted at j. This is a fundamental quantity for parametrizing network behavior. Similarly, we define *effective impedance* $\boldsymbol{\zeta}_{ij}$ as the complex-valued voltage phasor difference between nodes i and j of a resistive+reactive network when $(1+0j)$ amp of fundamental frequency current is injected at i and extracted at j. $\boldsymbol{\zeta}_{ij}$ is computed from $\mathbf{Z}$ as:

$$\boldsymbol{\zeta}_{ij} = (e_i - e_j)^T \mathbf{Z} (e_i - e_j) \tag{13}$$

Effective impedance is symmetric: $\boldsymbol{\zeta}_{ij} = \boldsymbol{\zeta}_{ji}$. All pairwise effective impedances can be collected into matrix $\boldsymbol{\zeta} \in \mathbb{C}^{n\times n}$ such that $\boldsymbol{\zeta}(i,j) = \boldsymbol{\zeta}_{ij}$. $\boldsymbol{\zeta}_{\mathcal{A}} \in \mathbb{C}^{a\times a}$ is the collection of effective impedances between active nodes. We use subscripts rather than parentheses with $\boldsymbol{\zeta}$ to emphasize that $\boldsymbol{\zeta}$, unlike $\mathbf{Z}$, isn't structured in a useful manner. The entries of $\mathbf{Z}$ can be written in terms of $\boldsymbol{\zeta}$ as follows:

$$\mathbf{Z}(i,j) = -\frac{1}{2}\left(\boldsymbol{\zeta}_{ij} - \frac{1}{n}\sum_{k=1}^{n}(\boldsymbol{\zeta}_{ik} + \boldsymbol{\zeta}_{kj}) + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\boldsymbol{\zeta}_{kl}\right) \tag{14}$$

In radial electric networks, effective impedance are a valid distance metric, as impedances add in series, making them useful for distance-based latent tree reconstruction algorithms.

Kron Reduction

While $\mathbf{I} = \mathbf{Y}\mathbf{V}$ relates voltages and currents at *all* nodes, the Kron reduction reduces this equation to relate voltages and currents at active nodes *only* [146], [148]. (In general, the Kron reduction can eliminate any subset of nodes $\mathcal{U} \subseteq \mathcal{P}$. We assume $\mathcal{U} = \mathcal{P}$.) The Kron reduction is derived from Ohm's law using the fact that $\mathbf{I}_i^\mathsf{T} = \mathbf{0}$, $\forall i \in \mathcal{P}$. Without loss of generality, we partition the complete $\mathbf{V}$ and $\mathbf{I}$ matrices into active and passive nodal measurements, plugging in $\mathbf{I}_\mathcal{P} = 0$:

$$
\begin{aligned}
&\begin{bmatrix} \mathbf{I}_\mathcal{A} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{\mathcal{A}\mathcal{A}} & \mathbf{Y}_{\mathcal{P}\mathcal{A}}^\mathsf{T} \\ \mathbf{Y}_{\mathcal{P}\mathcal{A}} & \mathbf{Y}_{\mathcal{P}\mathcal{P}} \end{bmatrix} \begin{bmatrix} \mathbf{V}_\mathcal{A} \\ \mathbf{V}_\mathcal{P} \end{bmatrix} \leftrightarrow \begin{bmatrix} \mathbf{V}_\mathcal{A} \\ \mathbf{V}_\mathcal{P} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{\mathcal{A}\mathcal{A}} & \mathbf{Z}_{\mathcal{P}\mathcal{A}}^\mathsf{T} \\ \mathbf{Z}_{\mathcal{P}\mathcal{A}} & \mathbf{Z}_{\mathcal{P}\mathcal{P}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_\mathcal{A} \\ 0 \end{bmatrix} \\
&\mathbf{I}_\mathcal{A} = (\mathbf{Y}_{\mathcal{A}\mathcal{A}} - \mathbf{Y}_{\mathcal{P}\mathcal{A}}\mathbf{Y}_{\mathcal{P}\mathcal{P}}^{-1}\mathbf{Y}_{\mathcal{P}\mathcal{A}}^\mathsf{T})\mathbf{V}_\mathcal{A} \leftrightarrow \mathbf{V}_\mathcal{A} = \mathbf{Z}_{\mathcal{A}\mathcal{A}}\mathbf{I}_\mathcal{A} \\
&\mathbf{Y}_\mathsf{K} \triangleq \mathbf{Y}_{\mathcal{A}\mathcal{A}} - \mathbf{Y}_{\mathcal{P}\mathcal{A}}\mathbf{Y}_{\mathcal{P}\mathcal{P}}^{-1}\mathbf{Y}_{\mathcal{P}\mathcal{A}}^\mathsf{T} \leftrightarrow \mathbf{Z}_\mathsf{K} \triangleq \mathbf{Z}_{\mathcal{A}\mathcal{A}}
\end{aligned}
$$

The Kron reduced admittance matrix, $\mathbf{Y}_\mathsf{K} \in \mathbb{C}^{a \times a}$, is the Schur complement of $\mathbf{Y} \in \mathbb{C}^{n \times n}$, with respect to the passive node set $\mathcal{P}$. For Laplacian matrices, the Schur complement is the cumulative result of successive Gaussian eliminations of every passive node. The Kron reduced impedance matrix, $\mathbf{Z}_\mathsf{K} \in \mathbb{C}^{a \times a}$, is the $\mathbf{Z}_{AA}$ block of $\mathbf{Z} \in \mathbb{C}^{n \times n}$. $\mathbf{Y}_\mathsf{K}$, like $\mathbf{Y}$, is an admittance matrix and is therefore rank deficient. Consequently, $\mathbf{Z}_\mathsf{K} = \mathbf{Y}_\mathsf{K}^\dagger$ is an impedance matrix.

Physically, Kron reduction eliminates passive nodes, replacing the original network connections with a set of fictitious connections that preserve effective impedances between active nodes:

$$\forall i, j \in \mathcal{A} : \boldsymbol{\zeta}_{ij} = (e_i - e_j)^\mathsf{T}\mathbf{Z}_\mathsf{K}(e_i - e_j).$$

The set of pairwise effective impedances between active nodes are denoted $\boldsymbol{\zeta}_\mathcal{A}$. Therefore, the Kron reduction preserves the "input-output" behavior—the relationship between voltages and currents at active nodes—of the complete network.

There are many benefits to the Kron representation. It relates measurements at only the network active nodes. Practically, this makes the Kron representation useful when sensor installations or measurement availability is limited. However, the Kron reduction Y_K is still rank deficient.

Lemma 2. Y_K *is rank deficient with* $\mathbb{1} \in \ker(Y_K)$.

Proof. Since $\mathbf{Y}$ is a loopless Laplacian matrix, $\mathbf{Y}_K$ is also a loopless Laplacian [146]. Therefore, $\mathbf{Y}_K \mathbb{1} = 0$. □

4.3.2 Fundamental limitations of Electrical Network Data Sets

In general, $\mathbf{Y}$ is not recoverable from $\mathbf{V}$ and $\mathbf{I}$ measurements without prior knowledge of the network topology, even with full measurement coverage [61]. This is due to fundamental limits—established in Thm. 1—on the rank of $\mathbf{I} \in \mathbb{C}^{m \times t}$ and $\mathbf{V} \in \mathbb{C}^{m \times T}$ from arbitrary network node subset $\mathcal{M}$. Intuitively, a data matrix's rank upper-bounds the information it contains: informationally equivalent matrices must have equal ranks.

Theorem 1. ***(A)*** *For any* $\mathbf{V} \in \mathbb{C}^{m \times t}$ *matrix constructed from time series measurements of voltage phasors at all nodes in the set* $\mathcal{M}$ *such that* $\mathcal{M} \subseteq \mathcal{N}$*:* $\text{rank}(\mathbf{V}) \leqslant a$ ***(B)*** *For any* $\mathbf{V} \in \mathbb{C}^{m \times t}$ *matrix with* $t \geqslant a$ *constructed from time series measurements at a subset of nodes* $\mathcal{M}$ *such that* $\mathcal{A} \subseteq \mathcal{M} \subseteq \mathcal{N}$*:* $\text{rank}(\mathbf{V}) = a$

Proof. By Ohm's Law $\mathbf{I} = \bar{\mathbf{Y}}\mathbf{V}$ and by Kirchoff's current law, $\text{rank}(I) \leqslant a - 1$. If $m = n$, $\bar{\mathbf{Y}}$ is the complete network admittance. If $m < n$, $\bar{\mathbf{Y}}$ is the Kron reduction of $\mathbf{Y}$ with respect to nodes $\mathcal{M}$. From lemma 1, $\text{rank}(\bar{\mathbf{Y}}) = m - 1$. By Sylvester's rank inequality, $\text{rank}(\mathbf{I}) \geqslant \text{rank}(\mathbf{V}) + \text{rank}(\mathbf{Y}) - m \implies \text{rank}(\mathbf{V}) \leqslant a$, proving Part A. Now consider $\mathcal{A} \subseteq \mathcal{M} \subseteq \mathcal{N}$. By the definition of a graph Laplacian, $\dim(\ker(\bar{\mathbf{Y}})) = 1$, with $\ker(\bar{\mathbf{Y}}) = \mathbb{1}$. However, assuming positive voltage measurements, $\mathbb{1}$ is not in the span of $\ker(\mathbf{V})$. Therefore the kernel dimensions of the matrix product $\mathbf{I} = \mathbf{Y}\mathbf{V}$ are additive—$\dim(\ker(\mathbf{I})) = \dim(\ker(\mathbf{Y})) + \dim(\ker(\mathbf{V}))$—and Sylvester's inequality holds with equality: $\text{rank}(\mathbf{I}) + 1 = \text{rank}(\mathbf{V})$. Assuming some load variation, the bound on the rank of $\mathbf{I}$ holds with equality: $\text{rank}(\mathbf{I}) = a - 1$. Thus, $\text{rank}(\mathbf{V}) = a$. □

Thm.1 implies that $\mathbf{Y}$ cannot be estimated from $\mathbf{V}$ and $\mathbf{I}$ without prior knowledge of the network. That is, the information to uniquely specify $\mathbf{Y}$ does not exist in $\mathbf{V}$ and $\mathbf{I}$ alone. Consider when $\mathbf{V}$ is not full rank (when the network contains at least one passive node). Then, $\exists y \in \ker(\mathbf{V}^T)$

such that: $\mathbf{I} = \mathbf{YV} = (\mathbf{Y} + \mathbf{1}y^{\mathsf{T}})\mathbf{V}$. Similarly, $\exists z \in \ker(\mathbf{V}^{\mathsf{T}})$ such that: $\mathbf{V} = \mathbf{ZI} = (\mathbf{Z} + \mathbf{1}z^{\mathsf{T}})\mathbf{I}$. Thus $\mathbf{Y}$ and $\mathbf{Z}$ are not uniquely recoverable, as there are multiple possibilities for these matrices that are consistent with the measurements. Therefore, for a network with any passive nodes, unsupervised $\mathbf{Y}$ estimation with zero prior information is generally impossible. Yet, $\mathbf{Y}$ contains *two* distinct pieces of information: network connectivity and effective impedances. While Thm. 1(A) says it is generally impossible to estimate the network connectivity solely from $\mathbf{V}$ and $\mathbf{I}$, Thm. 1(B) implies it is always possible to estimate effective impedances *between active nodes*. That is, the network's input-output behavior is fully contained in active node measurements $\mathbf{V}_{\mathcal{A}}$ and $\mathbf{I}_{\mathcal{A}}$. Indeed, the matrix relating $\mathbf{V}_{\mathcal{A}}$ and $\mathbf{I}_{\mathcal{A}}$ *is* the Kron reduced admittance $\mathbf{Y}_{\mathsf{K}}$, and it can be *uniquely* recovered from $\mathbf{V}_{\mathcal{A}}$ and $\mathbf{I}_{\mathcal{A}}$ alone. This is established (as a particular case of a broader result), by Thm. 2:

Theorem 2. *Given measurements $\mathbf{V} \in \mathbb{C}^{m \times T}$ and $\mathbf{I} \in \mathbb{C}^{m \times T}$, at node set $\mathcal{M}$ such that $\mathcal{A} \subseteq \mathcal{M} \subseteq \mathcal{N}$, there is a unique matrix $\mathbf{Y}$ relating $\mathbf{I}$ to $\mathbf{V}$ according to $\mathbf{I} = \mathbf{YV}$ iff $\mathbf{V}$ has full row-rank, and a unique matrix $\mathbf{Z}$ relating $\mathbf{V}$ to $\mathbf{I}$ according to $\mathbf{V} = \mathbf{ZI}$ iff $\mathbf{I}$ has full row-rank. When unique, $\mathbf{Y}$ and $\mathbf{Z}$ are respectively the network impedance and admittance matrices or an appropriate reduction.*

Proof. Suppose $\ker(\mathbf{V}^{\mathsf{T}}) \neq \emptyset$ (ie $\mathbf{V}$ is not full row rank). Then, $\exists y \in \mathbb{C}^{1 \times m}$ such that $y\mathbf{V} = 0$. Let $\hat{\mathbf{Y}}$ be a solution to $\mathbf{I} = \mathbf{YV}$. Then, $\mathbf{I} = \hat{\mathbf{Y}}\mathbf{V} \implies \forall \alpha \in \mathbb{C} : \mathbf{I} = (\hat{\mathbf{Y}} + \alpha y)\mathbf{V}$. Therefore, there are an infinite number of possible solutions. Now suppose $\ker(\mathbf{V}^{\mathsf{T}}) = \emptyset$ ($\mathbf{V}$ does have full row rank). Let $\hat{\mathbf{Y}}_1$ and $\hat{\mathbf{Y}}_2$ be two possible solutions to $\mathbf{I} = \mathbf{YV}$. Then, $\mathbf{I} = \hat{\mathbf{Y}}_1\mathbf{V} = \hat{\mathbf{Y}}_2\mathbf{V} \implies (\hat{\mathbf{Y}}_1 - \hat{\mathbf{Y}}_2)\mathbf{V} = 0$. This contradicts $\ker(\mathbf{V}^{\mathsf{T}}) = \emptyset$, so there can not be two solutions. A symmetric proof applies for the uniqueness of $\mathbf{Z}$ when $\mathbf{I}$ has full row rank. □

While Thm. 1 indicates it isn't generally possible to recover $\mathbf{Y}$ from $\mathbf{Y}_{\mathsf{K}}$, in the special case of radial networks it *is* possible, as discussed next.

Radial Networks

It is a property of the Schur complement that $\mathbf{Y}$ cannot always be recovered from $\mathbf{Y}_{\mathsf{K}}$. The mapping from a matrix A to its Schur complement, $A_{\mathcal{U}}$, with respect to an arbitrary row and column subset $\mathcal{U}$ isn't generally injective, so recovery of A from $A_{\mathcal{U}}$ isn't guaranteed, even when A is restricted to the set of weighted Laplacians of fully-connected graphs. However, when A is further restricted to the set of Laplacians of tree graphs, denoted $\mathcal{T}$, with only passive nodes eliminated—$\mathcal{U} \subseteq \mathcal{P}$—the following holds:

Theorem 3. *The Schur complement mapping from* $A \in \mathcal{T}$ *to* $A_{\mathcal{P}}$ *is injective. That is, no Schur complement of* $A' \in \mathcal{T} \backslash A$ *with respect to any set of passive nodes can produce* $A_{\mathcal{P}}$*.*

Thm. 3 holds for trees with real or complex edge weights, and is stated here without proof. The proof is based on the crg, which demonstrates that a radial tree is exactly recoverable from exact distances. For radial electrical networks, effective impedances are a valid distance metric. Together, Thm. 1-3 imply Corollary 1:

Corollary 1. *For any radial electric network,* $\mathbf{Y}$ *is fully recoverable from perfect voltage and current PMU measurements at every active node.*

Estimating $\boldsymbol{\zeta}_{\mathcal{A}}$ is a necessary intermediary step in recovering the full network admittance model from active node measurements. In the next section we present a novel network representation which improves effective impedance estimation accuracy from noisy active node PMU measurements.

4.3.3 subKron Reduction

We introduce the *subKron reduction*, a further reduction of electric networks that builds on the Kron reduction. Recall, the Kron voltage-current relationships: $\mathbf{I}_{\mathcal{A}} = \mathbf{Y}_{K}\mathbf{V}_{\mathcal{A}} \leftrightarrow \mathbf{V}_{\mathcal{A}} = \mathbf{Z}_{K}\mathbf{I}_{\mathcal{A}}$. Since relative, not absolute voltages, drive power flows, we can subtract the substation voltage—assumed to be the first—from all other voltages, without modifying the left hand side (Lemma 1).

$$\begin{bmatrix} - & \mathbf{I}_{\mathcal{A}1}^{T} & - \\ & \vdots & \\ - & \mathbf{I}_{\mathcal{A}a}^{T} & - \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{1K}^{T} \\ \mathbf{Y}_{K1} & \mathbf{Y}_{sK} \end{bmatrix} \begin{bmatrix} - & 0 & - \\ & \vdots & \\ - & (\mathbf{V}_{\mathcal{A}a}^{T} - \mathbf{V}_{\mathcal{A}1}^{T}) & - \end{bmatrix} \tag{15}$$

We define $\mathbf{V}_{s\mathcal{A}} \in \mathbb{C}^{(a-1)\times t}$ and $\mathbf{I}_{s\mathcal{A}} \in \mathbb{C}^{(a-1)\times t}$ as the data matrices in (15) with the first row in each removed. The **subKron admittance matrix,** $\mathbf{Y}_{sK} \in$

$\mathbb{C}^{(a-1)\times(a-1)}$, relates $\mathbf{V}_{s\mathcal{A}}$ to $\mathbf{I}_{s\mathcal{A}}$. $\mathbf{Y}_{sK}$ is equivalent to $\mathbf{Y}_{K}$ with the first row and column removed. This condensation turns (15) to (16) below:

$$\mathbf{I}_{s\mathcal{A}} = \mathbf{Y}_{sK}\mathbf{V}_{s\mathcal{A}} \tag{16}$$

We can equivalently derive an expression for the **subKron impedance matrix** which relates $\mathbf{I}_{s\mathcal{A}}$ to $\mathbf{V}_{s\mathcal{A}}$:

$$\mathbf{V}_{s\mathcal{A}} = \mathbf{Z}_{sK}\mathbf{I}_{s\mathcal{A}} \tag{17}$$

The elements of $\mathbf{Z}_{sK}$ are:

$$\mathbf{Z}_{sK}(i,j) = \mathbf{Z}_{K}(i+1,j+1) - \mathbf{Z}_{K}(1,j+1) - \mathbf{Z}_{K}(1,i+1) + \mathbf{Z}_{K}(1,1) \tag{18}$$

Proof. We can derive the subKron impedance matrix as follows: $\mathbf{V}_{\mathcal{A}} - \mathbf{1}\mathbf{V}_{\mathcal{A}1}^{T} = \mathbf{Z}_{K}\mathbf{I}_{\mathcal{A}} - \mathbf{1}\mathbf{Z}_{K1}^{T}\mathbf{I}_{\mathcal{A}}$. $\mathbf{Z}_{K1}^{T}$ is the first row of $\mathbf{Z}_{K}$ and, by symmetry, also the first column. Expanding the above expression gives:

$$\begin{bmatrix} - & 0 & - \\ & \vdots & \\ - & (\mathbf{V}_{\mathcal{A}a} - \mathbf{V}_{\mathcal{A}1})^{T} & - \end{bmatrix} = \begin{bmatrix} - & (\mathbf{Z}_{K1} - \mathbf{Z}_{K1})^{T} & - \\ & \vdots & \\ - & (\mathbf{Z}_{Ka} - \mathbf{Z}_{K1})^{T} & - \end{bmatrix} \mathbf{I}_{\mathcal{A}}$$

Where $\mathbf{Z}_{K1}, ..., \mathbf{Z}_{Ka}$ are the rows (or equivalently the transposed columns) of $\mathbf{Z}_{K}$. By conservation of current, $I_{\mathcal{A}1} = -\sum_{i=2}^{a} I_{\mathcal{A}i}$, allowing us to rewrite the above as follows:

$$\mathbf{I}_{\mathcal{A}} = \begin{bmatrix} - & \mathbf{I}_{\mathcal{A}1} & - \\ & \vdots & \\ - & \mathbf{I}_{\mathcal{A}a} & - \end{bmatrix} = \begin{bmatrix} -\mathbf{1}^{T} \\ \mathbb{I}_{(a-1)} \end{bmatrix} \begin{bmatrix} - & \mathbf{I}_{\mathcal{A}2} & - \\ & \vdots & \\ - & \mathbf{I}_{\mathcal{A}a} & - \end{bmatrix}$$

$$\mathbf{V}_{\mathcal{A}} - \mathbf{1}\mathbf{V}_{\mathcal{A}1}^{T} = (\mathbf{Z}_{K} - \mathbf{1}\mathbf{Z}_{K1}^{T}) \begin{bmatrix} -\mathbf{1}^{T} \\ \mathbb{I}_{(a-1)} \end{bmatrix} \begin{bmatrix} - & \mathbf{I}_{\mathcal{A}2} & - \\ & \vdots & \\ - & \mathbf{I}_{\mathcal{A}a} & - \end{bmatrix}$$

where $\mathbb{I}_{(a-1)}$ is the $(a-1) \times (a-1)$ identity matrix. Define $\bar{\mathbf{Z}} = (\mathbf{Z}_{K} - \mathbf{1}\mathbf{Z}_{K1}^{T}) \begin{bmatrix} -\mathbf{1}^{T} \\ \mathbb{I}_{(a-1)} \end{bmatrix}$. Then the elements of $\bar{\mathbf{Z}} \in \mathbb{C}^{a\times(a-1)}$ are related to the values in $\mathbf{Z}_{K}$ by:

$$\bar{\mathbf{Z}}(i,j) = \mathbf{Z}_{K}(i,j+1) - \mathbf{Z}_{K}(1,j+1) - \mathbf{Z}_{K}(1,i) + \mathbf{Z}_{K}(1,1)$$

Using the symmetry of $\mathbf{Z}_K$, we can see that $\forall j : \ \bar{\mathbf{Z}}(1,j) = 0$: $\bar{\mathbf{Z}}(1,j) = \mathbf{Z}_K(1,j+1) - \mathbf{Z}_K(1,j+1) - \mathbf{Z}_K(1,1) + \mathbf{Z}_K(1,1) = 0$ Therefore, the first row of $\bar{\mathbf{Z}}$ is 0, corresponding to the row of 0s in $\mathbf{V}_{\mathcal{A}} - 1\mathbf{V}_{\mathcal{A}1}^T$. Discarding both rows of zeros reduces the equation dimensionality and leaves us with a square matrix $Z_{sK} \in \mathbb{C}^{(a-1)\times(a-1)}$.

$$\begin{bmatrix} - & (\mathbf{V}_{\mathcal{A}2} - \mathbf{V}_{\mathcal{A}1})^T & - \\ & \vdots & \\ - & (\mathbf{V}_{\mathcal{A}A} - \mathbf{V}_{\mathcal{A}1})^T & - \end{bmatrix} = \mathbf{Z}_{sK} \begin{bmatrix} - & \mathbf{I}_{\mathcal{A}2} & - \\ & \vdots & \\ - & \mathbf{I}_{\mathcal{A}A} & - \end{bmatrix}$$

$\mathbf{Z}_{sK}$ is the **subKron impedance matrix** with elements defined by Eq. (18), which comes directly from the values in $\bar{Z}$. The two subKron forms of Ohm's Law are then:

$$[\text{Admittance}]\ \mathbf{I}_{s\mathcal{A}} = \mathbf{Y}_{sK}\mathbf{V}_{s\mathcal{A}} \leftrightarrow \mathbf{V}_{s\mathcal{A}} = \mathbf{Z}_{sK}\mathbf{I}_{s\mathcal{A}}\ [\text{Impedance}]$$

□

The subKron representation has lower dimensionality—by one—than the Kron representation, but is informationally equivalent:

Lemma 3. $\mathbf{Y}_K$ *is always recoverable from* $\mathbf{Y}_{sK}$ *for a network without shunt admittances.*

Proof. $\mathbf{Y}_K$ is a loopless Laplacian. $\mathbf{Y}_K\mathbf{1} = 0$ and $\mathbf{Y}_K^T = \mathbf{Y}_K$. $\mathbf{Y}_{sK}$ corresponds to $\mathbf{Y}_K$ with the first row and column removed. We can recover the first row and column of $\mathbf{Y}_K$ from the elements of $\mathbf{Y}_{sK}$ as follows: $i \in \{2, ..., a\}$: $\mathbf{Y}_K(i,1) = \mathbf{Y}_K(1,i) = -\sum_{j=1}^{a-1} \mathbf{Y}_{sK}(i,j)$ and $\mathbf{Y}_K(1,1) = -\sum_{i=2}^{a} \mathbf{Y}_K(i,1)$. □

An important feature of the subKron representation is its rank:

Lemma 4. *Unlike* $\mathbf{Y}$ *or* $\mathbf{Y}_K$, $\mathbf{Y}_{sK}$ *is always full rank.*

Proof. This follows from Kirchhoff's matrix tree theorem, which states that the total weight of trees in a graph is equal to *any* cofactor of the graph's weighted Laplacian.

$$T(\mathbf{Y}_K) \triangleq \sum_{T \in \mathcal{T}(\mathbf{Y}_K)} w(T)$$

$\mathsf{T}(\mathbf{Y}_K)$ denotes the total weight of spanning trees in the graph described by Kron impedance matrix $\mathbf{Y}_K$, T is one particular spanning tree of $\mathbf{Y}_K$ in the set of all spanning trees $\mathcal{T}(\mathbf{Y}_K)$, and $w(\mathsf{T})$ is the weight of tree T (the sum of the weights of the edges in T). One cofactor of $\mathbf{Y}_K$ is $\mathbf{Y}_K(1,1)\det(\mathbf{Y}_{sK})$. Therefore: $\mathbf{Y}_K(1,1)\det(\mathbf{Y}_{sK}) = \mathsf{T}(\mathbf{Y}_K)$. Since $|\mathcal{T}(\mathbf{Y}_K)| \geqslant 1$, we have $\mathsf{T}(\mathbf{Y}_K) > 0$. Therefore, $\det(\mathbf{Y}_{sK}) > 0$, proving $\mathbf{Y}_{sK}$ is full rank. □

Consequently, we can convert between the subKron impedance and admittance representations by taking the true matrix inverse: $\mathbf{Z}_{sK} = \mathbf{Y}_{sK}^{-1}$. In graph theory terms, the subKron matrix $\mathbf{Y}_{sK}$ is the reduced Laplacian of $\mathbf{Y}_K$. In the following section, we delve deeper into the physical meaning of the Kron and subKron network matrices, by returning to effective impedances.

4.3.4 Effective Impedances Between Active Nodes

Effective impedances are useful to concretely establish what is lost and preserved in the Kron and subKron reductions. An electrical network admittance matrix contains two fundamental pieces of information: network connectivity and effective impedances between network node pairs. Kron reduction discards the connectivity information, but preserves the effective impedances between the active nodes, which characterize the network's input-output properties. By Lemma 3, subKron reduction also preserves inter-active node effective impedances. Therefore, though different in dimensionality and definition, all the network representations introduced in Sections 4.3.1-4.3.3 may be used to calculate effective impedances between active nodes.

Lemma 5 below establishes useful relationships between the elements of subKron matrix $\mathbf{Z}_{sK}$ and the effective impedances of the original network, ζ.

Lemma 5. *The* $(i,j)^{th}$ *element of* $\mathbf{Z}_{sK}$ *is the effective impedance of the* ***shared path*** *between nodes* $i, j \in \mathcal{A}$ *and the substation.*

Proof. Combining equations (14) and (18):

$$\mathbf{Z}_{sK}(i,j) = \frac{1}{2}\left[\zeta_{1(j+1)} + \zeta_{1(i+1)} - \zeta_{(i+1)(j+1)} - \zeta_{11}\right]$$

□

$$\begin{tikzcd}
\mathbf{Y} \arrow[rr, "\text{Kron Reduction}"] \arrow[d, leftrightarrow, "(\circ)^{\dagger}"'] & & \mathbf{Y}_K \arrow[rr, leftrightarrow, "\text{subKron Reduction}"] \arrow[d, leftrightarrow, "(\circ)^{\dagger}"'] & & \mathbf{Y}_{sK} \arrow[d, leftrightarrow, "(\circ)^{-1}"'] \\
\mathbf{Z} \arrow[rr, "\text{Kron Reduction}"] \arrow[dr, leftrightarrow] & & \mathbf{Z}_K \arrow[rr, leftrightarrow, "\text{subKron Reduction}"] \arrow[dr, leftrightarrow] & & \mathbf{Z}_{sK} \arrow[dl, leftrightarrow] \\
& \zeta \arrow[rr] & & \zeta_{\mathcal{A}} &
\end{tikzcd}$$

Figure 29: Relationships between the different network models. Bidirectional arrows indicate informational equivalence of representations.

Corollary 2. *The diagonal elements of* $\mathbf{Z}_{sK}$ *are the effective impedances to the substation.*

Proof. By Eq. (18), the diagonal elements of $\mathbf{Z}_{sK}$ are related to the elements of $\mathbf{Z}_K$:

$$Z_{sK}(i,i) = Z_K(i+1,i+1) + Z_K(1,1) - 2Z_K(1,i+1) = \zeta_{1(i+1)}$$

$\square$

The relationships between the different network models and the effective impedances are summarized in Fig. 29. Note that $\boldsymbol{\zeta}_{\mathcal{A}} \subseteq \boldsymbol{\zeta}$.

4.3.5 Impedance Estimation

The Kron and subKron representations are informationally equivalent, but estimating them from real-world measurements are not identical challenges. Noise in real world measurements makes network impedance estimation—i.e. "inverting" Ohm's Law to infer $\mathbf{Y}$ or $\mathbf{Z}$ from $\mathbf{V}$ and $\mathbf{I}$—nontrivial. Here, impedance estimation refers to estimation of either the complete ($\mathbf{Y}$ or $\mathbf{Z}$), Kron ($\mathbf{Y}_K$ or $\mathbf{Z}_K$) or subKron ($\mathbf{Y}_{sK}$ or $\mathbf{Z}_{sK}$) network representations. As we establish in the following, the major advantage of the subKron reduction is that it can be estimated with better accuracy than the Kron reduction in the presence of noise.

Ordinary Least Squares

A simple regression method is *ordinary least squares* (ols). Given data matrices $\mathbf{R} \in \mathbb{C}^{n \times t}$, $\mathbf{P} \in \mathbb{C}^{n \times t}$ which are known to obey an underlying linear relationship parametrized by matrix $\mathbf{M} \in \mathbb{C}^{n \times n}$, we can write: $R \approx MP$ where we use $\approx$ rather than $=$ to reflect that $\mathbf{R}$ and $\mathbf{P}$ contain noise. Then, ols estimates $\mathbf{M}$ from $\mathbf{R}$ and $\mathbf{P}$ as

$$\hat{\mathbf{M}} = \arg\min_{\mathbf{M}} ||\mathbf{R} - \mathbf{M}\mathbf{P}||_F$$

The solution to this objective, which is the best linear unbiased estimate of $\mathbf{M}$, has a closed form:

$$\hat{\mathbf{M}} = (\mathbf{P}^H\mathbf{P})^{-1}\mathbf{P}^H\mathbf{R}$$

There are no constraints on the structure of $\hat{\mathbf{M}}$ in the above formulation.

Many alternatives to ols exist including those which regularize the objective, explicitly account for noise in both $\mathbf{P}$ and $\mathbf{R}$ or enforce special structure on $\mathbf{M}$. We focus on ols estimation because of its intuitive objective, lack of tuning parameters, and computational efficiency, which makes it suited to online applications of impedance estimation such as fault detection. We also find ols performs comparably to alternative techniques in practice. Note that ols can be replaced by any regression technique within our larger proposed method.

For real-world applicability, we must understand the sensitivity of estimate $\hat{\mathbf{M}}$ to noise in the measurements. Depending on the structure of the $\mathbf{P}^H\mathbf{P}$ matrix, $(\mathbf{P}^H\mathbf{P})^{-1}$ can contain very large values that magnify noise in $\mathbf{P}^H\mathbf{R}$ and produce a poor $\hat{\mathbf{M}}$ solution. The *condition number* of $\mathbf{P}^H\mathbf{P}$ is a metric for understanding the noise magnification of $(\mathbf{P}^H\mathbf{P})^{-1}$ and is defined as the ratio of its largest and smallest eigenvalues:

$$\text{cond}(Q) \triangleq \frac{\sigma_{max}(Q)}{\sigma_{min}(Q)}$$

A larger $\text{cond}(\mathbf{P}^H\mathbf{P})$ means $(\mathbf{P}^H\mathbf{P})^{-1}$ will magnify noise, and $\hat{\mathbf{M}}$ will be more noise sensitive. At the extremes, $\text{cond}(Q) = 1$ means Q is very well-conditioned, while $\text{cond}(Q) = \infty$ means it is rank deficient and not invertible. $\text{cond}(\mathbf{P}^H\mathbf{P})$ is related to the correlation between the rows of $\mathbf{P}$. If the rows of $\mathbf{P}$ are highly correlated, $\text{cond}(\mathbf{P}^H\mathbf{P})$ is large, and the estimate is more noise sensitive.

Conditioning of Impedance Models

The ols network impedance estimation from measurement data is formulated as follows:

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} ||\mathbf{I} - \mathbf{Y}\mathbf{V}||_F = (\mathbf{V}^H\mathbf{V})^{-1}\mathbf{V}^H\mathbf{I} \tag{19}$$

$|| \circ ||_F$ indicates the Frobenius norm. Note the inevitable inversion of a data matrix product: $\mathbf{V}^H\mathbf{V}$ in (19). Expressions for $\hat{\mathbf{Y}}_K$, $\hat{\mathbf{Y}}_{sK}$, $\hat{\mathbf{Z}}_K$ and $\hat{\mathbf{Z}}_{sK}$ are determined by active node measurements alone (Thm. 2). The full set of ols model estimate equations are given below:

Complete:

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} ||\mathbf{I} - \mathbf{Y}\mathbf{V}||_2 = (\mathbf{V}^H\mathbf{V})^{-1}\mathbf{V}^H\mathbf{I}$$

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{Z}} ||\mathbf{V} - \mathbf{Z}\mathbf{I}||_2 = (\mathbf{I}^H\mathbf{I})^{-1}\mathbf{I}^H\mathbf{V}$$

Kron reduced:

$$\hat{\mathbf{Y}}_K = \arg\min_{\mathbf{Y}} ||\mathbf{I}_{\mathcal{A}} - \mathbf{Y}\mathbf{V}_{\mathcal{A}}||_2 = (\mathbf{V}_{\mathcal{A}}^H\mathbf{V}_{\mathcal{A}})^{-1}\mathbf{V}_{\mathcal{A}}^H\mathbf{I}_{\mathcal{A}}$$

$$\hat{\mathbf{Z}}_K = \arg\min_{\mathbf{Z}} ||\mathbf{V}_{\mathcal{A}} - \mathbf{Z}\mathbf{I}_{\mathcal{A}}||_2 = (\mathbf{I}_{\mathcal{A}}^H\mathbf{I}_{\mathcal{A}})^{-1}\mathbf{I}_{\mathcal{A}}^H\mathbf{V}_{\mathcal{A}}$$

subKron reduced:

$$\hat{\mathbf{Y}}_{sK} = \arg\min_{\mathbf{Y}} ||\mathbf{I}_{s\mathcal{A}} - \mathbf{Y}\mathbf{V}_{s\mathcal{A}}||_2 = (\mathbf{V}_{s\mathcal{A}}^H\mathbf{V}_{s\mathcal{A}})^{-1}\mathbf{V}_{s\mathcal{A}}^H\mathbf{I}_{s\mathcal{A}}$$

$$\hat{\mathbf{Z}}_{sK} = \arg\min_{\mathbf{Z}} ||\mathbf{V}_{s\mathcal{A}} - \mathbf{Z}\mathbf{I}_{s\mathcal{A}}||_2 = (\mathbf{I}_{s\mathcal{A}}^H\mathbf{I}_{s\mathcal{A}})^{-1}\mathbf{I}_{s\mathcal{A}}^H\mathbf{V}_{s\mathcal{A}}$$

Each formulation will have different noise sensitivity since each inverts a different data matrix. The rank constraints of Thm. 1 immediately indicate an issue with estimating **Y** or **Z** from measurements at *every* network node: $\mathbf{V}, \mathbf{I} \in \mathbb{C}^{n \times t}$. Thm. 1 establishes **I** is rank deficient, making computation of $(\mathbf{I}^H\mathbf{I})^{-1}$ impossible. When $n \geqslant p + 1$ (the network contains more than one passive node), **V** will also be rank deficient, making computation of $(\mathbf{V}^H\mathbf{V})^{-1}$ similarly impossible. Note Thm. 1 is for *noiseless* measurement matrices. With noisy **V** and **I**, the rank constraints won't hold exactly, but will nonetheless manifest in the matrix conditioning. Therefore, though it may be possible to compute $(\mathbf{I}^{H}\mathbf{I})^{-1}$ and $(\mathbf{V}^{H}\mathbf{V})^{-1}$ for real measurements, these matrices will be highly ill-conditioned, and the resulting impedance estimates will be noise sensitive to the point of impracticality. Similarly, $\mathbf{Z}_K$ cannot be estimated directly as $\mathbf{I}_{\mathcal{A}}$ is rank deficient.

While $\mathbf{Y}$, $\mathbf{Z}$, and $\mathbf{Z}_K$ cannot be estimated uniquely, Thm. 2 establishes that $\mathbf{Y}_K$, $\mathbf{Y}_{sK}$ and $\mathbf{Z}_{sK}$ can, since $\mathbf{V}_{\mathcal{A}}, \mathbf{V}_{s\mathcal{A}}$ and $\mathbf{I}_{s\mathcal{A}}$ are full rank. This is another perspective from which to see the Kron reduction as the best *possible* estimate of the network impedance given no prior information. While $\mathbf{Y}_K$, $\mathbf{Z}_{sK}$ and $\mathbf{Y}_{sK}$ are informationally equivalent, estimating them from real data will have differing accuracies as each entails the inversion of a different data matrix with its own condition number. The following conditioning trend holds for the measurement data:

$$\text{cond}(\mathbf{I}_{s\mathcal{A}}^{H}\mathbf{I}_{s\mathcal{A}}) < \text{cond}(\mathbf{V}_{s\mathcal{A}}^{H}\mathbf{V}_{s\mathcal{A}}) < \text{cond}(\mathbf{V}_{\mathcal{A}}^{H}\mathbf{V}_{\mathcal{A}}) \tag{20}$$

The conditioning improvement from $\mathbf{V}_{\mathcal{A}}$ to $\mathbf{V}_{s\mathcal{A}}$ is intuitive, as taking voltage differences is akin to common mode removal [115], reducing data correlation. While nodal currents are predominantly driven by individual loads which often behave independently, nodal voltages are correlated by the network structure itself. Therefore, voltage measurements tend to be more correlated than currents, causing $\mathbf{V}_{s\mathcal{A}}$ to be more poorly conditioned than $\mathbf{I}_{s\mathcal{A}}$. Thus, $\hat{\mathbf{Y}}_{sK}$ is more noise robust than $\hat{\mathbf{Y}}_K$, and $\hat{\mathbf{Z}}_{sK}$ estimation is the most noise robust. Ultimately, the improved data conditioning provides better $\boldsymbol{\zeta}_{\mathcal{A}}$ estimates and, in turn, more accurate network reconstruction via the Recursive Grouping algorithm, described next.

4.3.6 Recursive Grouping Algorithm

To reiterate, the Kron reduction preserves the effective impedances between active nodes. Thm. 3 tells us that in a radial network, effective impedances between every pair of active nodes is sufficient to recover the full network structure. In other words, with one piece of apriori information about the network—specifically that it is radial—we are guaranteed recovery of the network topology and the impedances between *all* nodes just from $\boldsymbol{\zeta}_{\mathcal{A}}$.

The recursive grouping (rg) algorithm, presented and described in [144], reconstructs a radial network from real-valued information distances d between the "observed" nodes $\mathcal{O}$, defined here as the set of nodes for which the information distances are known but the parent node is unknown. By comparing the pairwise quantities d_{ij} to the triplet quantities

$\Phi_{ijk} \triangleq d_{ik} - d_{jk}, \forall(i,j,k) \in \mathcal{O}$, $\mathcal{O}$ is recursively shrunk until the tree is reconstructed, as demonstrated on the toy network in Fig. 30. Using resistance and reactance as the information distances in two separate, real-valued reconstructions, [145] applied rg to radial distribution networks. With noisy data, this separation into real-valued reconstructions is a source of unnecessary reconstruction error.

We present *complex recursive grouping* (crg) is a modification of relaxed rg that uses complex-valued effective impedances directly in the reconstruction. That is, $\mathbf{d}_{ij} = \boldsymbol{\zeta}_{ij}$. To understand crg, it is important to have a rudimentary understanding of its algorithmic progenitors, described in the following interlude.

Interlude: Recursive Grouping & Relaxed Recursive Grouping

Recursive grouping recursively reconstructs the radial network, progressively shrinking $\mathcal{O}$ as parent nodes are identified and introduced according to the distance comparisons of Lemma 6. For distribution networks, the *initial* $\mathcal{O}$ is the set $\mathcal{A}$, which includes at least all of the leaf nodes. Once a parent is identified or introduced, its children nodes are removed from $\mathcal{O}$.

To determine connectivity relationships between observed nodes, rg compares pairwise distance d_{ij} with the distance difference $\Phi_{ijk} \triangleq d_{ik} - d_{jk}$, for all possible combinations $i, j, k \in \mathcal{O}$. The following Lemma 6 [144] describes the relationship between d_{ij} and Φ_{ijk} that hold for nodes which are parent-child pairs or siblings.

Lemma 6. ***(i)*** $d_{ij} = \Phi_{ijk}, \forall k \in \mathcal{O}\backslash(i,j)$ *iff* i *is a leaf node in* $\mathcal{O}$*, and* j *is the parent of* i*.*
(ii) $-d_{ij} < \Phi_{ijk} = \Phi_{ijk'} < d_{ij}, \forall k, k' \in \mathcal{O}\backslash(i,j)$ *iff* i *and* j *are leaf nodes in* $\mathcal{O}$*, and siblings.*

If two or more sibling nodes are discovered but their parent node is absent in $\mathcal{O}$, a new parent node is added to the tree. As new nodes are added, the effective impedances between them and the rest of the network are calculated using linear operations (eqns. $(13) - (14)$ in [144]). Thus, overall, rg is a linear operator.

[144] also introduces "relaxed recursive grouping" for application in scenarios where the inter-node distance estimates ($\hat{d}$) are contaminated by noise and errors because of which the relationships of Lemma 6 will no

longer hold exactly. For this scenario, the parent-child criteria of Lemma 6 is replaced with:

$$|\hat{d}_{ij} - \hat{\Phi}_{ijk}| \leqslant \epsilon, \forall k \in \mathcal{K}_{ij}, \tag{21}$$

and the sibling group criteria of Lemma 6 is replaced with:

$$\hat{\Lambda}_{ij} \triangleq \max_{k \in \kappa_{ij}}(|\hat{\Phi}_{ijk}|) - \min_{k \in \kappa_{ij}}(|\hat{\Phi}_{ijk}|) \leqslant \epsilon. \tag{22}$$

Here, as per [144], $\mathcal{K}_{ij}$ is a subset of nodes $\mathcal{O}$ that are within some proximity of both i and j according to our distance metric. Eqs. (21) and (22) introduce a bias, because it is more likely that node pair (i, j) will be recognized as a parent-child or sibling pair if set $\mathcal{K}_{ij}$ contains fewer nodes. To avoid this, we modify the relaxed rg tests to consider the average of $\hat{\Phi}_{ijk}$ value over $k \in \mathcal{K}_{ij}$ rather than the max. The proposed relaxed rg parent-child and sibling tests are precisely the crg tests described next.

There is one final caveat. In order for relaxed RG to converge, it must select at least two nodes in $\mathcal{O}$ to be a parent/children pair or siblings at each iteration. But with imperfect $\hat{d}$ estimates, it is possible that no nodes in $\mathcal{O}$ satisfy the parent-child or sibling tests for a given ϵ. This can be addressed using the expanding ϵ technique presented in [145] in which ϵ is iteratively increased until some nodes satisfy the parent-child or sibling tests.

In crg, which uses complex-valued distances, Lemma 6 becomes:

Lemma 7. ***(i)*** *$\boldsymbol{\zeta}_{ij} = \boldsymbol{\Phi}_{ijk}, \forall k \in \mathcal{O}\backslash(i, j)$ iff i is a leaf node in $\mathcal{O}$, and j is the parent of i.*
(ii) *$\boldsymbol{\Phi}_{ijk} = \boldsymbol{\Phi}_{ijk'} \neq \boldsymbol{\zeta}_{ij}, \forall k, k' \in \mathcal{O}\backslash(i, j)$ iff i and j are leaf nodes in $\mathcal{O}$, and siblings.*

While crg uses complex impedances, the inequality tests of crg must be performed on the real number field $\mathbb{R}$ which, unlike $\mathbb{C}$, is ordered. Formulating real-valued inequality tests for complex-valued impedances requires care. A natural suggestion is to use the lexicographic ordering; that is, to compare magnitudes. However, with noise and imperfect $\hat{\boldsymbol{\zeta}}$ estimates, this may lead to reconstruction errors. Instead, we use the ϵ-neighborhood of $\hat{\boldsymbol{\Phi}}_{ijk}$ in the complex plane for some tolerance choice ϵ. A complex number $\mathbf{z}$ is in the ϵ-neighborhood of $\hat{\boldsymbol{\Phi}}_{ijk}$ if $|\mathbf{z} - \hat{\boldsymbol{\Phi}}_{ijk}| \leqslant \epsilon$. As in [144], we define $\mathcal{K}_{ij}$ as the subset of nodes close to both i and j, with closeness quantified by parameter τ:

$$\mathcal{K}_{ij} \triangleq \{k \in \mathcal{O}\backslash(i, j) | \max(\hat{\mathbf{d}}_{ik}, \hat{\mathbf{d}}_{jk}) \leqslant \tau\}$$

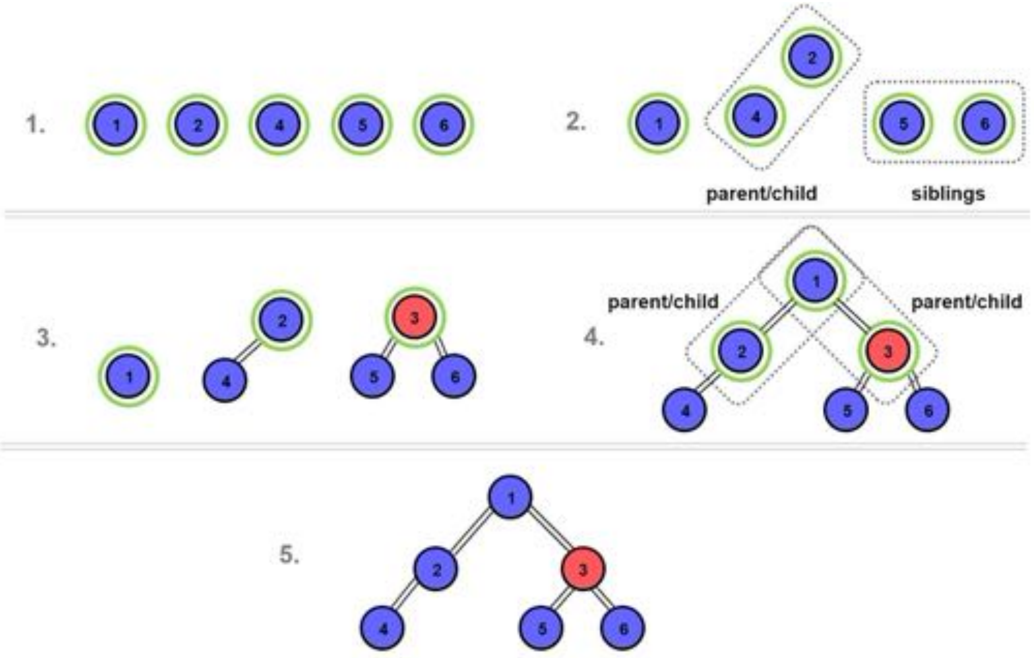


Figure 30: Applying crg to the toy network of Fig. 28. The nodes in $\mathcal{O}$ are highlighted in green in each step.

The parent-child test in crg is:

$$\hat{\beta}_{ij} \triangleq \frac{1}{|\mathcal{K}_{ij}|} \sum_{k \in \mathcal{K}_{ij}} (|\hat{\mathbf{d}}_{ij} - \hat{\boldsymbol{\Phi}}_{ijk}|) \leqslant \epsilon \tag{23}$$

The crg sibling test is:

$$\hat{\Gamma}_{ij} \triangleq \frac{1}{|\mathcal{K}_{ij}|(|\mathcal{K}_{ij}|-1)} \sum_{k \in \mathcal{K}_{ij}} \sum_{k' \in \mathcal{K}_{ij}/k} (|\hat{\boldsymbol{\Phi}}_{ijk} - \hat{\boldsymbol{\Phi}}_{ijk'}|) \leqslant \epsilon, \tag{24}$$

where $\hat{\beta}_{ij}$ is the average distance between the complex quantities $\hat{\mathbf{d}}_{ij}$ and $\hat{\boldsymbol{\Phi}}_{ijk}$ for $k \in \mathcal{K}_{ij}$, and $\hat{\Gamma}_{ij}$ is the average distance between $\hat{\boldsymbol{\Phi}}_{ijk}$ and $\hat{\boldsymbol{\Phi}}_{ijk'}$ for $k, k' \in \mathcal{K}_{ij}$. (24) may incorrectly include parent nodes in sibling sets. This is resolved by removing the parent node from the set of siblings, once the parent is identified by (23). Finally, the equations that determine the effective impedances for the new parent nodes (eqns. $(27)-(28)$ in [144]) are linear, and do not need to be modified for complex quantities.

4.3.7 Simulation and Results

We test our topology recovery methods on synthetic data generated by simulating seven different radial test feeders with the power flow tool MATPOWER [149]. To emulate realistic conditions, we use minute-resolution real power consumption data of nearby homes from the Pecan Street project, thereby preserving load correlations that exist in the real world. These correlations manifest in voltage and current measurements and can affect estimation performance. The seven cases have 4, 9, 12, 13, 13, 18, and 29 load nodes. The two 13 load cases differ in their topologies. All test cases have 3 passive nodes, except the 12 load case which has 4. Five of the seven test cases are visualized in Fig. 31. For each case, we simulate and collect time series phasor measurements at the active (load and substation) nodes *only*. To model real µPMU measurements, we add *complex additive white Gaussian noise* (AWGN) to the voltage and current measurements generated by MATPOWER. This a reasonable noise model choice based on empirical PMU noise studies [45]. The noise level is specified as a signal-to-noise ratio (SNR): the mean squared magnitude of the true measurement time series divided by the noise variance. Note that the noise sensitivity of ols is determined by the condition number of the inverted data matrices. By definition, the condition number reflects estimate sensitivity to the *norm* of the noise, *not* its distribution. Therefore, though we use AWGN in our simulations, the relative performance of the methods will be the same for other noise distributions [150].
Generating voltage measurements with MATPOWER takes on average 0.98 ms per load per time point. That is, simulating a 10 node network over 1000 time points takes 9.8 seconds. The ols network estimation process takes 0.7 s. per node per trial.

Effective Impedance Estimation Results

From the noisy measurements of each test case, we estimate the effective impedances $\boldsymbol{\zeta}_{\mathcal{A}}$ via three different approaches: estimating either $\hat{\mathbf{Y}}_K$, $\hat{\mathbf{Y}}_{sK}$, or $\hat{\mathbf{Z}}_{sK}$ and then computing $\boldsymbol{\zeta}_{\mathcal{A}}$ using the equations of Section 4.3.4. We determine the error between the three different estimates of $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$—from each of $\hat{\mathbf{Y}}_K$, $\hat{\mathbf{Y}}_{sK}$ and $\hat{\mathbf{Z}}_{sK}$—and the true $\boldsymbol{\zeta}_{\mathcal{A}}$. These are all matrix quantities, so the

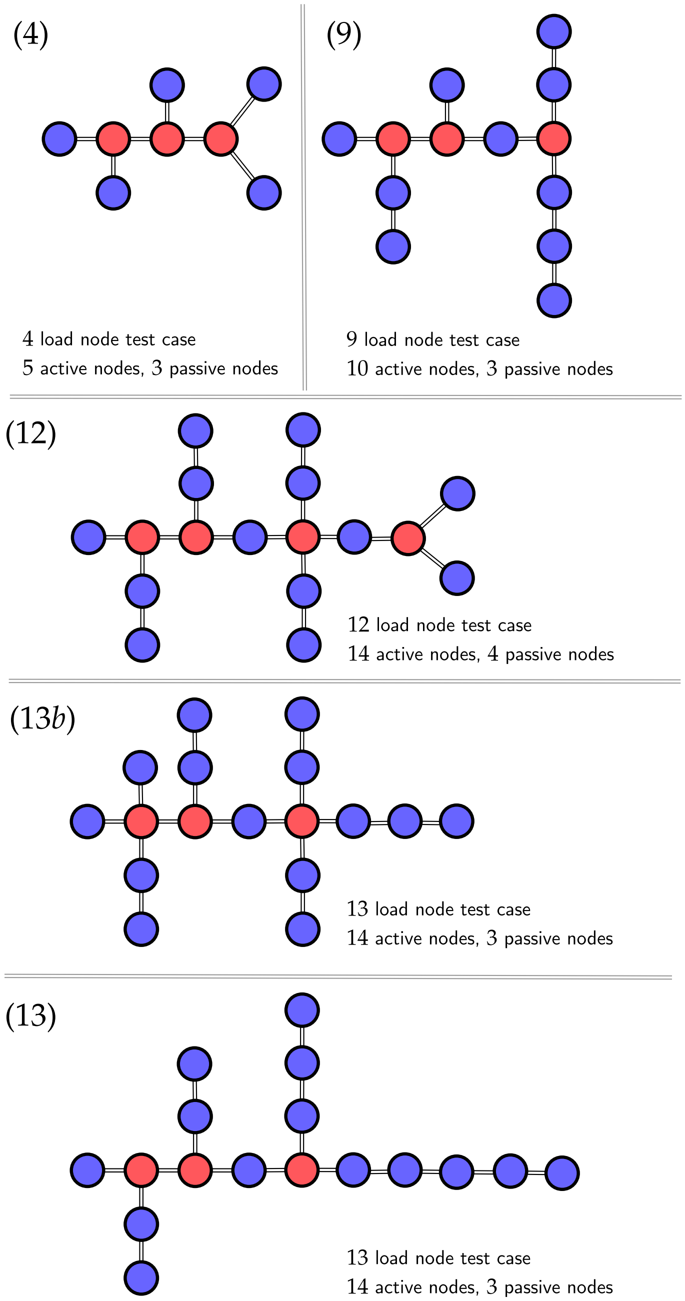


Figure 31: Visualization of five of the total seven test networks. Passive nodes are red, active nodes are blue. The set of active nodes is the set of load nodes plus the substation. The measured nodes are all active nodes.

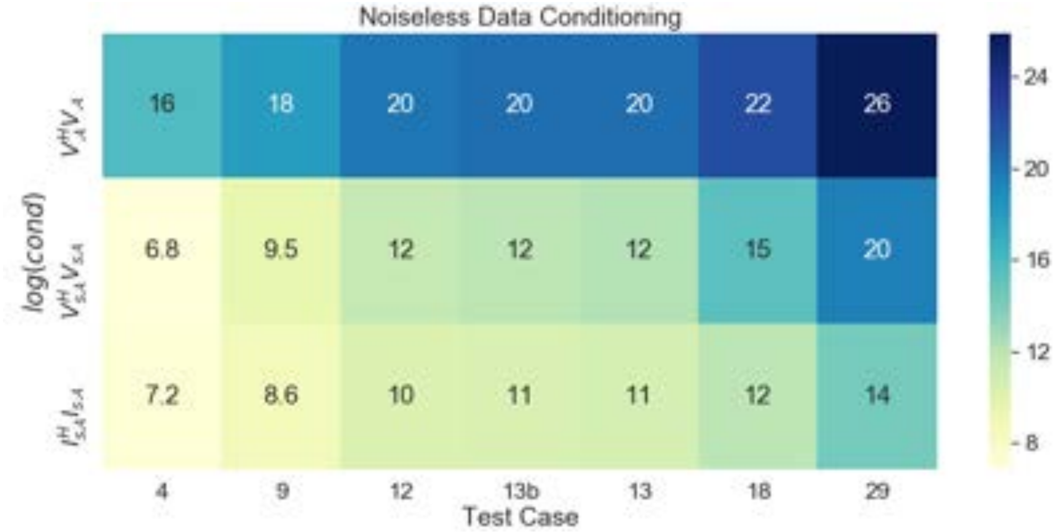
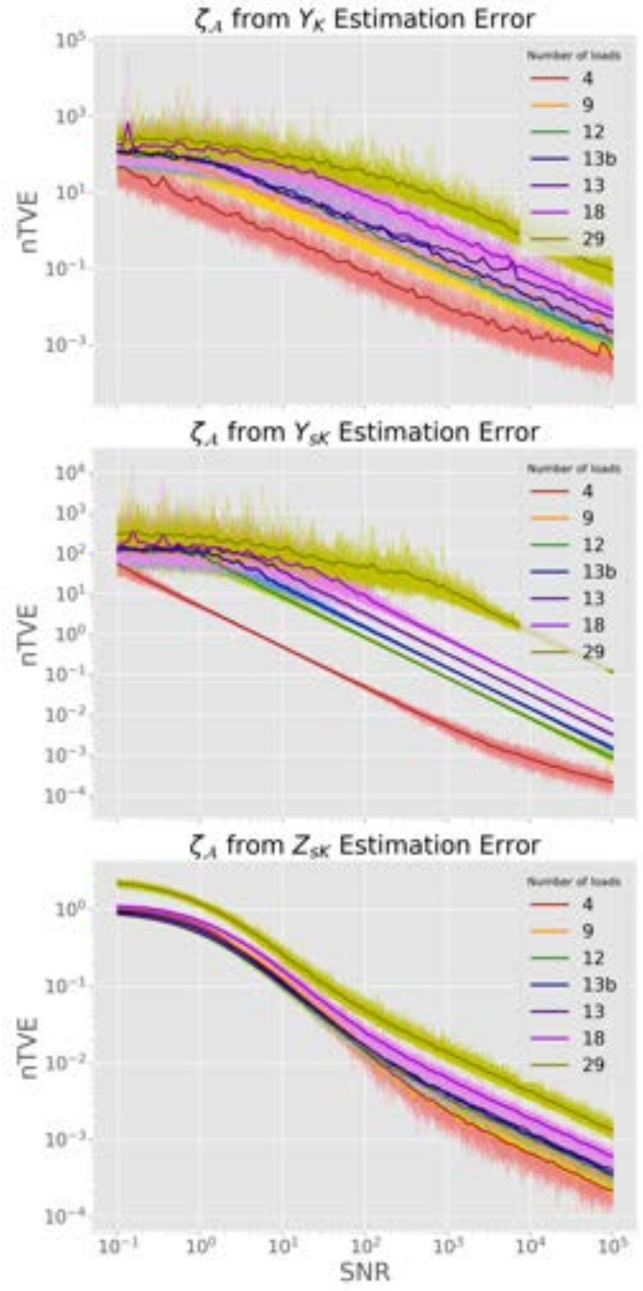


Figure 32: Log of the condition number of noiseless measurements for each test case. Larger values lead to greater noise sensitivity.

error is quantified by and reported as the normalized Total Vector Error (nTVE). The nTVE between complex matrices $\mathbf{M}$ and $\hat{\mathbf{M}}$ is defined as:

$$\text{nTVE}(\mathbf{M}, \hat{\mathbf{M}}) \triangleq \frac{||\mathbf{M} - \hat{\mathbf{M}}||_2}{||\mathbf{M}||_2}$$

$\boldsymbol{\zeta}_{\mathcal{A}}$ estimation accuracy versus SNR is shown in Fig. 33 for each of the three estimation approaches. Across approaches, we see errors are larger and decay more slowly with increasing SNR for test cases with more loads. The corresponding conditioning trend for the test cases is recorded in Fig. 32. As the number of loads (and consequently measurement points) grows, the nodes are more electrically proximate and therefore the voltage measurements more correlated. This leads to poorer conditioning resulting in increased estimation error and noise sensitivity. This is a compelling and perhaps counterintuitive result—more PMU measurements do not necessarily improve impedance estimation accuracy if there is noise in the measurements. Fig. 33 also conveys the advantage of the subKron representation over the Kron representation. $\boldsymbol{\zeta}_{\mathcal{A}}$ estimation via $\hat{\mathbf{Y}}_{\mathrm{K}}$, $\hat{\mathbf{Y}}_{\mathrm{sK}}$ and $\hat{\mathbf{Z}}_{\mathrm{sK}}$ is affected by the conditioning of $\mathbf{V}_{\mathcal{A}}$, $\mathbf{V}_{\mathrm{s}\mathcal{A}}$ and $\mathbf{I}_{\mathrm{s}\mathcal{A}}$ respectively. Since $\mathbf{V}_{\mathcal{A}}$ is the worst conditioned, $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ extracted from Kron admittance $\mathbf{Y}_{\mathrm{K}}$ has high error at low SNRs and consistently high estimate variance. This result represents the performance of existing methods assuming they use ols estimation. In

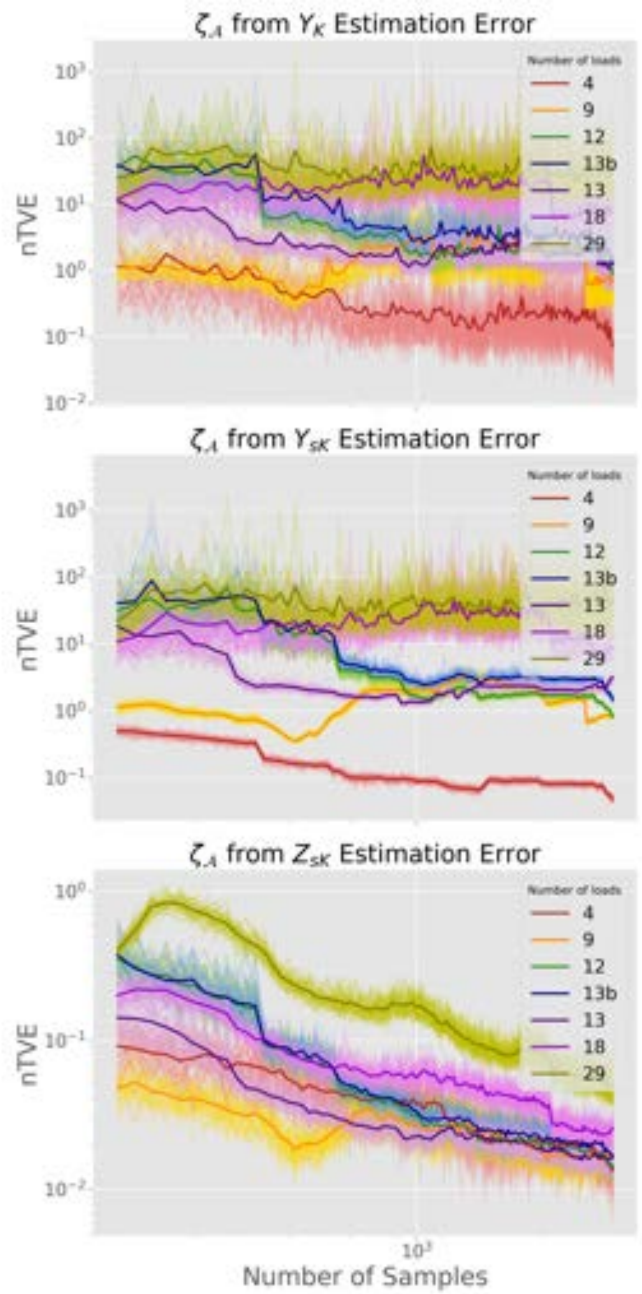


$\zeta_{\mathcal{A}}$ from $Y_{\mathcal{K}}$ Estimation Error

$\zeta_{\mathcal{A}}$ from $Y_{s\mathcal{K}}$ Estimation Error

$\zeta_{\mathcal{A}}$ from $Z_{s\mathcal{K}}$ Estimation Error

Figure 33: Estimation error of $\zeta_{\mathcal{A}}$ via three ols approaches for five test cases with varying noise in $\mathbf{I}$ and $\mathbf{V}$. Light bands indicate error across 100 trials per SNR value with different noise instances. Dark lines show average error.



$\zeta_{\mathcal{A}}$ from Y_K Estimation Error

$\zeta_{\mathcal{A}}$ from Y_{sK} Estimation Error

$\zeta_{\mathcal{A}}$ from Z_{sK} Estimation Error

Figure 34: Estimation error of $\zeta_{\mathcal{A}}$ via three ols approaches for five test cases with varying number of measurement time points t used in estimation. The noise level in $\mathbf{I}$ and $\mathbf{V}$ is constant SNR $= 100$. Light bands indicate error across 100 trials per t choice with different noise instances. Dark lines show average error.

comparison, $\hat{\zeta}_{\mathcal{A}}$ extracted from $\mathbf{Z}_{sK}$ is clearly the most noise-robust with orders of magnitude lower error at low SNRs and dramatically lower estimate variance across the entire SNR test range. Estimation error drops more rapidly with increasing SNR for $\hat{\zeta}_{\mathcal{A}}$ extracted from $\mathbf{Z}_{sK}$ than from $\mathbf{Y}_{K}$. The improvement is especially pronounced for the larger test cases, as in these cases $\mathbf{Y}_{K}$ and $\mathbf{Y}_{sK}$ estimation require inversion of the increasingly correlated and ill-conditioned voltage measurements, while $\mathbf{I}_{sK}$ estimation inverts the mostly uncorrelated current measurements.

While low mean error is obviously critical, low error variance is also important. The high variance of noise sensitive estimation methods has operational consequences. For example, a dramatic change in the network impedance estimate may be due simply to measurement noise, as opposed to real topological changes or faults, leading to frequent false alarms that waste time *and* reduce operator trust in both the sensors and algorithms. Low error variance is an important advantage of the subKron representation over prior work.

Fig. 34 plots $\zeta_{\mathcal{A}}$ estimation error for increasing number of measurement data points t. The SNR is fixed. The $\mathbf{Z}_{sK}$ method provides orders of magnitude better performance at small values of t indicating that it is more suitable than existing methods for real time applications. The $\mathbf{Z}_{sK}$ method also provides better performance at large values of t, and unlike the $\mathbf{Y}_{K}$ method, increasing t results in a significant reduction of $\hat{\mathbf{Z}}_{sK}$ error across all of the test networks.

Tree Reconstruction Results

To evaluate crg reconstruction performance, we compute the Robinson-Foulds metric (RF) used in computational biology on phylogenetic trees [151]. RF quantifies the difference between two trees that share an active node set. It assesses the validity of each reconstructed edge independent of reconstruction performance upstream or downstream of that edge. We introduce the normalized Robinson-Foulds metric (nRF), defined as the RF metric divided by the total number of active nodes: $\mathrm{nRF} = \frac{\mathrm{RF}}{A}$. Normalization roughly enables performance comparison across networks of different sizes. To evaluate estimation performance, we use nRF to compare the topology reconstruction against the true network. Thus, the nRF serves as an error metric with lower scores indicating better recovery.

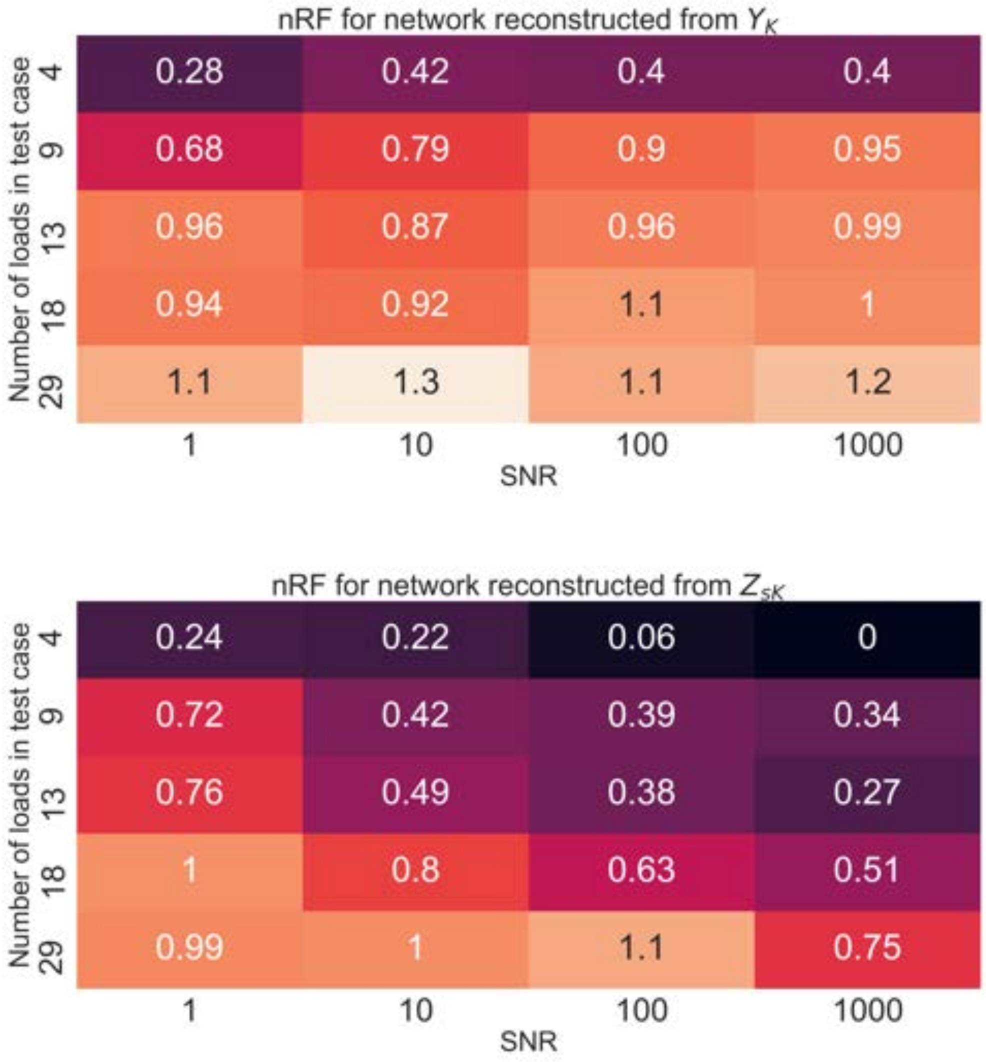
The performance of crg is evaluated on two different effective impedance input data sets. The "baseline" data set consists of effective impedances $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ calculated from the estimate of $\hat{\mathbf{Y}}_{K}$—representing the approach of existing methods that do not consider measurement data conditioning. The second data set consists of $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ calculated from $\hat{\mathbf{Z}}_{sK}$. Fig. 35 shows crg consistently performs better when provided with the more accurate effective impedances calculated from $\hat{\mathbf{Z}}_{sK}$, over those calculated from $\hat{\mathbf{Y}}_{K}$. Performance is significantly better across a range of feeder sizes for SNRs from 10 to 1000.

The execution time of crg scales non-linearly with network size. On $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ from $\hat{\mathbf{Y}}_{K}$, crg runs in .06, .39 and 107 s on the 4, 9, and 29 load networks respectively. On $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ from $\hat{\mathbf{Z}}_{sK}$, crg is faster—.06, .34 and 65 s, respectively. Therefore, crg is faster on more accurate effective impedances because the ϵ tolerance does not need to be increased as often and in each iteration it is more likely that multiple observed nodes are connected.

* * *

This section described tools for extracting visualizable and actionable topology information solely from PMU data with no prior system information except the assumption of radial structure. By starting with fundamental physics, the tools have strong theoretical grounding and well-defined performance guarantees and noise sensitivity. This physics-oriented approach yields both practical tools as well as insightful theoretical results. This section made the following contributions:

- It is generally impossible to uniquely recover the complete network even from comprehensive $\mathbf{V}$ and $\mathbf{I}$ phasor measurements. However, it is always possible to recover $\boldsymbol{\zeta}_{\mathcal{A}}$, pairwise effective impedances between active nodes, from $\mathbf{V}$ and $\mathbf{I}$ measurements at all active nodes. Happily, in radial networks, $\boldsymbol{\zeta}_{\mathcal{A}}$ is sufficient for complete network recovery.
- The subKron representation, though informationally equivalent to the Kron reduced representation, proffers a more noise robust approach for $\boldsymbol{\zeta}_{\mathcal{A}}$ estimation.
- For radial systems, the complex recursive grouping (crg) algorithm recovers the full network from $\boldsymbol{\zeta}_{\mathcal{A}}$ and performs well even when $\hat{\boldsymbol{\zeta}}_{\mathcal{A}}$ is noisy.
- Our simulations demonstrate the efficacy and practical challenges of the subKron and crg methods. While subKron-based $\boldsymbol{\zeta}_{\mathcal{A}}$ estimation



nRF for network reconstructed from Y_K

Number of loads in test case	1	10	100	1000
4	0.28	0.42	0.4	0.4
9	0.68	0.79	0.9	0.95
13	0.96	0.87	0.96	0.99
18	0.94	0.92	1.1	1
29	1.1	1.3	1.1	1.2

nRF for network reconstructed from Z_{sK}

Number of loads in test case	1	10	100	1000
4	0.24	0.22	0.06	0
9	0.72	0.42	0.39	0.34
13	0.76	0.49	0.38	0.27
18	1	0.8	0.63	0.51
29	0.99	1	1.1	0.75

Figure 35: nRF of radial network reconstructions from $\mathbf{Y}_K$ (top) and $\mathbf{Z}_{sK}$ (bottom). The subKron method leads to lower reconstruction error than the Kron method across test case sizes for SNRs $10 - 1000$. Performance is comparable for SNR $= 1$. nRF values are averaged over 5 trials.

> outperforms Kron-based estimation, all impedance estimation techniques become increasingly noise sensitive as network size grows. Also, crg execution time increases nonlinearly with reconstruction dimension. These practical limits create SNR-dependent limits on the size of networks that can be estimated.

The main usability limitation of this approach—and indeed this class of approaches—is the onerous sensor coverage demands. PMU voltage and current measurements at *every* active node in a distribution network—which in practice is the majority of network nodes—is a formidable ask. Reaching this level of coverage at the distribution level remains a distant dream. To reach usability, we must ease this measurement burden, while preserving the beneficial features of an underlying physics model. However, before striding in this direction, we must alleviate a common but significant assumption implicit in this approach. The tools described in this section assume a balanced three phase network that can be represented as a single phase system. Every nodes is assumed to have all three phases, and all measurements used are from one of the phases, with symmetric behavior assumed on the others. This limits the tool application to many distribution networks, which have unbalanced behavior across phases, single or two phase laterals and many nodes with some subset of all three phases. We extend physics-based topology estimation to this unbalanced three phase scenario in the next section.

4.4 EXTENSION TO THREE PHASE NETWORKS

This section presents a **g**reedy algorithm for joint **p**hase and **t**opology identification, termed *gpt*, which extends topology identification to generalized, unbalanced radial distribution networks. Like the approach in Section 4.3, gpt is rooted in a physics model, lending similar benefits of theoretical justification and precise assumptions. Unlike the tools of Section 4.3, gpt recovers only the network connectivity without impedances, but also has lower data requirements, demanding only voltage measurements at all nodes. The gpt algorithm surpasses the prior work in topology and phase estimation in the following respects.

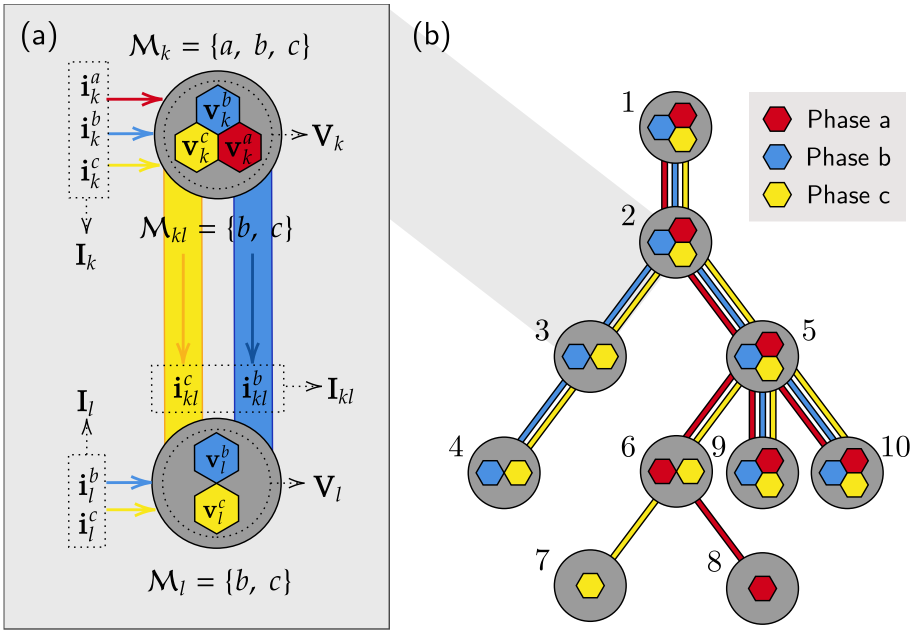


Figure 36: (a) Notation visualized for two nodes and connecting line. (b) *ToyNet*: A toy network with three, two, and one phase nodes used as a running example.

- It is provably correct under realistic assumptions and runs in polynomial time.
- It is applicable to real distribution networks, where some buses may have a subset of all three phases.
- It requires *only* voltage measurements; either phasors or magnitudes (with slight differences in performance).
- It utilizes voltage *statistics* enabling successful phase identification even from voltage magnitudes alone and in the presence of phase shifting transformers [152].

Based on a novel, linearized, multi-phase power flow model mapping nodal current injections to nodal voltage phasors, gpt requires that the network is radial with diagonally dominant line impedance matrices to guarantee correctness. When phases are known, gpt reduces to provable, greedy multi-phase topology learning generalizing prior work in [130] for the balanced setting. When topology is known apriori, it reduces to a lo-

cal approach to phase identification. We demonstrate gpt's performance and improvement over prior work in both phase and topology recovery on multiple IEEE test networks simulated in Open-DSS, an open source distribution system simulator that can handle unbalanced networks [153].
This section is organized into multiple sub-parts. Section 4.4.1 presents a linearized model for unbalanced, three-phase networks with missing phases which is the theoretical basis of gpt. Section 4.4.2 theoretically derives and justifies a proximity metric for phase identification. Section 4.4.3 justifies a distance metric for topology identification once phases have been identified. Section 4.4.4 puts the prior parts together to propose gpt: an algorithm for joint phase and topology recovery. Finally, Section 4.4.5 presents validation of gpt on non-linear voltage data for three IEEE distribution test networks, simulated in OpenDSS. We compare gpt's performance with algorithms in prior work, demonstrating that gpt outperforms the prior work, and is robust to non-ideal measurements.

Notation

Notation is summarized below and visualized in Fig. 36(a). While overall consistent with other sections, the notation here is more elaborate, reflecting the complexity of three phase, unbalanced networks.

$\mathcal{N}$	Node set	$\mathcal{E}$	Edge set
$\mathbf{V}_k = \begin{bmatrix} \mathbf{v}_k^a \\ \mathbf{v}_k^b \\ \mathbf{v}_k^c \end{bmatrix}$	Node k voltages	$\mathbf{I}_k = \begin{bmatrix} \mathbf{i}_k^a \\ \mathbf{i}_k^b \\ \mathbf{i}_k^c \end{bmatrix}$	Node k current injections
$\mathcal{M}_k$	Node k phases	$\mathcal{M}_{kl}$	Line kl phases
$\mathbf{V}$	All nodal voltages	$\mathbf{I}$	All nodal injections
$\mathbf{Y}_{kl}$	Multiphase admittance matrix of line kl	$\mathbb{Y}$	Network admittance matrix
$\mathbb{Y}_{k,l}$	$\mathbb{Y}$ block for nodes k & l	$\mathbb{Y}_{k,l}^{\phi,\psi}$	$\mathbb{Y}$ element for phase ϕ at node k & phase ψ at node l
A^T	Transpose	A^H	Conjugate transpose

4.4.1 Unbalanced Three Phase Model

Before introducing the unbalanced three phase network model, it is helpful to review the *single phase balanced* power flow model. In the single phase case, each line ij has an associated scalar admittance $\mathbf{y}_{ij}$ and impedance $\mathbf{z}_{ij} = \mathbf{y}_{ij}^{-1}$. The single phase voltage phasors and current injections are related by the $(|\mathcal{N}| \times |\mathcal{N}|)$ system *admittance matrix* $\mathbb{Y}$ as $\mathbf{I} = \mathbb{Y}\mathbf{V}$ with the form:

$$\mathbb{Y}_{i,j} = -\mathbf{y}_{ij}, \;\; \mathbb{Y}_{i,i} = \sum_{ij \in \mathcal{E}} \mathbf{y}_{ij} \Rightarrow \mathbb{Y} = A\mathbf{D}A^{T}$$

$\mathbb{Y}$ can be factored into $|\mathcal{N}| \times |\mathcal{E}|$ *incidence* matrix A and diagonal line impedance matrix $\mathbf{D}$ [154]. Without loss of generality, we choose all edges to be directed away from the network “root”, generally the point of common coupling (PCC) or substation. If edge ij is oriented from i to j, the corresponding elements of A are

$$A_{i,ij} = 1, \;\; A_{j,ij} = -1, \;\; A_{r,ij} = 0 \text{ if } r \neq i \neq j$$

where $A_{i,ij}$ is the i^{th} element of the column corresponding to edge ij. By definition, as $\mathbb{Y}1 = 0$, $\mathbb{Y}$ is not invertible. An invertible reduced admittance matrix, $\underline{\mathbb{Y}}$, is constructed by choosing a reference node r and removing the corresponding row and column of $\mathbb{Y}$ (recall the transformation from the Kron to subKron matrices in Section 4.3). Since the system is lossless, its inverse relates voltages and currents as follows:

$$\underline{\mathbb{Z}} = \underline{\mathbb{Y}}^{-1}, \;\; \underline{\mathbf{V}} = \underline{\mathbb{Z}}\underline{\mathbf{I}}$$

$\underline{\mathbf{V}}$ contains voltages differences to the reference voltage while $\underline{\mathbf{I}}$ contains current injections at non-reference nodes. Let $\mathcal{E}_i$ and $\mathcal{E}_j$ denote the edge sets on the unique path in the radial system to r from nodes i and j respectively. The value of $\underline{\mathbb{Z}}_{i,j}$ is given by :

$$\underline{\mathbb{Z}}_{i,j} = \sum_{kl \in (\mathcal{E}_i \cap \mathcal{E}_j)} \mathbf{z}_{kl}.$$

Thus, the elements of $\underline{\mathbb{Z}}$ correspond to the impedances of common paths between node pairs and the reference [114], [130].

The *unbalanced three phase model* follows from the single phase one. To clarify definitions, we use *ToyNet* (Fig. 36b), a simple, unbalanced, three phase radial network, as a running example. We begin with the model for a multiphase line ij, with phases $\mathcal{M}_{ij} \subseteq \{a, b, c\}$. The voltage across ij is related to the current along each phase of the line by line impedance matrix $\mathbf{Y}_{ij}$:

$$\mathbf{I}_{ij} = \mathbf{Y}_{ij}(\mathbf{V}_i - \mathbf{V}_j) \tag{25}$$

$\mathbf{Y}_{ij}$ is the inverse of the $(|\mathcal{M}_{ij}| \times |\mathcal{M}_{ij}|)$ line impedance matrix, $\mathbf{Z}_{ij}$. Eq. (25) for line 56 in ToyNet is:

$$\begin{bmatrix} \mathbf{i}_{56}^{a} \\ \mathbf{i}_{56}^{c} \end{bmatrix} = \mathbf{Y}_{56} \begin{bmatrix} \mathbf{v}_5^{a} - \mathbf{v}_6^{a} \\ \mathbf{v}_5^{c} - \mathbf{v}_6^{c} \end{bmatrix}$$

The node i current injection, denoted $\mathbf{I}_i$, is a vector of injections on each phase of i, and is given by the sum of line flows: $\mathbf{I}_i = \sum_{ij \in \mathcal{E}} \mathbf{I}_{ij}$. Building up from the current-voltage relations across individual lines in (25), the multi-phase voltages and currents injections across the network are related by $\mathbf{I} = \hat{\mathbb{Y}}\mathbf{V}$. Note this model can describe a network with a subset of phases at some nodes. $\hat{\mathbb{Y}}$ is the multi-phase system admittance matrix with dimensions $(\sum_{i \in \mathcal{N}} |\mathcal{M}_i|) \times (\sum_{i \in \mathcal{N}} |\mathcal{M}_i|)$. The i, j *block* of $\hat{\mathbb{Y}}$ is

$$\hat{\mathbb{Y}}_{i,j} = -\mathbf{Y}_{ij}, \;\; \hat{\mathbb{Y}}_{i,i} = \sum_{ij \in \mathcal{E}} \mathbf{Y}_{ij} \tag{26}$$

Block $\hat{\mathbb{Y}}_{i,j}$ is $(|\mathcal{M}_i| \times |\mathcal{M}_j|)$, so $\mathbf{Y}_{ij}$ must be appropriately zero-padded or reduced if i and j don't have all the same phases. For ToyNet, $\hat{\mathbb{Y}}$ has the structure visualized in Fig. 37.

Remark 1. *$\hat{\mathbb{Y}}$ can be factored into an incidence matrix $\hat{\mathrm{A}}$, which captures the endpoints of each edge, and $\hat{\mathbf{D}}$, a block diagonal matrix of line admittances: $\hat{\mathbb{Y}} = \hat{\mathrm{A}}\hat{\mathbf{D}}\hat{\mathrm{A}}^{\mathsf{T}}$. $\hat{\mathbf{D}}$ has dimensions $(\sum_{ij \in \mathcal{E}} |\mathcal{M}_{ij}|) \times (\sum_{ij \in \mathcal{E}} |\mathcal{M}_{ij}|)$, with line admittance matrices $\mathbf{Y}_{ij}$ along the diagonal. $\hat{\mathrm{A}}$ is $(\sum_{i \in \mathcal{N}} |\mathcal{M}_i|) \times (\sum_{ij \in \mathcal{E}} |\mathcal{M}_{ij}|)$ dimensional. Its rows correspond to phases at each bus, and columns to phases of each edge. With edges directed toward the root, assume edge $ij \in \mathcal{E}$ is oriented from i to j. Then for every $ij \in \mathcal{E}$, with $\phi \in \mathcal{M}_{ij}$: $\hat{\mathrm{A}}_{i,ij}^{\phi\phi} = 1$, $\hat{\mathrm{A}}_{j,ij}^{\phi\phi} = -1$. All other elements of $\hat{\mathrm{A}}$ are zero. $\hat{\mathrm{A}}$ for ToyNet is visualized in Fig. 37.*

Inverting the model

$\hat{\mathbb{Y}}$ maps voltages to current injections, but for phase and topology identification we need the inverse mapping. By definition (26), $\hat{\mathbb{Y}}$ is singular.

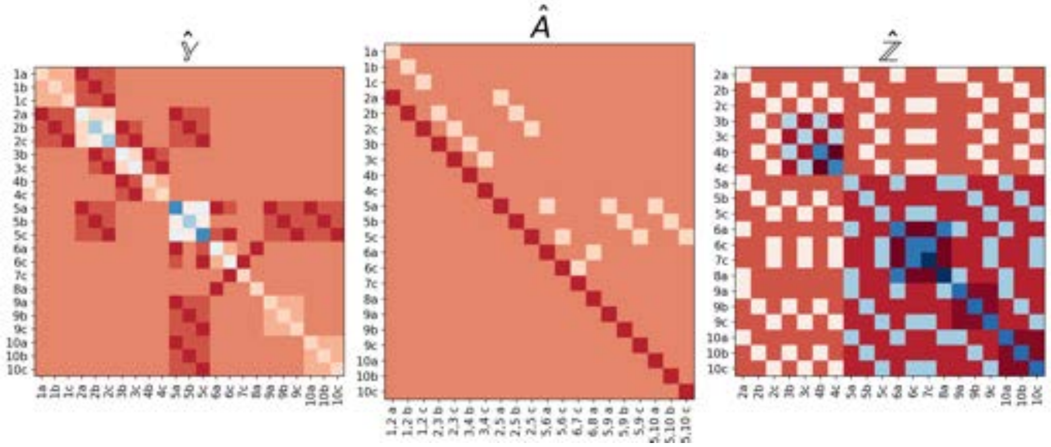


Figure 37: Visualizing the structure of three phase admittance, adjacency and impedance matrices for ToyNet.

Again, a reduction denoted $\underline{\hat{\mathbb{Y}}}$ is invertable. To obtain $\underline{\hat{\mathbb{Y}}}$, we remove the *three* rows and columns of $\hat{\mathbb{Y}}$ corresponding to the three phases at reference node r. $\underline{\hat{\mathbb{Y}}}$ can be factored as: $\underline{\hat{\mathbb{Y}}} = \underline{A}\hat{\mathbf{D}}\underline{A}^{\mathsf{T}}$ where $\underline{A}$ is obtained from $\hat{A}$ by deleting the three rows corresponding to r.

To derive $\underline{\hat{\mathbb{Y}}}^{-1}$, we begin with the right pseudoinverse of $\underline{A}^{\mathsf{T}}$, which has the following properties.

Lemma 8. *Let $\underline{B}$ be the right pseudoinverse of $\underline{A}^{\mathsf{T}}$, with rows corresponding to nodes and columns to edges. For $i \in \mathcal{N}$ with phases $\mathcal{M}_i$, let $\mathcal{E}_i$ be the edge set of the unique path to r. Then,*

$$\underline{B}^{\phi\psi}_{i,kl} = \begin{cases} -1 & \forall \phi = \psi \in \mathcal{M}_i,\ \forall kl \in \mathcal{E}_i \\ 0 & \text{otherwise} \end{cases}$$

Proof. $\underline{B}^{\phi}_{i}$ is the row of $\underline{B}$ corresponding to phase ϕ at node i, while $\underline{B}^{\phi}_{,ij}$ is the column corresponding to phase ϕ of edge ij. If $\phi \neq \psi$, $\underline{B}^{\phi\mathsf{T}}_{,ij}\underline{A}^{\psi}_{,kl} = 0$. A column of $\underline{A}$ has only two nonzero elements, so for $\phi = \psi$, we have

$$\underline{B}^{\phi\mathsf{T}}_{,ij}\underline{A}^{\phi}_{,kl} = -1\delta(ij \in \mathcal{E}_k) + 1\delta(ij \in \mathcal{E}_l)$$

For any edge $kl \neq ij$, we will have either $(ij \in \mathcal{E}_k), (ij \in \mathcal{E}_l)$ *or* $(ij \notin \mathcal{E}_k), (ij \notin \mathcal{E}_l)$. Thus $\underline{B}^{\phi\mathsf{T}}_{,ij}\underline{A}^{\phi}_{,kl} = 0$ for $kl \neq ij$. If $kl = ij$, we have $(ij \notin \mathcal{E}_k), (ij \in \mathcal{E}_l)$ and $\underline{B}^{\phi\mathsf{T}}_{,ij}\underline{A}^{\phi}_{,ij} = 1$. Thus, $\underline{B}^{\phi\mathsf{T}}_{,ij}\underline{A}^{\psi}_{,kl} = 1$ iff $ij = kl, \phi = \psi \Rightarrow \underline{B}^{\mathsf{T}}\underline{A} = I$.

Now consider, $\underline{B}_i^{\phi}\underline{A}_j^{\psi T}$, the inner product of rows. If $\phi \neq \psi$, this is 0. Consider when $\phi = \psi$ and $i \neq j$. If j does not lie along the path from i to the reference, then $jk \notin \mathcal{E}_i$, i.e. there is no edge connected to j in $\mathcal{E}_i$, and $\underline{B}_i^{\phi}\underline{A}_j^{\phi T} = 0$. In contrast, if j lies along the path from i to the reference, there *must* be two edges $kj, jl \in \mathcal{E}_i$ oriented to and away from j respectively, as the path passes through j. Then, $\underline{B}_i^{\phi}\underline{A}_j^{\phi T} = (-1 \times -1) + (1 \times -1) = 0$. If $i = j$, only edge $il = jl \in \mathcal{E}_i$, and $\underline{B}_i^{\phi}\underline{A}_i^{\phi T} = 1$. Therefore, $\underline{B}_i^{\phi}\underline{A}_j^{\psi T} = 1$ iff $i = j, \phi = \psi$ □

Therefore, $\hat{\underline{\mathbb{Y}}}^{-1} = \hat{\underline{\mathbb{Z}}}$ can be written as follows.

Theorem 4. *The inverse of* $\hat{\underline{\mathbb{Y}}}$ *is given by:*

$$\hat{\underline{\mathbb{Z}}} = \hat{\underline{\mathbb{Y}}}^{-1} = \underline{B}\hat{\mathbf{D}}^{-1}\underline{B}^T \tag{27}$$

where $\hat{\mathbf{D}}^{-1}$ *is block diagonal matrix of line impedance matrices:* $\hat{\mathbf{D}}^{-1}_{ij,ij} = \mathbf{Z}_{ij}$. *Further, the element of* $\hat{\underline{\mathbb{Z}}}$ *corresponding to phase* ϕ *at node* i *and phase* ψ *at node* j *is given by:*

$$\hat{\underline{\mathbb{Z}}}_{ij}^{\phi\psi} = \sum_{kl \in (\mathcal{E}_i \cap \mathcal{E}_j)} \mathbf{Z}_{kl}^{\phi\psi} \tag{28}$$

Proof. Using the structure of $\underline{B}$ from Lemma 8, we have

$$\hat{\underline{\mathbb{Z}}}\hat{\underline{\mathbb{Y}}} = (\underline{B}\hat{\mathbf{D}}^{-1}\underline{B}^T)(\underline{A}\hat{\mathbf{D}}\underline{A}^T) = I.$$

Thus $\hat{\underline{\mathbb{Z}}} = \hat{\underline{\mathbb{Y}}}^{-1}$.
Consider the block of $\hat{\underline{\mathbb{Z}}}$ corresponding to nodes i and j. Based on Lemma 8:

$$\hat{\underline{\mathbb{Z}}}_{ij}^{\phi\psi} = \sum_{kl \in \mathcal{E}} \underline{B}_{i,kl}^{\phi\phi}\mathbf{Z}_{kl}^{\phi\psi}\underline{B}_{j,kl}^{\psi\psi T} \qquad = \sum_{kl \in (\mathcal{E}_i \cap \mathcal{E}_j)} \mathbf{Z}_{kl}^{\phi\psi}$$

□

Intuitively, (28) says that a change in current injection on phase ψ at node j will affect the voltage at phase ϕ at node i, proportional to the $(\phi\psi)$ impedance of the shared path $(\mathcal{E}_i \cap \mathcal{E}_j)$ from i, j to r. In our definition, $\hat{\underline{\mathbb{Y}}}$ and $\hat{\underline{\mathbb{Z}}}$ are ordered with the phases of each node or edge grouped together.

If all phases exist at all nodes, this is equivalent to a permutation of the three-phase model in [134], where entries for one phase across all nodes and edges are grouped together.
Using this theorem, in the unbalanced three phase model, voltages are related to currents as:

$$\underline{\mathbf{V}} = \underline{\hat{\mathbb{Z}}\mathbf{I}} \tag{29}$$

Here, $\underline{\mathbf{V}}$ contains nodal voltage *differences* with the voltage for the *matching phase* at the reference: $\underline{\mathbf{V}}_i^\phi = \mathbf{v}_i^\phi - \mathbf{v}_r^\phi$. $\underline{\mathbf{I}}$ contains all current injections except at the reference. Note that there is no assumption for all nodes having all three phases. In the next section, we will use the model of (29) to determine patterns in voltage statistics to enable phase and topology recovery.

Current and Voltage Statistics

We treat voltages as random variables driven by current via the model of (29). In our theoretical analysis, we assume

1. current injections are uncorrelated across nodes and phases (including at a single node).

$$\mathrm{cov}(\mathbf{i}_i^\phi, \mathbf{i}_j^\psi) \neq 0 \text{ iff } (i = j) \cap (\phi = \psi) \tag{30}$$

2. current injections have equal variance at all nodes.

$$\forall i, \phi : \; var(\mathbf{i}_i^\phi) = s^2 \tag{31}$$

As they are predominantly determined by loads—which are uncorrelated over time intervals on the order of seconds—current injections can be modeled as uncorrelated across nodes and phases when using high resolution measurements such as from PMUs. We assume PMUs report at 120Hz, but our methods apply if resolution is sufficient for measurements to be detrended to remove inter-nodal correlations. Assumption (31) is stronger but permissible in reasonably balanced networks. In Section 4.4.5, we evaluate how deviations from these assumptions impact recovery performance.

4.4.2 Voltage Covariance for Phase Matching

Voltage covariances are informative for phase identification. Under Assumptions (30,31), the covariance of the voltage of phase ϕ at node i and phase ψ at node j, in the three-phase model (29), is given by:

$$\mathrm{cov}(\mathbf{v}_i^{\phi}, \mathbf{v}_j^{\psi}) = \mathrm{cov}(\hat{\underline{\mathbb{Z}}}_i^{\phi}\underline{\mathbf{I}}, \hat{\underline{\mathbb{Z}}}_j^{\psi}\underline{\mathbf{I}}) = s^2\mathrm{Re}((\hat{\underline{\mathbb{Z}}}_i^{\phi})^H\hat{\underline{\mathbb{Z}}}_j^{\psi}) \tag{32}$$

We are interested in the sum of covariances for a particular *phase ordering* between nodes i and j. Consider the case where the phases at i are a subset of those at j ($\mathcal{M}_i \subseteq \mathcal{M}_j$). Let $\mathcal{O}$ denote the ordering/permutation of phases at j, where $\mathcal{O}(\phi)$ denotes the specific phase at j matched to the phase ϕ at i. Then, the covariance sum for matching $\mathcal{O}$, denoted by $c_{ij}^{\mathcal{O}}$, is:

$$c_{ij}^{\mathcal{O}} = \sum_{\phi\in\mathcal{P}_i} \mathrm{cov}(\mathbf{v}_i^{\phi}, \mathbf{v}_j^{\mathcal{O}(\phi)}) \tag{33}$$

Let $\hat{\underline{\mathbb{Z}}}_i$ denote the rows of $\hat{\underline{\mathbb{Z}}}$ corresponding to *all* phases at node i, and $\hat{\underline{\mathbb{Z}}}_j^{\mathcal{O}}$ denote the rows corresponding to the phases at j ordered according to $\mathcal{O}$. Then $c_{ij}^{\mathcal{O}}$ is:

$$\begin{aligned} c_{ij}^{\mathcal{O}} &= s^2\mathrm{Re}(vec(\hat{\underline{\mathbb{Z}}}_i)^H vec(\hat{\underline{\mathbb{Z}}}_j^{\mathcal{O}})) = s^2\mathrm{Re}(\sum_{k\in\mathcal{N}} vec(\hat{\underline{\mathbb{Z}}}_{ik}^H) vec(\hat{\underline{\mathbb{Z}}}_{jk}^{\mathcal{O}})) \\ &= s^2\mathrm{Re}\left[\sum_{k\in\mathcal{N}} \sum_{\substack{mn\in(\mathcal{E}_i\cap\mathcal{E}_k)\\ pq\in(\mathcal{E}_j\cap\mathcal{E}_k)}} vec(\mathbf{Z}_{mn}^{\mathcal{M}_i})^H vec(\mathbf{Z}_{pq}^{\mathcal{O}})\right] \end{aligned} \tag{34}$$

The last equality follows from (28). The contribution of a node k to $c_{ij}^{\mathcal{O}}$ is the dot product of the common path lengths between i, k and j, k. The following result shows how $c_{ij}^{\mathcal{O}}$ enables phase matching.

Theorem 5. *Consider $c_{ij}^{\mathcal{O}}$ given in (33) for $\mathcal{M}_i \subseteq \mathcal{M}_j$. If condition (35) holds for each pair of line impedance matrices, then $c_{ij}^{\mathcal{O}}$ is maximized when $\mathcal{O}$ corresponds to the correct phase matching between i and j.*

$$\begin{gathered} \forall \mathcal{M} \in \{\mathcal{M}_1, \ldots, \mathcal{M}_n\},\ \forall st, kl \in \mathcal{E} : \\ \mathcal{M} = \arg\max_{\mathcal{O}} \mathrm{Re}\left[vec(\mathbf{Z}_{st}^{\mathcal{M}})^H vec(\mathbf{Z}_{kl}^{\mathcal{O}(\mathcal{M})})\right] \end{gathered} \tag{35}$$

where $\mathcal{M}$ ranges over every nodal phase set $\mathcal{M}_i$.

Proof. If (35) holds, *every* term in the summation in (34) is maximized by $\mathcal{O} = \mathcal{M}_i$. Therefore, $\mathcal{M}_i$ maximizes the sum, and $\mathcal{M}_i = \arg\max_{\mathcal{O}} c_{ij}^{\mathcal{O}}$ □

(35) is a condition on every pair of edges, st and kl, in the network. It states that for every row subset of line impedance matrix $\mathbf{Z}_{st}$ (corresponding to each nodal phase set $\mathcal{M}_1, ..., \mathcal{M}_n$), the matching rows of $\mathbf{Z}_{kl}$ produce the largest vectorized dot product. This is reasonable as real line impedance matrices are diagonally dominant. Condition (35) depends on the particular network considered. In a network where all nodes have all three phases ($\mathcal{M} = \mathcal{M}_i = \{a, b, c\}$) the condition on the vectorized dot product involves the full three phase line impedance matrices. if some nodes have a subset of phases, it will involve sub-matrices of impedance matrices. Note that, in general, $cov(\mathbf{v}_i^{\phi}, \mathbf{v}_j^{\psi}) > 0$ even if $\phi \neq \psi$. If node i has phases a, b and node j has phases a, c ($\mathcal{M}_i \not\subseteq \mathcal{M}_j$), minimizing c_{ij} will incorrectly match b, c. The gpt algorithm avoids such scenarios by ordering nodes; discussed later. In summary, Theorem 5 allows us to use $c_{ij}^{\mathcal{O}}$ as a proximity metric for phase matching.

4.4.3 Voltage Difference Variances for Topology

We use voltage difference variances for topology recovery. Define d_{ij} to be the sum of the variance of the voltage differences between correctly matched phases of nodes i, j. Assuming $\mathcal{M}_i \subseteq \mathcal{M}_j$, d_{ij} is:

$$d_{ij} = \sum_{\phi \in \mathcal{M}_i} var(\mathbf{v}_i^{\phi} - \mathbf{v}_j^{\phi}) = \sum_{\phi \in \mathcal{M}_i} \mathbb{E}[(\mathbf{v}_i^{\phi} - \mathbf{v}_j^{\phi}) - \mathbb{E}(\mathbf{v}_i^{\phi} - \mathbf{v}_j^{\phi})]^2 \tag{36}$$

Lemma 9 establishes trends in d_{ij} along one phase.

Lemma 9. *Given the voltage on phase* ϕ *at node* i*:*

$$\arg\min_j d_{ij}^{\phi} \triangleq \arg\min_j var(\mathbf{v}_i^{\phi} - \mathbf{v}_j^{\phi}) \in \text{Parent/Child of } i \tag{37}$$

Proof. Expanding the difference, we obtain:

$$d_{ij}^{\phi} = var(\mathbf{v}_i^{\phi} - \mathbf{v}_j^{\phi}) = \sum_{n \in \mathcal{N}} \sum_{\psi \in \mathcal{M}_n} s_n^2 |\hat{\mathbb{Z}}_{in}^{\phi\psi} - \hat{\mathbb{Z}}_{jn}^{\phi\psi}|^2 \tag{38}$$

where s_n is the injection variance at node n. If the paths from nodes k and l to r merge at node n, $\hat{\underline{\mathbb{Z}}}_{k,l}^{\phi,\psi} = \mathbf{e}_n^{\phi\psi}$, the impedance of the path from n to r along phase coupling ϕ, ψ:

$$\mathbf{e}_n^{\phi\psi} = \sum_{ij \in \mathcal{E}_n} \mathbf{Z}_{ij}^{\phi\psi} \tag{39}$$

To determine the minimizer of (37), consider two cases visualized in Fig. 38. In case A, j is the common ancestor of node i, k on the path to the root. In case B, j is an ancestor of i, while k is an ancestor of j. In both cases, we show that $d_{ij}^{\phi}, d_{jk}^{\phi} < d_{ik}^{\phi}$. Put together, for a given i, the minimizer j of d_{ij}^{ϕ} is either the parent or child of i.

Case A. We split the sum in (38) into the regions $\mathcal{N}_i$ in Fig. 38a. Using (39) in (38) for each region, we have

$$\begin{aligned} d_{ik}^{\phi} - d_{ij}^{\phi} = & \sum_{n \in \mathcal{N}_1, \psi \in \mathcal{M}_n} 0 + \sum_{n \in \mathcal{N}_2, \psi \in \mathcal{M}_n} 0 + \sum_{n \in \mathcal{N}_4, \psi \in \mathcal{M}_n} 0 \\ & + \sum_{n \in \mathcal{N}_3, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_j^{\phi\psi} - \mathbf{e}_n^{\phi\psi}|^2 - |\mathbf{e}_j^{\phi\psi} - \mathbf{e}_j^{\phi\psi}|^2 \right) \\ & + \sum_{n \in \mathcal{N}_5, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_j^{\phi\psi} - \mathbf{e}_k^{\phi\psi}|^2 - |\mathbf{e}_j^{\phi\psi} - \mathbf{e}_j^{\phi\psi}|^2 \right) > 0 \end{aligned}$$

A similar argument shows $d_{ik}^{\phi} - d_{kj}^{\phi} > 0$.

Case B. Now we split (38) over the regions in Fig. 38b. Using (39), we have

$$\begin{aligned} d_{ik}^{\phi} - d_{ij}^{\phi} = & \sum_{n \in \mathcal{N}_1, \psi \in \mathcal{M}_n} 0 + \sum_{n \in \mathcal{N}_2, \psi \in \mathcal{M}_n} 0 \\ & + \sum_{n \in \mathcal{N}_3, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_n^{\phi\psi} - \mathbf{e}_k^{\phi\psi}|^2 - |\mathbf{e}_n^{\phi\psi} - \mathbf{e}_n^{\phi\psi}|^2 \right) \\ & + \sum_{n \in \mathcal{N}_4, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_j^{\phi\psi} - \mathbf{e}_k^{\phi\psi}|^2 - |\mathbf{e}_j^{\phi\psi} - \mathbf{e}_j^{\phi\psi}|^2 \right) \\ & + \sum_{n \in \mathcal{N}_5, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_n^{\phi\psi} - \mathbf{e}_k^{\phi\psi}|^2 - |\mathbf{e}_n^{\phi\psi} - \mathbf{e}_j^{\phi\psi}|^2 \right) \\ & + \sum_{n \in \mathcal{N}_6, \psi \in \mathcal{M}_n} s_n^2 \left(|\mathbf{e}_i^{\phi\psi} - \mathbf{e}_k^{\phi\psi}|^2 - |\mathbf{e}_i^{\phi\psi} - \mathbf{e}_j^{\phi\psi}|^2 \right) > 0 \end{aligned}$$

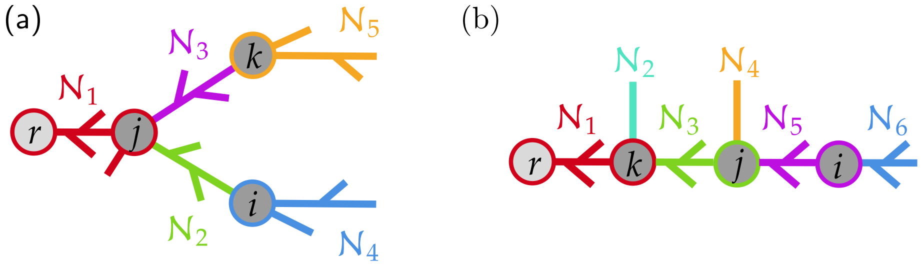


Figure 38: Regions of the radial network when (a) Case A: k lies off the path between i and the reference. (b) Case B: i lies along the path from j to the reference.

A similar analysis shows $d_{ik}^{\phi} - d_{kj}^{\phi} > 0$. Thus the minimum is given by the parent/child of i. $\square$

Applying Lemma 9 to all matched phases between two nodes gives the following result:

Theorem 6. *Given node* i*, the node* j *which minimizes* d_{ij} *in (36) is either a parent or child of* i.

If phases at each node are known, Thm. 6 enables correct topology recovery with a greedy algorithm based on distance d_{ij}. Note that Lemma 9 and Thm. 6 hold for all uncorrelated injections even with unequal variances. Thus, Assumption 31 can be relaxed for topology learning. We now have the tools for joint phase and topology recovery, detailed in the next section.

4.4.4 Joint Phase & Topology Identification

We propose gpt (Alg. 2): a *greedy* algorithm for joint phase and topology identification based on the nodal voltage properties of Sections 4.4.2-4.4.3. The gpt method computes $c_{ij}^{\mathcal{O}}$'s (32) exhaustively (for all matching options), selecting maxima for phase matching (Theorem 5). Based on phase matchings, it computes d_{ij}'s (36) exhaustively (for all node pairs), selecting minima for topology recovery (Theorem 6). Next, gpt greedily builds a tree with node set $\mathcal{T}$, starting from node i_0 and iterating till all nodes have been added. In each iteration, a new node is added to the tree by choosing node $i \notin \mathcal{T}$, which has the minimum value of d_{ij} for all $j \in \mathcal{T}$, using the *getNext* algorithm. When node i is added to the tree, it's phases are determined based on their matching to the phases of node j.

Note that gpt adds 3 phase, then 2 phase, then 1 phase nodes to the tree. The initial node must be three phase, making the reference an intuitive choice. By adding nodes in this order, gpt implicitly enforces the crucial fact that number of phases never increases moving from the substation to the network ends (a single phase node is never the parent of a three phase node) and avoids issues that can arise when applying a naive greedy algorithm to a network with a variable number of phases at each node. For example, suppose we are recovering the topology of *ToyNet*. All nodes have been added to $\mathcal{T}$ except 6, 7, and 8. To recover the correct topology, we should connect node 6 to 5 first. Then nodes 7 and 8 will get connected to 6 naturally, as $d_{76} < d_{75}$ and $d_{86} < d_{85}$. However, consider d_{65} and d_{75},

$$d_{65} = var(\mathbf{v}_6^a - \mathbf{v}_5^a) + var(\mathbf{v}_6^c - \mathbf{v}_5^c), \;\; d_{75} = var(\mathbf{v}_7^c - \mathbf{v}_5^c)$$

We have no guarantee that $d_{65} < d_{75}$ due to the presence of additional phase variance in d_{65} illustrating how an algorithm that doesn't order nodes by decreasing number of phases may return an incorrect topology, unlike gpt.

Alternative Estimation Scenarios

In the general scenario, gpt recovers both phase and topological connectivity from voltage measurements. Our theoretical results also establish estimation methods for restricted setting.

Phase Identification with Topology Information. If topology is known, phases can be identified by greedily matching adjacent nodes using $c_{ij}^{\mathcal{O}}$ (32) across edges $ij \in \mathcal{E}$. [67], [155], [156] similarly use the Pearson correlation coefficient of voltages as the distance, which is related to the covariance but not theoretically justified. Unlike [155]–[159] which cluster all nodal voltages to recover phase, gpt is highly local. K-means is a popular clustering algorithm choice [158]. However, even if the correct phase matching is the globally optimal solution of the k-means cost, the optimization is non-convex and may not converge to the global minima. Our greedy approach, however, is guaranteed to result in the optimal solution.

Topology Estimation with Phase Information. If phase labels are known, d_{ij} (36) can be directly minimized to recover topology and gpt reduces to greedy spanning tree learning generalizing prior work for the single phase case [130]. Compared to [133], [134] that use conditional independence tests and need matrix inversions, gpt has improved sample performance, as demonstrated in Section 4.4.5.

Estimation using voltage magnitudes only While gpt is based on nodal voltage phasors, it can also use voltage magnitudes $v_i^{\phi} = |\mathbf{v}_i^{\phi}|$. This is theoretically justified by linearizing (25) for line kl:

$$\begin{aligned} \mathbf{I}_{kl} &= \mathbf{Y}_{kl} \begin{bmatrix} e^{j\theta^a}(v_k^a e^{j\underline{\theta}_k^a} - v_l^a e^{j\underline{\theta}_l^a}) \\ e^{j\theta^b}(v_k^b e^{j\underline{\theta}_k^b} - v_l^b e^{j\underline{\theta}_l^b}) \\ e^{j\theta^c}(v_k^c e^{j\underline{\theta}_k^c} - v_l^c e^{j\underline{\theta}_l^c}) \end{bmatrix} \\ &\approx \mathbf{Y}_{kl}\mathbf{D}_r((V_k - V_l) + j(\underline{\theta}_k - \underline{\theta}_l)) \end{aligned}$$

where θ^{ϕ} is the phase ϕ reference angle and $\underline{\theta}_k^{\phi} = \theta_k^{\phi} - \theta^{\phi}$. The linearization assumes small magnitude deviations from the reference and small angle difference between neighboring nodes. Properties of voltage magnitudes across the network can then be derived under assumptions on $\mathbf{Y}_{kl}\mathbf{D}_r, \mathbf{I}_{kl}$ to obtain gpt for phase and topology recovery.

4.4.5 Simulation Experiments

We present simulation results of gpt. We measure average errors in phase and topology recovery, normalized by network size:

$$\text{Topology Error} = \frac{\text{wrong edges} + \text{missing edges}}{\text{total edges}},$$

$$\text{Phase Error} = \frac{\text{wrong nodal phases}}{\text{total nodal phases}}$$

Further, we evaluate the algorithm's sensitivity to the following parameters.

- *Measurement noise*: We add white noise $\mathbf{n}$ to original measurement $\mathbf{v}_i$: $\tilde{\mathbf{v}}_i = \mathbf{v}_i + \mathbf{n}$, defining noise level$(\tilde{\mathbf{v}}_i) = \frac{var(\mathbf{n})}{var(\mathbf{v}_i)}$. As gpt uses voltage covariances, it depends on relative precision and not absolute accuracy and is immune to the stable transducer errors that afflict distribution PMU data [160].

- *Number of measurement samples*: Assuming 120 Hz distribution PMU measurements [41], we record performance on 1 second to 1 minute of voltage data.
- *Load Correlations*: We test gpt's sensitivity to the assumption of uncorrelated injections, by varying the correlations of the loads while maintaining their variance. This is done by setting load covariance matrix $\Sigma = \sigma^2((1-\epsilon)\mathbb{I} + \epsilon \mathbf{1}\mathbf{1}^T)$. As $\epsilon \to 1$, injections become more correlated.

Three IEEE distribution test networks are simulated in OpenDSS: the 13 and 34 bus networks have some one and two phases buses, while the 37 bus network has all three phase buses [161]. We modify the models by adding loads at every bus, and by disabling voltage regulators, which invalidate the assumption of voltages driven by injections. We fluctuate the load injections at each phase at each bus, and simulate the network with power flow to obtain non-linear voltages.

Fig. 39 plots topology recovery accuracy for three noise levels ranging from 0 (no noise) to 10, with 1 second to 1 minute of voltage magnitude or phasor measurements. PMUs are highly precise; and the noise level would realistically be ~ 0.001 [17], [45]. Nevertheless, gpt performs well under more noise as measurement samples increase. For all test networks and measurement durations, gpt achieves perfect topology recovery from voltage magnitudes for 0 and 0.001 noise. Insets in Fig. 39 show recovered topologies across trials. Note how errors are localized to a few nodes, and lower for voltage magnitudes. Fig 39 also compares performance on voltage magnitudes to Liao2019 [133], showing that gpt outperforms it across scenarios.

Fig. 40 presents gpt's phase matching error on the same three networks and noise levels averaged across several sample durations (we found phase matching error to be invariant to sample duration). The error is compared to that of the approach in Olivier2018 [155]. The methods have comparable performance, except on the 13 bus network, where gpt outperforms Olivier2018 across SNRs.

Fig. 41 shows topology recovery sensitivity as injections stray from the uncorrelated assumption (30). Error increases rapidly as loads become more correlated. In reality, over short time durations, it is reasonable to assume that injections will be uncorrelated across nodes or can be de-trended [133]. We use at most one minute of data to recover phase and topology: short enough that the uncorrelated assumption should hold well.

The polynomial time gpt is suitable for real time application, taking on the order of seconds to recover phase and topology for the IEEE test cases shared here. On the largest 37 bus test case, the algorithm completes in 15 seconds.

* * *

This section presented gpt, an algorithm for joint phase and topological connectivity identification from voltage measurements in unbalanced three phase networks where each bus can have single, two, or three phases. The algorithm is grounded in a linear physics model of the system, and therefore has well-defined assumptions under which it generates provably correct results. By applying gpt to non-linear voltages from three test feeders simulated under realistic conditions, we also demonstrated the efficacy of gpt in practice. We showed that gpt is robust to measurement non-idealities, and outperforms the prior work in both phase and topology recovery.

In addition to the gpt algorithm itself, the novel, three phase linear model which forms its foundation is an important contribution for the development of additional estimation and monitoring tools in unbalanced networks.

While gpt demands fewer measurements (at the cost of impedance information) than the subKron method of Section 4.3, its sensor coverage requirements remain high for true usability. The following section describes a set of topology monitoring tools that push further in the direction of usability.

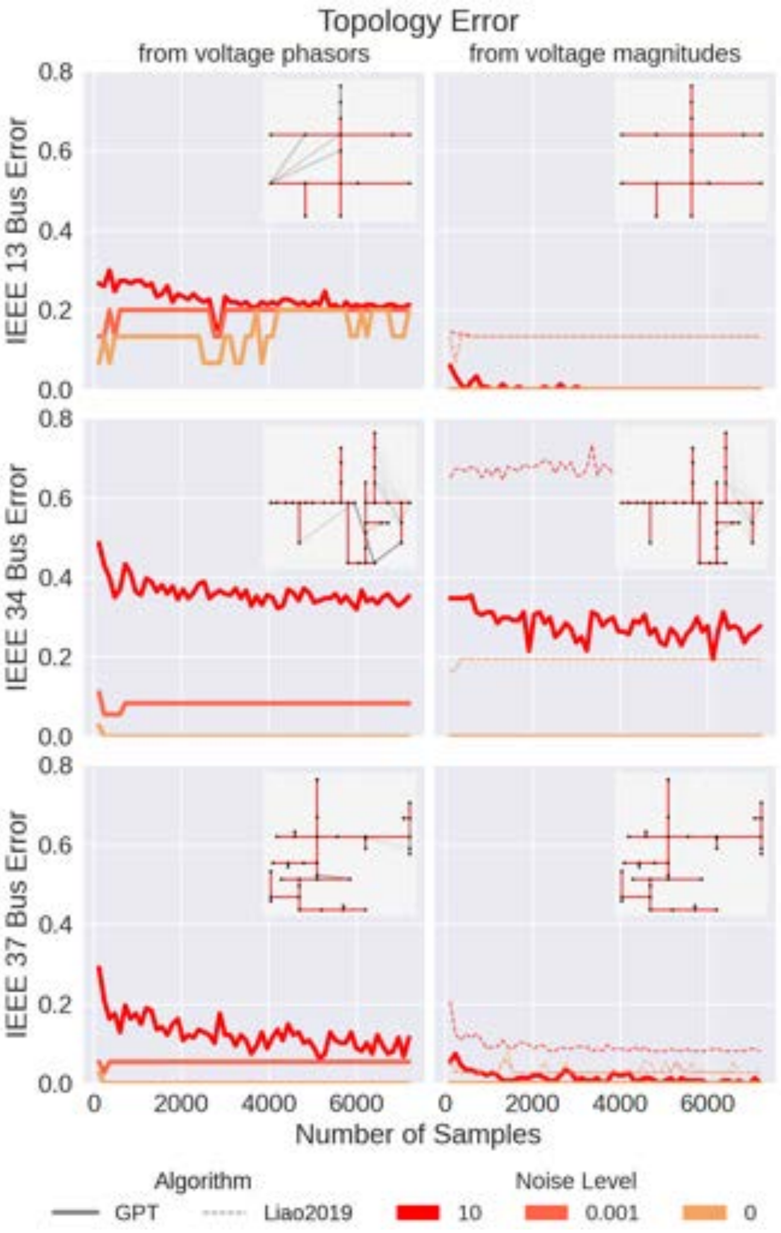


Figure 39: Plot of gpt topology error vs number of samples for three test feeders and three noise levels. Samples are assumed to arrive at 120Hz. Insets show estimated (gray) and true (red) network lines across trials, with the opacity of gray lines indicating how many times the edge was recovered. We evaluated gpt on voltage phasor and magnitude data, with performance on magnitudes compared to state of the art in Liao2019 [133].

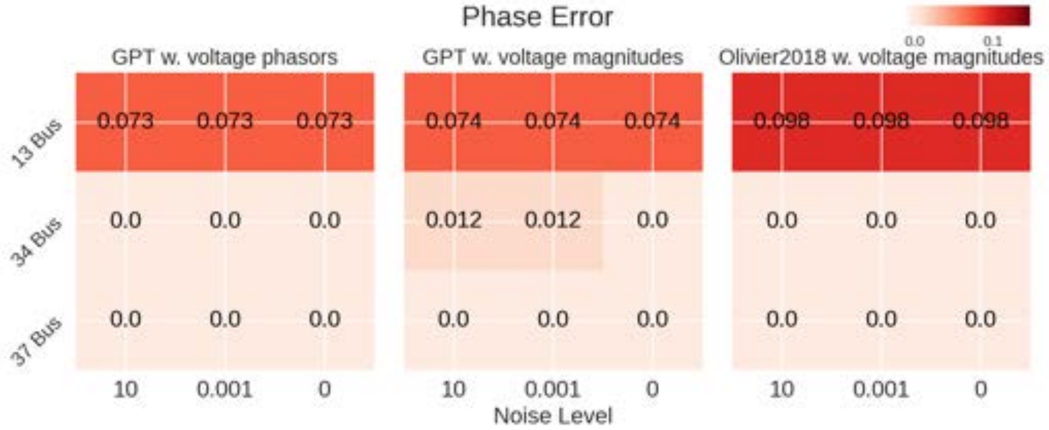


Figure 40: Plot of gpt phase matching error for three test feeders and three noise levels using voltage phasor and magnitude data. The last table allows comparison with the state of the art phase matching method of Olivier2018 [155]. Mostly, gpt has comparable performance to Olivier2018, but significantly outperforms it on the 13 bus system.

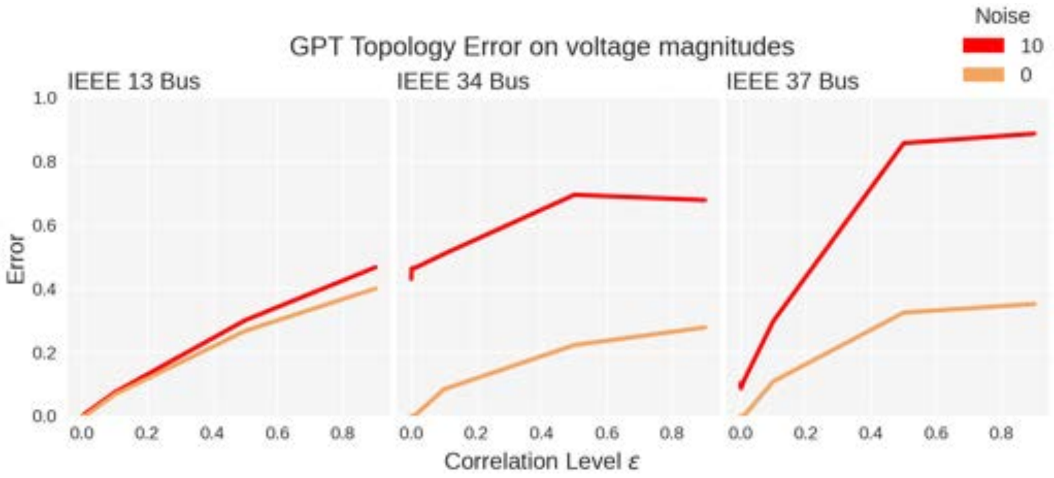


Figure 41: Plot of gpt topology error from voltage magnitudes as injections become increasingly correlated ($\epsilon \to 1$) for three test feeders and two noise levels. Notice the rapid rise in error with increasing correlation particularly for the largest network.

A 2: $[\mathcal{E}, \mathcal{P}] = \text{gpt}(\mathbf{V})$

```
Input  : $\mathbf{V}$ is a $(\sum_{i \in \mathcal{N}} |\mathcal{M}_i|) \times t$ matrix of voltage data (phasor or
         magnitude)
Output : $\mathcal{E}$ is the network edge set, $\mathcal{P}$ is the phase ordering of each node
$\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ // set of three, two, one phase nodes in $\mathcal{N}$
forall $i, j \in \binom{\mathcal{N}}{2}$ do
    $\forall \mathcal{O}$ compute $c_{ij}^{\mathcal{O}}$
    phase-match$(i, j) \leftarrow \arg\max_{\mathcal{O}} c_{ij}^{\mathcal{O}}$ // Eq. 33
    dists$(i, j) \leftarrow d_{ij}$ for phase-match$(i, j)$// Eq. 36
end
// Add first three phase node to tree $\mathcal{T}$ and set phases
$\mathcal{T} \leftarrow \{i_0\}$// Initialize tree
$\mathcal{P}(i_0) \leftarrow \{a, b, c\}$// Initialize phases
$\mathcal{N}_3 \leftarrow \mathcal{N}_3 \setminus i_0$
while $\mathcal{N}_3 \neq \emptyset$ do
    // Add three phase nodes
    $i, j \leftarrow$ get-next(dists, $\mathcal{T}$, $\mathcal{N}_3$)// Call to Alg. 3
    $\mathcal{N}_3 \leftarrow \mathcal{N}_3 \setminus i$// Remove i from unconnected set
    $\mathcal{T} \leftarrow \mathcal{T} \cup i$// Add i to tree
    $\mathcal{E} \leftarrow \mathcal{E} \cup e_{ij}$// Add edge between i & j
    $\mathcal{P}(i) \leftarrow$ phase-match(i, j)// Record i's phase matching to j
end
while $\mathcal{N}_2 \neq \emptyset$ do
    // Add two phase nodes
    $i, j \leftarrow$ get-next(dists, $\mathcal{T}$, $\mathcal{N}_2$)
    $\mathcal{N}_2 \leftarrow \mathcal{N}_2 \setminus i$,
    $\mathcal{T} \leftarrow \mathcal{T} \cup i$, $\mathcal{E} \leftarrow \mathcal{E} \cup e_{ij}$, $\mathcal{P}(i) \leftarrow$ phase-match(i, j)
end
while $\mathcal{N}_1 \neq \emptyset$ do
    // Add one phase nodes
    $i, j \leftarrow$ get-next(dists, $\mathcal{T}$, $\mathcal{N}_1$)
    $\mathcal{N}_1 \leftarrow \mathcal{N}_1 \setminus i$,
    $\mathcal{T} \leftarrow \mathcal{T} \cup i$, $\mathcal{E} \leftarrow \mathcal{E} \cup e_{ij}$, $\mathcal{P}(i) \leftarrow$ phase-match(i, j)
end
```

A 3: $[i, j]$ = get-next(dists, $\mathcal{T}$, $\mathcal{N}$)

```
Input  : dists contains pairwise distances between all nodes
         $\mathcal{T}$ is the node set in the tree so far
         $\mathcal{N}$ is the nodes to be added
Output : $i$ is the node from $\mathcal{N}$ to be added to $\mathcal{T}$
         $j$ is the node in $\mathcal{T}$ connected to $i$
// Find node in $\mathcal{N}$ nearest to any node in $\mathcal{T}$
$d_{ij} \leftarrow \infty$, $i \leftarrow$ none, $j \leftarrow$ none
forall $b \in \mathcal{T}$, $a \in \mathcal{N}$ do
    if $\text{dists}(a, b) < d_{ij}$ then
        $d_{ij} \leftarrow \text{dists}(a, b)$
        $i \leftarrow a$, $j \leftarrow b$
    end
end
```

4.5 JUSTIFIED HEURISTICS

The previous sections described heuristic and physics-based tools for topology monitoring. Each tool had its strengths, but fell short of the criteria of usability in one or more key respects. This section describes a set of topology tools termed "justified heuristics", which stride toward usability by attempting to meld the benefits of underlying physics models with the low and flexible sensor requirements of heuristic approaches. The section begins by establishing physics-based theory connecting the mathematical properties of voltage measurements in general—including rank, correlations, and clusters—to network topology. This enables the theoretical justification of existing heuristic approaches—clarifying what they reveal of the system's physics and the implicit assumptions involved—and the proposal of novel analytic tools. The proposed tools generate continuous outputs targeted at human users. These outputs will manifest significant changes if there are underlying transformations in the physical system. However, the outputs will also change more marginally due to noise—either arising from the input data or due to system deviations from the tools' implicit assumptions. Human users can and must distinguish these types of changes, enabling the proposed analytics to be successful even if their underlying assumptions do not perfectly hold at all times or if there is some noise in the inputs. With the same intent, the outputs of these analytics can also be passed to statistical baselining methods before being relayed to human users, as described briefly in Chapter 3 and discussed further in Chapter 6. Beyond specific analytics, the results of this section lay the groundwork for improved understanding and use of grid voltage measurements in a range of applications, including topology identification, phase identification, anomaly detection, and missing data recovery. The results enable an otherwise overwhelming amount of voltage data to be transformed into intuitive, visualizable system information with relatively simple algorithms. This work is similar in spirit to [162], which also seeks to explain how simple analytics reveal aspects of the underlying system, but does not take a graph theoretic approach as is done here.

Notation

The notation here is consistent with previous sections. $A \in \mathbb{R}^{n \times m}$ is an n-by-m real-valued matrix. $\mathbf{A} \in \mathbb{C}^{n \times m}$ is an n-by-m complex-valued matrix while $\mathbf{A}^*$ and $\mathbf{A}^T$ are the conjugate and transpose of $\mathbf{A}$ respectively. Therefore, $\mathbf{A}^H = (\mathbf{A}^*)^T$. $\mathbf{A}_{ij}$ is the (i, j) element of $\mathbf{A}$. We often deal with the rows and columns of matrices, so they are explicitly denoted: $\mathbf{A}_{-i}$ is the i^{th} column and $\mathbf{A}_{i-}$ is the i^{th} row of matrix $\mathbf{A}$. The electric network is represented by a graph denoted $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with node set $\mathcal{N}$ and edge set $\mathcal{E}$. $|\mathcal{N}| = n$ and $|\mathcal{E}| = e$ is the number of nodes and edges in the graph respectively.

Measurement Model

For an electrical network represented by a graph $\mathcal{G}$ with n nodes, time series of nodal voltage and current phasors over t time points can be collected into matrices $\mathbf{V} \in \mathbb{C}^{n\times t}$ and $\mathbf{I} \in \mathbb{C}^{n\times t}$ respectively. The voltages and currents are related by the now-familiar matrix form of Ohm's Law

$$\mathbf{I} = \mathbf{Y}\mathbf{V} \leftrightarrow \mathbf{V} = \mathbf{Z}\mathbf{I} \tag{40}$$

Recall that $\mathbf{Y}$ is the network admittance matrix, or the Laplacian, of $\mathcal{G}$. It captures the complete network structure, consisting of connections and impedances. $\mathbf{Y}$ is symmetric and can be diagonalized as $\mathbf{Y} = \mathbf{U}D_\lambda\mathbf{U}^T$. The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{Y}$, while D_λ is a diagonal matrix of the eigenvalues of $\mathbf{Y}$, denoted $\lambda_1 \leqslant ... \leqslant \lambda_n$. By definition of the Laplacian, $\lambda_1 = 0$. The network impedance matrix $\mathbf{Z}$ is the *pseudoinverse* of $\mathbf{Y}$, defined as:

$$\mathbf{Z} \triangleq \mathbf{W}D_{1/\lambda}\mathbf{W}^T, \tag{41}$$

$$\mathbf{W} \triangleq \mathbf{U}^* \tag{42}$$

where $D_{1/\lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{Z}$, denoted $\gamma_1 \leqslant ... \leqslant \gamma_N$. The eigenvalue set of $\mathbf{Z}$ consists of the zero eigenvalues and the reciprocals of the *non-zero* eigenvalues of $\mathbf{Y}$. Therefore, $\mathbf{Z}$ preserves the null space of $\mathbf{Y}$. The rich, venerable field of graph theory has established many properties of graph Laplacian matrices and their eigenvalues and eigenvectors (see [163] for a summary), some of which will be instrumental here. In this work we use complex-valued, phasor measurements, but our results can be extended to magnitude measurements through power flow linearizations, such as the following simple one, which arises by inversion of decoupled power flow [164]:

$$V \approx XQ$$

where $V \in \mathbb{R}^{n \times t}$ contains nodal voltage magnitudes and $Q \in \mathbb{R}^{n \times t}$ contains nodal reactive power injections. $X \in \mathbb{R}^{n \times n}$ is the system's real-valued reactance matrix which, like $\mathbf{Z}$, is the pseudoinverse of a Laplacian encoding the same graph structure as $\mathbf{Y}$ but with real valued edge weights. By applying the assumptions made here on complex current injections $\mathbf{I}$ to Q instead, equivalent results can be derived for V as done here for phasor $\mathbf{V}$.

Assumptions

To derive various informative properties of voltage measurements, it is necessary to make enabling but reasonable assumptions. The analytic derivations in this section make one or both of the following assumptions:

A1 Current injections are uncorrelated and have equal variance across nodes. That is:

$$\bar{\mathbf{I}} \triangleq (\mathbf{I} - \frac{1}{t}\mathbf{I}\mathbf{1}\mathbf{1}^T) \implies \bar{\mathbf{I}}\bar{\mathbf{I}}^H = \sigma_{\mathbf{I}}^2 I_n \tag{43}$$

where I_n is the $n \times n$ identity matrix and $\sigma_{\mathbf{I}}^2$ is the current injection variance at every node. This assumption is similar to one made in Section 4.4, and is reasonable because currents are driven primarily by the power injections of disparate loads which should be statistically independent across nodes over short time scales.

A2 $D_{1/\lambda}$ is rank $k-1$ with $k-1$ approximately equal eigenvalues and $n-k+1$ zero eigenvalues. That is:

$$\gamma_{n-(k+1)} \approx \ldots \approx \gamma_n \triangleq \gamma \tag{44}$$

$$0 = \gamma_1 \approx \ldots \approx \gamma_{n-k} \tag{45}$$

The rationale for this assumption is based on graph theory and will be clarified in Section 4.5.1.

Which assumptions are at play in each derivation will be made clear in the text.

4.5.1 Voltage Data Rank

It is well known that grid voltage measurements are approximately low rank: that is, $\mathbf{V}$ can be closely approximated by a low rank matrix. This

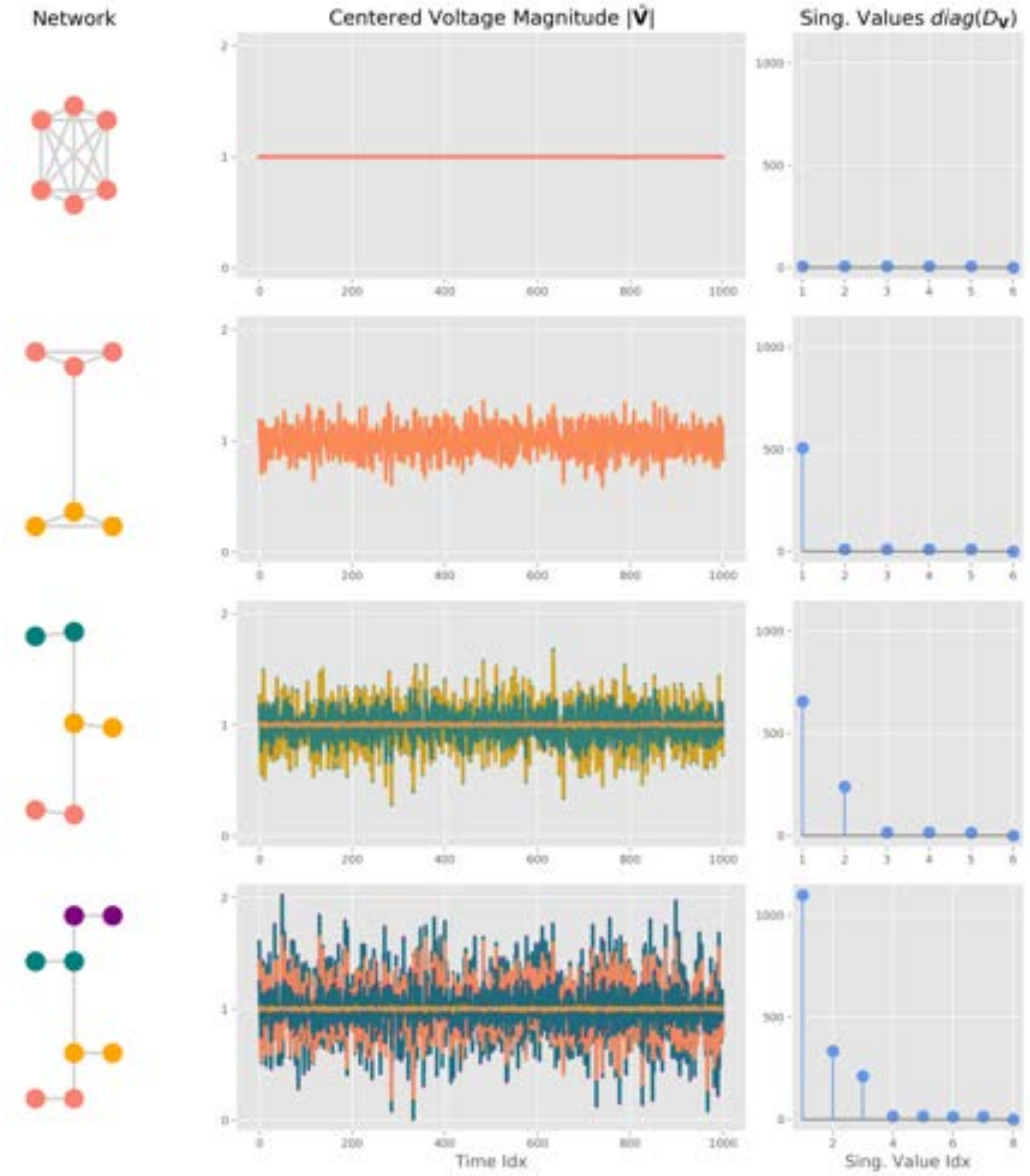


Figure 42: Voltage measurements and their singular values for four test networks with differing clusters. Shorter line length indicates lower admittance and higher impedance, or greater electrical proximity.

property motivates approaches to measurement compression, missing measurement recovery, and event detection [135]–[137]. It also motivates a new approach to system identification and event localization in [61] and is used to detect cyber attacks in [165]. However, despite its wide use, this low rank property is not theoretically established in prior work, and a discussion of the reasons and extent of this observed phenomenon is lacking. In fact, as we will show, the approximate rank of $\mathbf{V}$ is linked to the structure of the network graph $\mathcal{G}$. Define $\bar{\mathbf{V}} \triangleq \mathbf{V} - \frac{1}{t}\mathbf{V}\mathbf{1}\mathbf{1}^{T} = \mathbf{Z}\bar{\mathbf{I}}$ to be the mean centered voltages with singular value decomposition $\mathbf{A}D_{\bar{\mathbf{V}}}\mathbf{B}^{T}$ where $D_{\bar{\mathbf{V}}}$ is a diagonal matrix containing the singular values of $\bar{\mathbf{V}}$. Then, under assumption (A1):

$$\bar{\mathbf{V}}\bar{\mathbf{V}}^{H} = \mathbf{A}D^{2}_{\bar{\mathbf{V}}}\mathbf{A}^{H} = \sigma^{2}_{\mathbf{I}}\mathbf{Z}\mathbf{Z}^{H} = \sigma^{2}_{\mathbf{I}}\mathbf{W}D^{2}_{1/\lambda}\mathbf{W}^{T} \tag{46}$$

Therefore, the singular values of $\bar{\mathbf{V}}$ are the scaled eigenvalues of $\mathbf{Z}$, which in turn are the reciprocals of the eigenvalues of $\mathbf{Y}$. The number of zero eigenvalues of Laplacian $\mathbf{Y}$ equals the number of connected components—or independent nodes groups—in graph $\mathcal{G}$. Furthermore, a *near zero* eigenvalue of $\mathbf{Y}$ indicates a cluster or community of nodes that are highly connected to each other but “easily" separable (connected by low weight edges) from the rest of $\mathcal{G}$ [166]. Due to the inversion of (41), it is the near zero eigenvalues of $\mathbb{Y}$ that are the dominant eigenvalues of $\mathbf{Z}$. Therefore, if $\mathcal{G}$ contains k node clusters, $\bar{\mathbf{V}}$ will have $k-1$ dominant eigenvalues and can be well approximated by a rank k matrix.

This result is demonstrated in Fig. 42. The first column visualizes four toy networks with varied topologies and realistic line impedances. Current injections are generated for each node to statistically abide by the assumption of (A1). Next, nodal voltage phasors are computed via (40), plotted in the second column. The final column visualizes the eigenvalues of the voltage data. Notice the first network has a single, tight-knit community including all network nodes. Therefore the centered voltage magnitudes $\bar{\mathbf{V}}$ appear as a flat line since variation from the mean time series is minimal. Correspondingly, the voltage measurements have zero dominant singular values. In other words, most of the variation or information in $\mathbf{V}$ is captured in the average voltage time series. The other three toy networks have two, three, and four node communities respectively, reflected in their voltage singular values. Inter-community lines are higher impedance than intra-community lines, indicated by shorter line length in the network visualizations of Fig. 42.

This mathematical result on the singular values of voltage data has practical importance. It motivates using the structure of $\mathcal{G}$ to inform any applications relying on the rank of $\mathbf{V}$. Voltage data rank is useful in multiple applications, including compressing voltage data or recovering missing voltage measurements through low rank matrix recovery. Conversely, the rank of $\mathbf{V}$ indicates the structure of the network and can indicate and alert operators to any changes in $\mathcal{G}$. Notice in Fig. 42 that the voltage singular values are more compact and easily visualized than the raw voltage data. Now with theoretically justified physical meaning, they are an intuitive, visualizable quantity for system monitoring.

4.5.2 Voltage Clustering

Clustering voltage time series directly to infer the structure of $\mathcal{G}$ is an established heuristic technique for topology monitoring [167]. Yet, the theoretical underpinnings of voltage clustering are not established, and therefore what clustering reveals about the structure of $\mathcal{G}$ in general, and why, is poorly understood. This section provides theoretical justification of this heuristic technique through the lens of spectral clustering: a popular technique for clustering the nodes of a graph. In brief, the results of spectral clustering state that given Laplacian $\mathbf{Y} = \mathbf{U}D_\lambda\mathbf{U}^T$ of graph $\mathcal{G}$, the nodes of $\mathcal{G}$ can be separated into k clusters that maximize inter-cluster edge weights and minimize intra-cluster edge weights by applying k-means to the *rows* of matrix $\mathbf{U}^{(k)} \in \mathbb{C}^{n\times k}$. The columns of $\mathbf{U}^{(k)}$ are the first k eigenvectors of Laplacian $\mathbf{Y}$ (equivalently the first k columns of $\mathbf{U}$)—that is the eigenvectors corresponding to the k largest eigenvalues $\lambda_{n-k}, ..., \lambda_n$. K-means clustering is a standard algorithm which partitions data points into a specified number of clusters, where each data point is assigned to the cluster with the closest centroid or mean. Spectral clustering—derived through the relaxation of a non-convex optimization function over graph $\mathcal{G}$—can in practice be implemented with k-means (see [168] for a more in-depth study). Here we will prove that, under some assumptions, directly clustering the voltage time series $\bar{\mathbf{V}}_{1-}, ..., \bar{\mathbf{V}}_{n-}$ (collected as rows in the complete measurement matrix $\mathbf{V}$) produces the same result as spectral clustering applied to $\mathbf{Y}$. This result enables voltage clustering to be used for network structure monitoring.

Suppose $\mathcal{G}$ contains k node clusters. As discussed in Section 4.5.1, it is then reasonable to make assumption (A2), which implies:

$$\bar{\mathbf{V}} \approx \gamma(\mathbf{U}^{(k)})^*(\mathbf{U}^{(k)})^H\bar{\mathbf{I}} \tag{47}$$

The result of applying k-means clustering to the rows of $\mathbf{U}^{(k)}$—denoted $\mathbf{U}^{(k)}_{1-}, \ldots, \mathbf{U}^{(k)}_{n-}$—will depend on the pairwise distances between rows under the chosen clustering distance metric [169]. A common distance metric choice is the Euclidean distance which is preserved under conjugation and multiplication by an orthogonal matrix. Then, under (A1), the distance between voltage time series $\mathbf{V}_{i-}$ and $\mathbf{V}_{j-}$ is:

$$\|\bar{\mathbf{V}}_{i-} - \bar{\mathbf{V}}_{j-}\|_2 = \\ \|\mathbf{U}^{(k)}_{i-}(\mathbf{U}^{(k)})^H\bar{\mathbf{I}} - \mathbf{U}^{(k)}_{j-}(\mathbf{U}^{(k)})^H\bar{\mathbf{I}}\|_2 = \sigma_{\mathbf{I}}\gamma\|\mathbf{U}^{(k)}_{i-} - \mathbf{U}^{(k)}_{j-}\|_2 \tag{48}$$

Together, (47) and (48) show that the ordering of pairwise distances between rows are identical between $\bar{\mathbf{V}}$ and $\mathbf{U}^{(k)}$, which means that applying k-means clustering to the rows of $\bar{\mathbb{V}}$ (the voltage time series $\bar{\mathbf{V}}_{1-}, \ldots, \bar{\mathbf{V}}_{n-}$) is equivalent to applying spectral clustering to $\mathcal{G}$. This result is demonstrated in Fig. 43, which uses the same toy networks as Fig. 42. K-means clustering is applied to the centered nodal voltage time series $\bar{\mathbf{V}}$ of each network. The resulting node cluster membership is indicated by node color. We see that the results of voltage clustering match those of spectral clustering: correctly grouping nodes according to their underlying topological proximity in $\mathcal{G}$. This result explains why voltage clustering techniques for coherency identification, such as [170], succeed at the transmission level, and extends the technique to distribution, even in the absence of inertial generators, under the stated assumptions.

4.5.3 Voltage Correlations

Using voltage correlations for system monitoring or topology estimation is a common heuristic technique [95]. Indeed, it was integral to the topology heuristic described earlier in this Thesis (Section 4.2). Here, based on the physics model and assumptions (A1) and (A2), we propose and justify an extension: clustering the rows of the voltage correlation matrix, denoted $\mathbf{S}^{(\mathbf{V})}$. We can show that, under the given assumptions, this is equivalent to spectral clustering on $\mathcal{G}$. Using assumption (A1), and Eqs. (40)-(41), $\mathbf{S}^{(\mathbf{V})}$ can be expanded as:

$$\mathbf{S}^{(\mathbf{V})} \triangleq \bar{\mathbf{V}}\bar{\mathbf{V}}^H = \sigma^2_{\mathbf{I}}\mathbf{U}^*D^2_{1/\lambda}\mathbf{U}^T \tag{49}$$

Incorporating (A2) gives:

$$\mathbf{S}^{(\mathbf{V})} \approx \sigma^2_{\mathbf{I}}\gamma^2(\mathbf{U}^{(k)})^*(\mathbf{U}^{(k)})^T \tag{50}$$

Since $(\mathbf{U}^{(k)})^T(\mathbf{U}^{(k)})^* = I_k$—the identity matrix—by the results of Section 4.5.2, clustering the rows of $\mathbf{S}^{(\mathbf{V})}$ will produce the same result as applying spectral clustering to $\mathcal{G}$. This result is visualized in Fig. 44, which shows the node cluster membership found by clustering the rows of $\mathcal{S}^{(\mathbf{V})}$, alongside $\mathcal{S}^{(\mathbf{V})}$ for each toy network. We see the clusters over $\mathcal{S}^{(\mathbf{V})}$—captured in the node color and indicated by the numbers segregating the rows of $\mathcal{S}^{(\mathbf{V})}$—match the physical clusters in $\mathcal{G}$.

A cursory comparison of Sections 4.5.2 and 4.5.3 may lead to the conclusion that clustering the pairwise correlations in $\mathbf{S}^{(\mathbf{V})}$ is exactly equivalent to clustering the voltage time series in $\bar{\mathbf{V}}$. However, notice that assumption (A2) is applied to $D^2_{1/\lambda}$ to derive approximation (50) for correlation clustering, while for voltage clustering it is directly applied to $D_{1/\lambda}$ to derive approximation (47). (A2) will tend to be more accurate on $D^2_{1/\lambda}$ than $D_{1/\lambda}$ as squaring the singular value matrix will cause the largest k eigenvalues to further dominate over the smaller. Therefore, clustering over $\mathbf{S}^{(\mathbf{V})}$ will tend to be more robust than clustering over $\bar{\mathbf{V}}$.

Failure Cases

A critical strength of justified heuristics is that by understanding the assumptions underlying the analytic approaches, it is possible to define when they will fail. This is vital for use in safety-critical infrastructure such as the electric grid, where operators will take rapid and impactful decisions based on the results of analytics. For the methods described here, we understand that they will fail when assumptions (A1) and/or (A2) do not hold. Fig. 45 shows how voltage time series clustering and voltage correlation clustering produce unexpected results—different from the expected equivalence to spectral clustering— when assumption (A1) is significantly violated. To create this failure case, current injections $\mathbf{I}$ were generated to have a particular correlation structure, visualized in Fig. 45. This is a pathological example, but current correlations can indeed stray far from (A1). For example, the presence of distributed generation (such as PV) at multiple nodes can produce high correlations in current injections. Electric demand at different households has been observed to be correlated through a range of factors [171]. This will manifest in correlations among $\mathbf{I}$ injections.

4.5.4 Voltage PCA

Principal component analysis (PCA) transforms data to a new, lower dimensional subspace while maximizing the preserved variance (see [172] for a more detailed study). Applying PCA to voltage measurements is a widespread heuristic used for grid monitoring. In [93], PCA is used to project a multitude of PMU voltage measurements onto a lower dimension subspace to reduce the data quantity and detect system changes. Linear systems theory is used to justify this approach, but no connection is made with the underlying graph structure. In [94], PCA is also used for event detection and localization, but again without theoretical justification. In [140], PCA is applied to voltage angle measurements for fault detection. The advantages of PCA for reducing unwieldy data sets into tractable, informative and visualizable ones has motivated its use in other grid measurement applications, such as bad data detection [138], [139]. However, to the best of our knowledge, the theory establishing the efficacy and physical meaning of PCA on voltage measurements is absent from the literature. Here, we connect PCA to spectral clustering. PCA enables the reduction of $\bar{\mathbf{V}} \in \mathbb{C}^{n \times t}$ to the lower dimensional $\hat{\mathbf{V}} \in \mathbb{C}^{n \times k}$. By definition, the principal components, contained in the matrix denoted $\mathbf{P}$, are the conjugate eigenvectors of $\bar{\mathbf{V}}^H \bar{\mathbf{V}}$, which can be expanded as follows:

$$\bar{\mathbf{V}}^H \bar{\mathbf{V}} = \bar{\mathbf{I}}^H \mathbf{U} D_{1/\lambda}^2 \mathbf{U}^H \bar{\mathbf{I}} \tag{51}$$

Under (A1), $\bar{\mathbf{I}}^H \mathbf{U}$ has orthogonal columns. Therefore:

$$\mathbf{P} = \mathbf{U}^H \mathbf{I} \tag{52}$$

where the rows of $\mathbf{P}$ are the principal components. To find $\hat{\mathbf{V}}$, $\bar{\mathbf{V}}^H$ is projected on to the first k principal components. Under (A2), this leads to:

$$\hat{\mathbf{V}} = \bar{\mathbf{V}}^* (\mathbf{P}^{(k)})^T = \sigma_{\mathbf{I}}^2 \gamma \mathbf{U}^{(k)} \tag{53}$$

Therefore, the PCA transformed data is the scaled top k eigenvectors of $\mathbf{Y}$. From (53), it is evident that clustering $\hat{\mathbf{V}}_{1-}, ..., \hat{\mathbf{V}}_{1-}$ is equivalent to clustering $\mathbf{U}_{1-}^{(k)}, ..., \mathbf{U}_{n-}^{(k)}$, which is spectral clustering. Therefore, PCA projects the measurements into a subspace with measurement clusters corresponding to node clusters in the graph $\mathcal{G}$.

This result is illustrated in Fig. 46. For each toy network, we choose $k = 2$, which allows for the visualization of the PCA projection in two dimensions. Notice that from a spectral clustering perspective, the choice of $k = 2$ means we are only bisecting $\mathcal{G}$. However, when plotting $\hat{\mathbf{V}} \in \mathbb{C}^{n \times 2}$, we see that the top two eigenvectors of $\mathbf{Y}$—contained in $\mathbf{U}^{(2)}$—are actually effective at separating the nodes into more defined, smaller clusters. In fact, the plots show that the nodes are well separated to reflect their true topological clusters in $\mathcal{G}$. In other words, $\mathbf{U}^{(2)}$ contains more information on the structure of $\mathcal{G}$ than a crude graph bisection.

* * *

This section described a suite of justified heuristics for topology monitoring that are theoretically justified through simple physics models and the application of realistic assumptions. Though they may not produce the level of detailed system information generated by other tools in this section, they represent the most usable of the tools described, best meeting the three requirements of using physically meaningful quantities, having low and flexible input information requirements, and considering the limits of real measurements, as detailed in Chapter 3. As this chapter captures, they are the culmination of a long journey spanning a range of tool forms. They also suggest a path forward toward the creation of additional usable tools, to be discussed further in the final chapter.

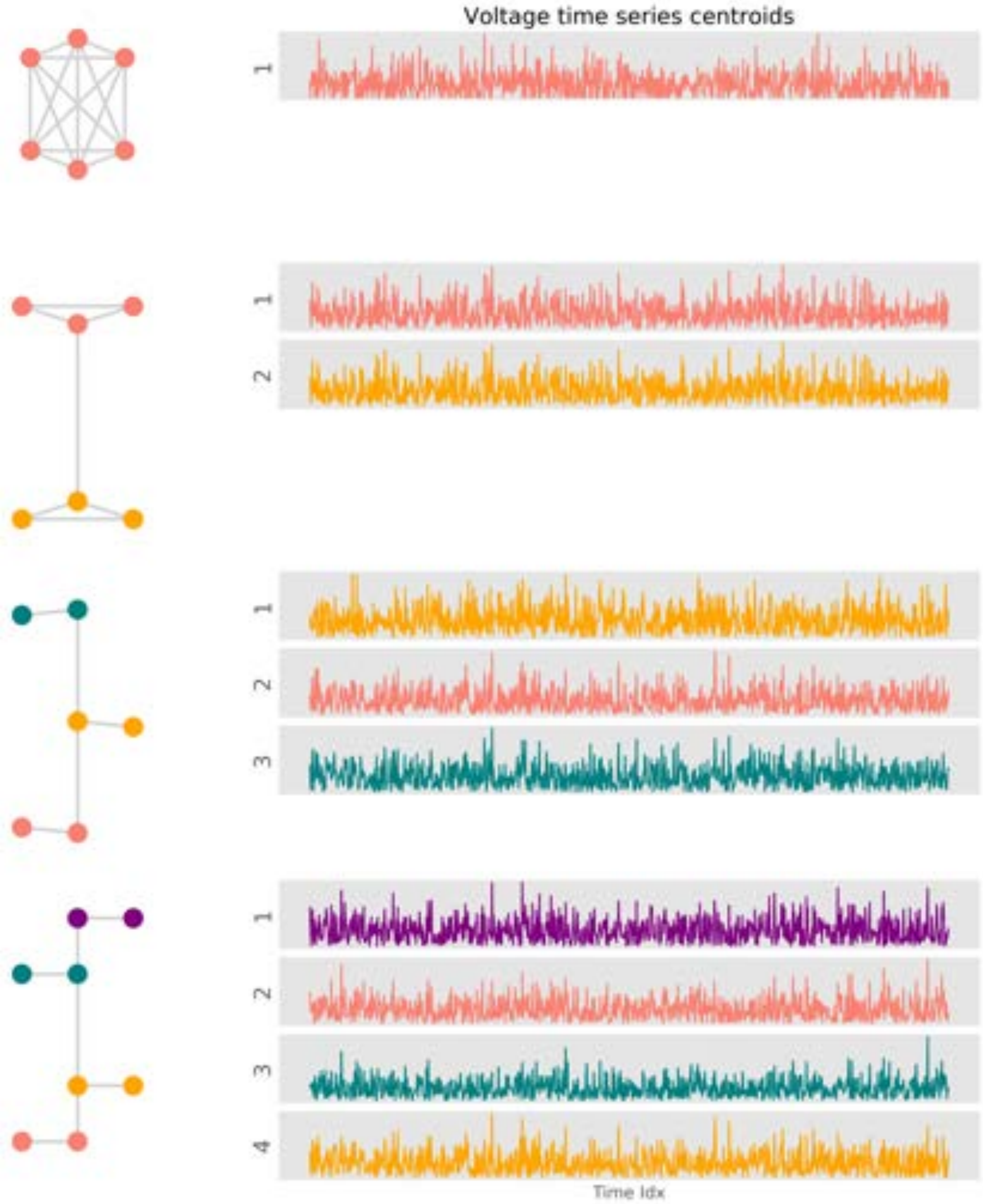


Figure 43: Visualizing node clustering through k-means clustering of nodal mean-centered voltage time series. Plots at right show cluster centroids which are the average time series across member nodes. Node color in the graphs at left indicates cluster membership and matches the corresponding centroid time series.

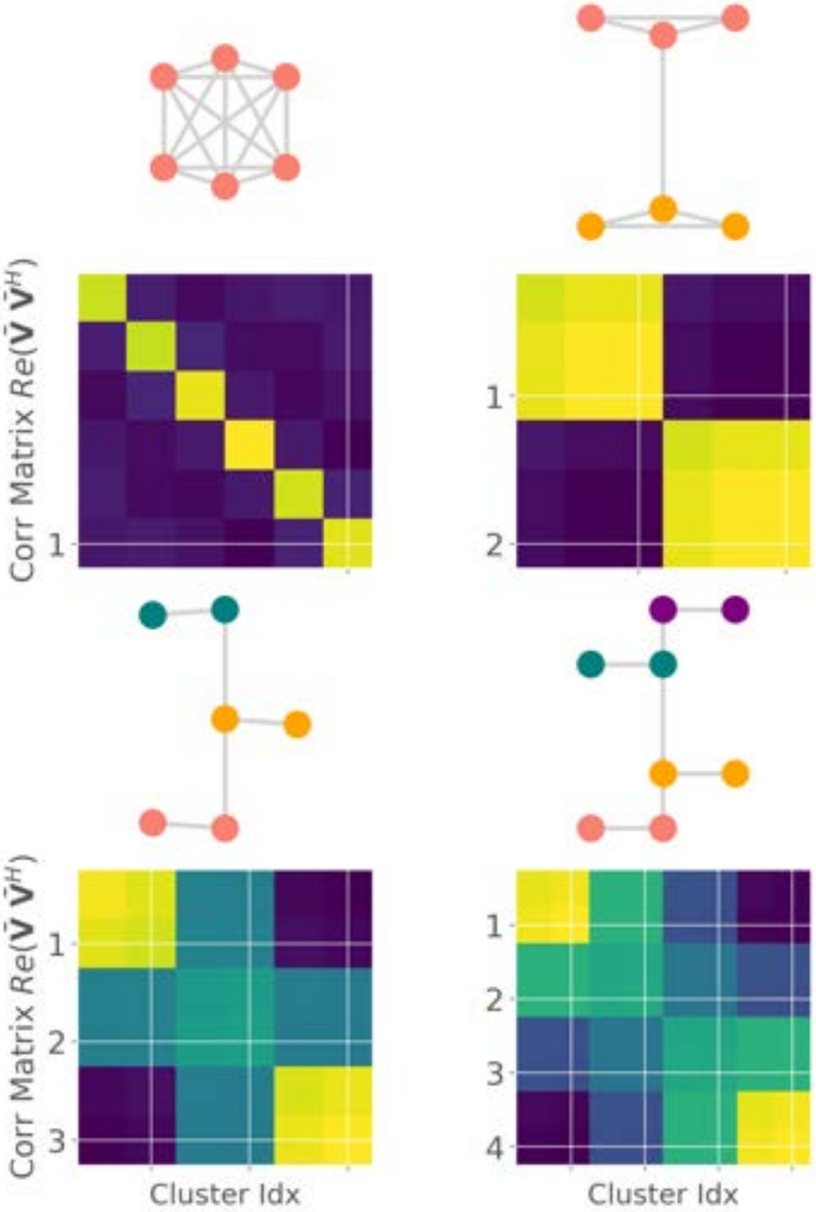


Figure 44: Results of node clustering by clustering voltage correlation matrix $\mathbb{S}^{(\mathbb{V})} = \bar{\mathbb{V}}\bar{\mathbb{V}}^{\mathrm{H}}$ with k-means. Node color indicates cluster membership. Plots show $\mathrm{Re}(\mathbb{S}^{(\mathbb{V})})$ for each network.

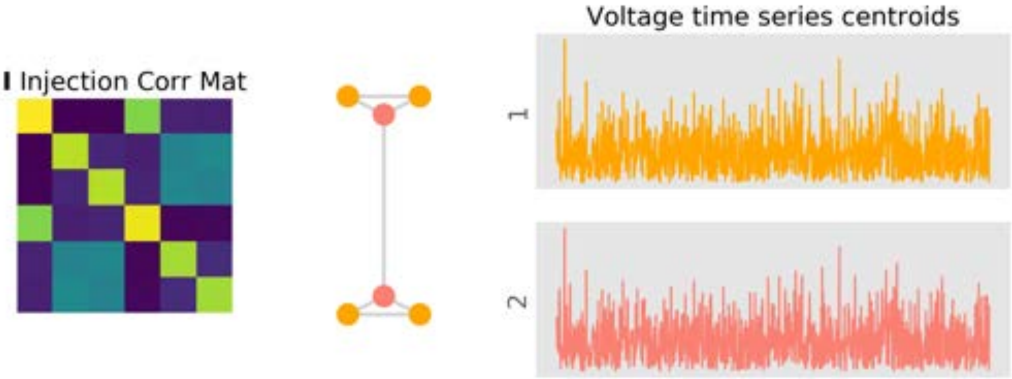


Figure 45: Results of node clustering by clustering nodal voltage time series with k-means. Node color indicates cluster membership. Plots at right show centroids of mean centered voltages for each cluster. In this case, the current injections are correlated across nodes with the correlation matrix shown at left, leading to failure of the voltage clustering.

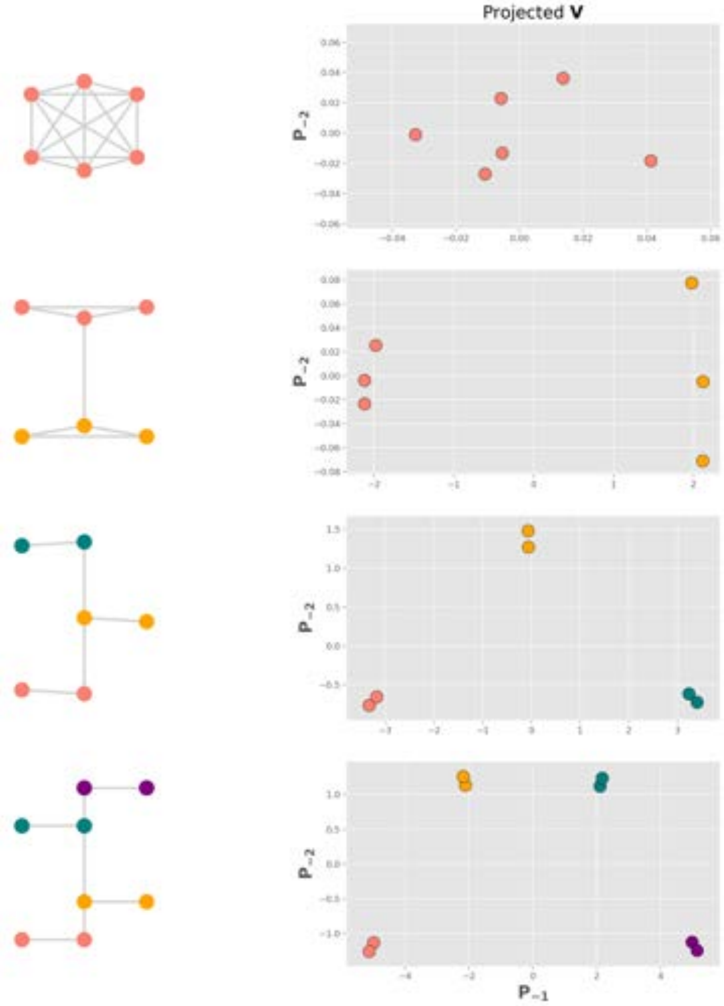


Figure 46: PCA with $k = 2$ applied to nodal measurements in $\bar{\mathbf{V}}$. Plots at right show the coefficients obtained by projecting nodal voltages on the first two principal components. Notice how the projected data is well separated into the underlying node clusters in $\mathcal{G}$.

5 EVENT DETECTION & CLASSIFICATION TOOLS

Includes work from [173], [174]

Incessantly and continually within the sprawling infrastructure of the electric grid, *things* occur. These multifarious things have varied causes, effects, and importance. Some, from the perspective of the grid operator, are exogenous but expected: a typical example is a reasonable change in electricity demand. Others arise from premeditated actions and are entirely foreseeable and generally desirable. Examples include the programmed response of capacitor banks, transformers, and other control devices. More unsettling are those things that are surprising and unwanted: the failure of crucial equipment, faults instigated by trees or animals touching power lines, and even the sabotage of grid devices by hackers, as infamously happened in Ukraine in 2015 [14]. These things are generically called *events*; a term that is appropriately imprecise given their variety[1].

By definition, knowing when and what events occur in an electric network (respectively the problem of event detection and event classification) is an integral part of situational awareness. Amidst the complexity of a typical electric grid, awareness of every event is practically impossible. Here, the chosen method of event detection is decisive in determining which events we have awareness of. This can lead to epistemic difficulty: ignorant of the unseen, how can we know if we have achieved situational awareness? Therefore, event detection tools must be created and continually refined through feedback between algorithm, user, and environment, until no event with *consequences* goes unnoticed. When this is the case, at least we can claim to have achieved adequate—if not total—situational awareness.

---

[1] While the term event is often earnestly debated in the literature and at conferences, really anything that happens in the system can legitimately be called an event. With such a capacious definition, more important than a universal consensus is clear specifications in each application of what constitutes an event.

This chapter presents work on tools for event detection and classification, with a focus on the latter. In the spirit of usability, these algorithmic tools run on single measurement streams and are envisioned as assisting a human user in identifying system events. Unlike the tools of the previous chapter, they are not based on physics models, but ensure transparency through intuitive approaches that can be applied to raw measurements (which retain direct physical interpretability) or physically meaningful computed quantities. The examples in this chapter apply these methods to raw measurements only.

Prior Work

Event detection is the first step in identifying important periods from long, mostly inconsequential data streams. All event detection algorithms identify changes in raw measurements or computed quantities. Numerous methods exist for event detection and classification. This section—by no means a comprehensive review—gives a flavor of the diverse approaches and aims.

The electric grid is never perfectly static, and real measurements inevitably contain some noise. Together, this means that measurements and computed quantities will always vary in time. Event detection methods must use thresholds to distinguish this normal variation from changes that indicate system events. Broadly then, the creation of any event detector entails two choices. First, the quantity in which events are to be detected must be selected. Should we look for events in voltage magnitudes, frequency, real power? Or perhaps events should be sought in an estimated impedance quantity or even a stream of voltage predictions? Events can be detected in any of these raw or computed streams. While some types of events may manifest across multiple streams, others may be distinguishable only in specific ones. Therefore, the choice of data stream is critical in determining which events will be discovered. This choice also has implications for transparency. A frequency event may manifest in voltage magnitude data, but detecting it in frequency data carries greater explanatory power for the human user.

The second choice for event detection is how to determine the event threshold. Often, thresholds are set to constant values. A constant threshold can be an intuitive and simple choice, especially if it is based on a widely acknowledged standard. For example, the American National Standards

Institute requires system voltage to remain within ten percent of nominal [175]. Setting 0.9 and 1.1 p.u. as constant thresholds for voltage magnitude events is therefore a tenable and intuitive choice. On the other hand, some choices of constant thresholds are opaque and even arbitrary, the single number obscuring the many iterations of tuning necessary to determine the perfect threshold for the tool designer's specific aims. Sometimes, when labeled events are available, this tuning is automated through a machine learning algorithm which sets the final threshold. More complex, evolving thresholds can also be chosen, using regression or statistical methods.

These two choices allow us to succinctly describe and compare the varied approaches to event detection presented in the literature. Both [94] and [93] apply dimensionality reduction to PMU voltage data, detecting events by applying constant thresholds to changes in the resulting eigenvalues or eigenvectors. [82] too detects events in PMU data of reduced dimensions, but rather than by a constant threshold, events and non-events are differentiated by a machine learning algorithm, which learns some concealed threshold. [176] also uses a machine learning method to determine a multidimensional threshold separating events, applying this to several streams computed from raw PMU data. [177], statistical, regression, and constant thresholds are used to detect events in voltage, current, and power streams. [92] takes a statistical tack, detecting events using multiple data streams simultaneously. A generative adversarial network learns the joint distribution of these multiple streams, flagging outliers as events. Similarly, [178] learns the distribution of PMU measurements—with fixed form but evolving parameters—and events are detected by thresholding changes in the estimated distribution parameters. [73] decomposes voltage and frequency measurements with the wavelet transform, using constant thresholds to flag excursions in the resulting wavelet coefficients as events.

After events are detected, they can be classified by type. Examples of event types include line trips, capacitor bank switching, topology changes, lightning strikes, animal contact, etc. Here again, numerous approaches are proposed in the literature, with the crucial distinction being between supervised approaches—that require (generally a large volume of) labeled examples—and unsupervised approaches—which do not require labeled data. Acquiring sufficient volumes of labeled data is arduous in the electric grid context, which leads to a great asymmetry in the practicality of supervised and unsupervised event detection approaches. Supervised ap-

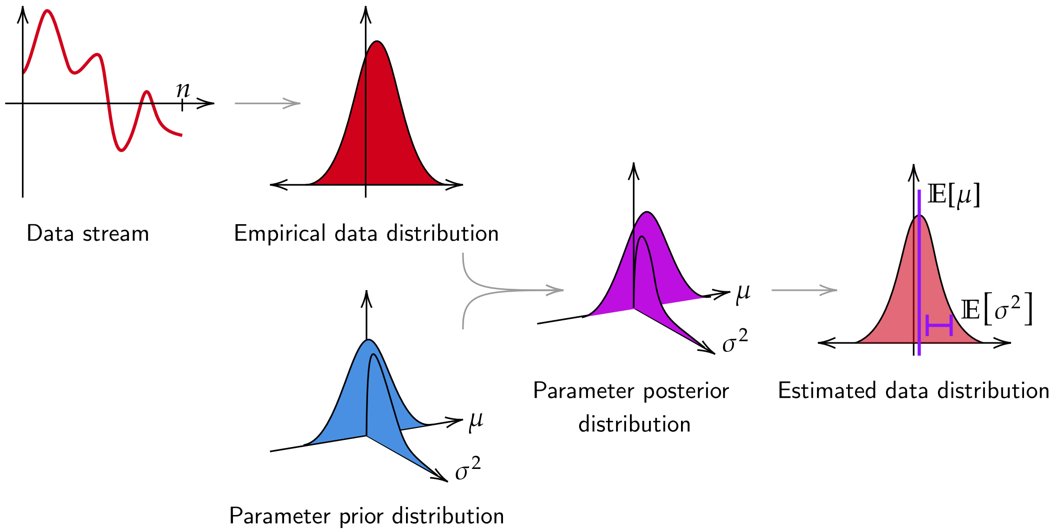


Figure 47: Visualization of bay-detect's maximum likelihood approach to data distribution estimation. Maximum likelihood estimation balances the empirical data distribution with the prior parameter distribution to find the most likely parameters.

proaches also suffer from limited transparency and indefinite or limited generalizability, especially when they use black-box machine learning models. While unsupervised approaches are less likely to manifest these issues, they are by no means immune. Some entail convoluted transformations of grid data that make their results frustratingly opaque.

[90], [179]–[183] train a variety of machine learning models to classify a variety of events from a variety of features derived from grid measurements. Support vector machines—which determine classification boundaries in a high-dimensional feature space—are a favored choice of model. Unsupervised approaches are also diverse. Some, such as [177] use physics based rules to distinguish event types. A popular class of approaches uses clustering algorithms; classifying events by grouping them according to their similarity in some feature space, as in [178], [184]. This approach will be discussed further in Section 5.2.

With the prior work mapped out, the next sections venture further afield, describing new approaches to event detection and classification.

5.1 BAYESIAN EVENT DETECTION

A statistical event detection framework is compelling for its transparency, modularity, and easy refinement. It consists of applying a statistical baselining method to a data stream and flagging outliers in the stream as events, conforming to the toolbox vision described in Section 3.5.

This section describes a Bayesian approach to statistical event detection, termed *bay-detect* [173]. Bay-detect assumes the input data stream samples are normally distributed with unknown mean μ and variance σ^2. While μ and σ^2 are assumed constant during the steady state, they may change following an event. Bay-detect attempts to estimate μ and σ^2 from the observed data in a maximum likelihood estimation (mle) framework, refining the estimates as more samples are observed. These parameters are assumed to arise from a prior distribution, chosen to be a normal-inverse-chi-squared ($NI\chi^2$) distribution for tractability. The mle approach balances the weight of the prior and that of the data, lending some stability to the parameter updates and allowing evolution of both the value of and confidence in the parameters. Fig. 47 visualizes this process, the mathematics of which are based on (and follow the notation of) the work in [185].

The $NI\chi^2$ distribution is a joint prior over the mean and variance, where the mean is normally distributed, while the variance follows an inverse-chi-squared distribution. The inverse-chi-squared distribution—which is positive valued and right-skewed—is a natural choice for the variance prior. The $NI\chi^2$ prior is written in Eq. 54 [185].

$$p(\mu, \sigma^2) = NI\chi^2(\mu, \sigma^2|\mu_i, \kappa_i, \sigma_i^2, \nu_i) = \mathcal{N}(\mu|\mu_i, \frac{\sigma^2}{\nu_i}) \cdot \chi^{-2}(\sigma^2|\sigma_i^2, \nu_o) \quad (54)$$

Variables μ_i, σ_i^2, κ_i and ν_i are the parameters of the prior. Initially ($i = 0$), they must be selected by the user, after which they evolve with time as data samples are observed. The parameters μ_i and σ_i specify the centering of the mean and variance priors respectively, whose product gives the joint prior. Parameters κ_i and ν_i capture the scale, or spread, of the mean and variance priors.

The critical step in the maximum likelihood approach of bay-detect is to determine the *posterior* parameter distribution by merging the guidance of the prior with that of empirical data samples. Conveniently, the posterior is also a $NI\chi^2$ distribution, with parameters $\mu_n, \kappa_n, \sigma_n^2, \nu_n$:

$$p(\mu, \sigma^2|x_1, ..., x_n) = NI\chi^2(\mu_n, \kappa_n, \sigma_n^2, \nu_n) \quad (55)$$

Given n data samples $x_1, ..., x_n$, with sample mean $\bar{x}$, the posterior $NI\chi^2$ parameters are defined by Eqs. 56-59.

$$\mu_n = \frac{\kappa_0 \mu_0 + n\bar{x}}{\kappa_n} \tag{56}$$

$$\kappa_n = \kappa_0 + n \tag{57}$$

$$\sigma_n^2 = \frac{1}{\nu_n}(\nu_0 \sigma_0^2 + \sum_{i=1}^{n}(x_i - \bar{x}^2) + \frac{n\kappa_0}{\kappa_n}(\mu_0 - \bar{x})^2) \tag{58}$$

$$\nu_n = \nu_0 + n \tag{59}$$

The posterior scale parameters κ_n and ν_n can be interpreted to indicate the confidence in the data distribution parameter estimates. They are equal to the number of samples observed, and as they increase, they tighten the spread of the posterior, reflecting the increasing certainty in the data distribution. Bay-detect utilizes the expectation over the posterior to produce the final data distribution parameter estimates. These final estimates are:

$$\mathbb{E}[\mu | x_1, ..., x_n] = \mu_n \tag{60}$$

$$\mathbb{E}[\sigma^2 | x_1, ..., x_n] = \frac{\nu_n}{\nu_n - 2}\sigma_n^2 \tag{61}$$

The method is easily made an *online* approach (able to handle streaming input data) by converting equations 56-59 into explicit update equations. Assume bay-detect has seen n data samples, $x_1, ..., x_n$, thus far—perhaps split over several streaming batches—from which the algorithm has obtained posterior parameters $\mu_n, \kappa_n, \sigma_n^2, \nu_n$. A new batch of k data points, $x_{n+1}, ..., x_{n+k}$ arrives. The posterior parameters can be updated based on the k new samples according to equations 62-65, where $\bar{x}$ is the sample mean of the k latest samples.

$$\mu_{n+k} = \frac{\kappa_n \mu_n + k\bar{x}}{\kappa_n} \tag{62}$$

$$\kappa_{n+k} = \kappa_n + k \tag{63}$$

$$\sigma_n^2 = \frac{1}{\nu_{n+k}}(\nu_n \sigma_n^2 + \sum_{i=1}^{k}(x_i - \bar{x})^2 + \frac{k\kappa_n}{\kappa_{n+k}}(\mu_n - \bar{x})^2) \tag{64}$$

$$\nu_{n+k} = \nu_n + k \tag{65}$$

In this way, as batches of data stream in, the posterior parameters are refined, resulting in new estimates of the data distribution parameters.

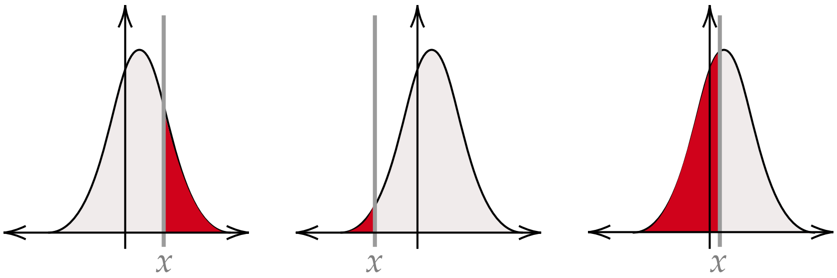


Figure 48: Visualizing the *tail probability* used by bay-detect to flag data point x as an event.

The first posterior parameter update incorporates the initial posterior parameter values: $\mu_0, \kappa_0, \sigma_0^2, \nu_0$. These must be specified by the user to reflect prior knowledge on the data distribution, and can indicate anything from highly accurate knowledge to essentially no knowledge. For example, setting κ_0 and ν_0 very close to 0 implies that there is no certainty on the data distribution.

Beginning with the data samples and proceeding through the sequence of empirical, prior, and posterior distributions, bay-detect ultimately determines the posterior distribution of *unseen* data given the observed samples. This predictive posterior is a generalized t-distribution denoted:

$$x \sim t_\nu(x|\mu, \sigma^2) \tag{66}$$

where μ is the mean, σ^2 is the scale, and ν is the degrees of freedom. The distribution resembles a Gaussian, but with heavier tails.

Given data points $x_1, ..., x_n$, the estimated posterior distribution of future point x is:

$$p(x|x_1, ..., x_n) = t_{\nu_n}\left(\mu_n, \frac{(1+\kappa_n)\sigma_n^2}{\kappa_n}\right) \tag{67}$$

x may be one or more steps into the future; there is no explicit time awareness in this formulation.

Using the predictive posterior, events can be detected probabilistically, based on a user specified probability threshold demarcating normal data from abnormal, or event, data. It is at this point that the statistical event detection approach especially shines. While other approaches require the selection of raw value thresholds—a process which generally demands a great deal of fumbling experimentation and results in seemingly arbitrary and non-intuitive values—bay-detect requests a probability from the user. For example, the user can specify that any data value with a probability < 0.001 according to the estimated data distribution should be considered an event. Such probability thresholds can be intuitively specified and interpreted. In the case of bay-detect, a *tail probability* is computed for data point x. Denoted p_{tail}, this is the cumulative probability in the tail demarked by x, as visualized in Fig. 48 and defined as:

$$\begin{aligned} F(x) &\triangleq \int_{y} p(y \leqslant x | x_1, ..., x_n) \\ p_{tail}(x) &= \min(F(x), 1 - F(x)) \end{aligned} \tag{68}$$

Given threshold ρ, data point x is flagged as an event if:

$$p_{tail}(x) \leqslant \rho \tag{69}$$

Once an event is flagged, bay-detect assumes that the data distribution may have changed, and the posterior parameters are reset to the prior values of $\mu_0, \kappa_0, \sigma_0^2, \nu_0$.

Algorithm 4 presents pseudocode of the bay-detect algorithm.

Demonstration

Fig. 49-50 visualize the results of applying bay-detect to simulated and real PMU voltage magnitude measurements with events. The time of events detected by bay-detect are indicated with red dashed vertical lines. When true event times are known, they are indicated by green vertical lines. The dashed orange time series visualizes the mean of the posterior predictive distribution, while the dark gray band visualizes its spread, together indicating the data distribution that bay-detect learns. Notice the gray band widening dramatically following an event, reflecting the resumption of uncertainty on the data distribution parameters. As more data points are observed, the gray band tightens, indicating the increasing certainty in the estimated data distribution.

Fig. 49 applies bay-detect to a simulated voltage magnitude time series containing a total of 14 events. The time series was generated by simulating the IEEE 14 Bus transmission test system [186] under a sequence of different event types, including load changes, capacitor bank switching,

A 4: $[E] = \text{bay-detect}(X)$

```
Input  : $X$ is a $1 \times t$ data time series
Output : $E$ is a $1 \times t$ binary vector where 1 indicates an event point
// Initialize posterior parameters
$\mu_0,\ \kappa_0,\ \sigma_0^2,\ \nu_0 \leftarrow 1,\ 0.1,\ 10,\ 0.1$ // Constants
$\mu,\ \kappa,\ \sigma^2,\ \nu \leftarrow X(0),\ \kappa_0,\ \sigma_0,\ \nu_0$ // Initial values
for $n \leftarrow 1$ to $t$ do
    $x \leftarrow X(n)$
    $cdf \leftarrow postProb(x, \mu, \kappa, \sigma^2, \nu)$
    // Compute tail probabilities
    $ptail \leftarrow \min(cdf, 1 - cdf)$ // Eq.68
    if $ptail \leqslant \rho$ then
        $E(n) \leftarrow 1$ // Event!
        $\mu, \kappa, \sigma, \nu \leftarrow x, \kappa_0, \sigma_0^2, \nu_0$ // Reset parameters after event
    else
        // Update parameters with new data point
        $\mu, \kappa, \sigma^2, \nu \leftarrow \text{update}([x],\ \mu,\ \kappa,\ \sigma^2,\ \nu)$
    end
end
```

A 5: $[\mu',\ \kappa',\ \sigma'^2,\ \nu'] = \text{update}(X_k,\ u,\ \kappa,\ \sigma^2,\ \nu)$

```
Input  : $X_k$ is $1 \times k$ vector of data with which to update parameters
         $u,\ \kappa,\ \sigma^2,\ \nu$ scalar posterior parameters.
Output : $\mu',\ \kappa',\ \sigma'^2,\ \nu'$ are updated parameters
$\bar{x} \leftarrow \frac{1}{k} \sum_{i=1}^{k} X_k(i)$ // Sample mean of $X_k$
$\mu' \leftarrow \frac{\kappa\mu + k\bar{x}}{\kappa}$ // Eq.62
$\kappa' \leftarrow \kappa + k$ // Eq.63
$\nu' \leftarrow \nu + k$ // Eq.65
$\sigma'^2 \leftarrow \frac{1}{\nu'}\left(\nu\sigma^2 + \sum_{i=1}^{k}(X_k(i) - \bar{x})^2 + \frac{k\kappa}{\kappa'}(\mu - \bar{x})^2\right)$ // Eq.64
```

A 6: $[cdf] = \text{postProb}(x,\ \mu,\ \kappa,\ \sigma^2,\ \nu)$

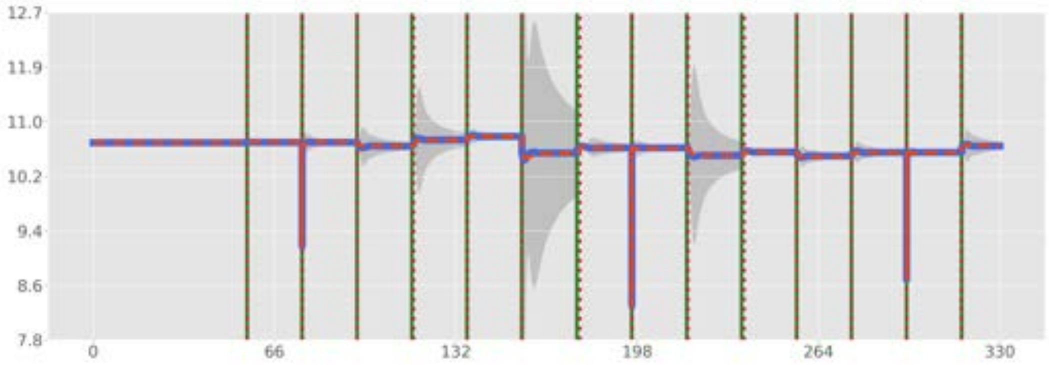
```
Input  : $x$ is scalar measurement to obtain posterior probability of given
         parameters $\mu,\ \kappa,\ \sigma^2,\ \nu$
Output : $cdf$ is cumulative probability of $x$
// Normalize data to use standard t-distribution
$\hat{\sigma}^2 \leftarrow \frac{1+\kappa}{\kappa}\sigma^2$
$\hat{x} \leftarrow \frac{x-\mu}{\sqrt{\hat{\sigma}^2}}$
$cdf \leftarrow t_\nu.cdf(\hat{x})$ // CDF of standard t-distribution
```

faults, and generator trips. One of the software PMUs "installed" at a bus in the network produced the synthetic voltage magnitude data plotted in Fig. 49. As the times of the simulated events were known, this data allowed for validation of bay-detect. Bay-detect successfully finds all the simulated events, detecting them at times closely aligned with the true event times. There are no false positives. Fig. 49b shows zoomed in results on a portion of the full time series.

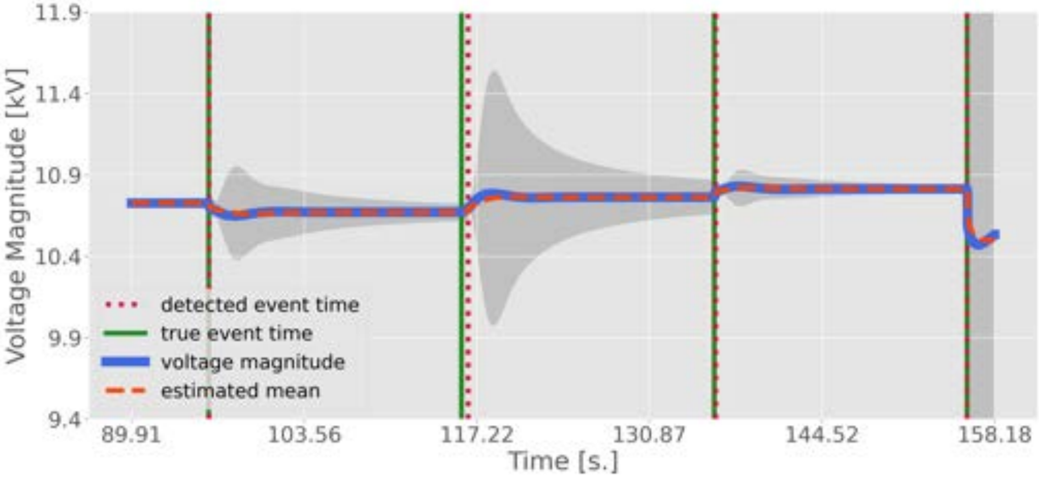
Fig. 50 applies bay-detect to a voltage magnitude time series from a μPMU installed on an operational distribution network. Event times for this data are not known, but some events—like the drastic voltage sag in Fig. 50a—are discernible by eye. Bay-detect successfully detects the voltage sag. In Fig. 50, bay-detect is applied to a period of real data containing no event. Here we see the learned posterior distribution evolving to match the data.

* * *

Once events are detected—with bay-detect or other techniques—they can be classified. The next section describes a clustering algorithm for event classification.

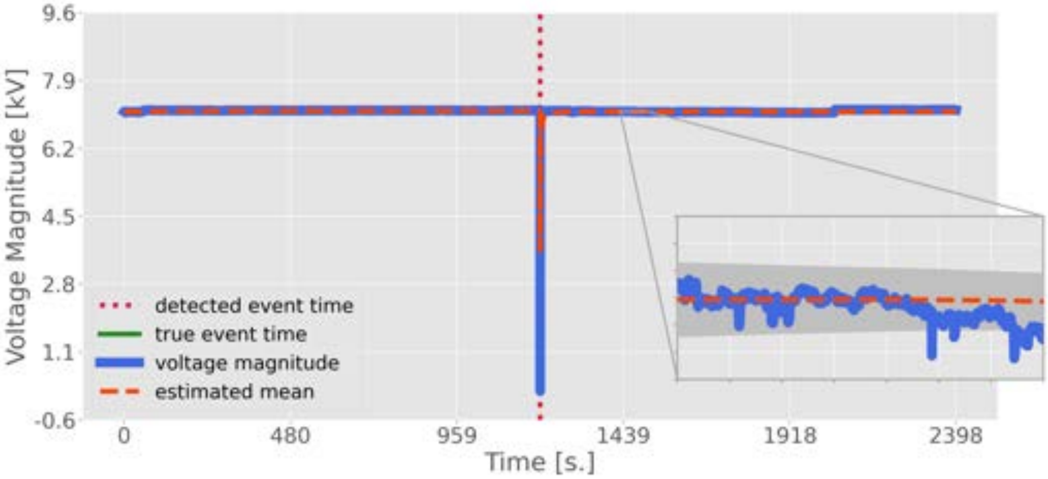


(a) Full time series with 14 events

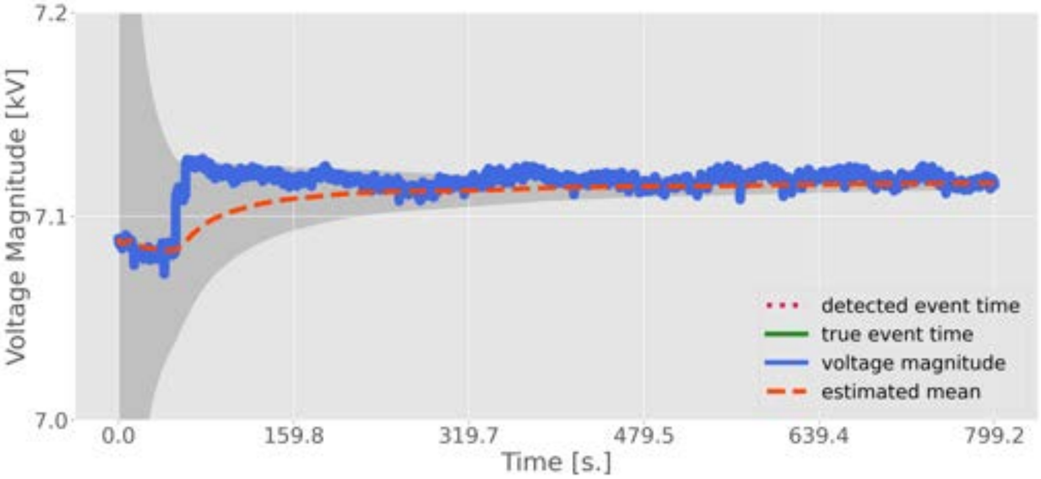


(b) Subseries with 4 out of 14 events

Figure 49: Bayesian event detection applied to a simulated voltage magnitude time series containing 14 events. Events were simulated on the IEEE 14 bus system and are of varied types. For realism, 30 dB of white noise was added to the simulated voltage data.



(a) A voltage sag event



(b) A non-event period

Figure 50: Bayesian event detection applied to voltage magnitude measurements from an operational distribution network.

5.2 STREAMING EVENT CLUSTERING

Clustering is a promising approach to event classification, as it addresses two key usability issues that mar other approaches. First, it is an *unsupervised* approach, meaning it does not require data with class labels. Second, clustering approaches can be highly transparent, especially if applied to raw measurements or simple features (conversely, clustering in abstruse feature spaces is highly opaque and provides little insight).

In the grid context, clustering has been extensively applied to load data [187]–[189], to find exemplars from large numbers of time series. Closest to the clustering application presented in this section are the approaches of [79], [190], [191], which apply off-the-shelf clustering algorithms to PMU time series measurements of events. This section presents k-*ShapeStream*, a novel clustering algorithm designed with time series grid measurements in mind. k-*ShapeStream* improves on the prior work in two critical respects. First, it is a *streaming* clustering approach, allowing clusters to be updated with new event data without requiring access to historical event data. This is a critical feature for algorithms intended to run online on continually updated measurement data streams. In contrast, the clustering approaches in prior works necessitate re-clustering the entire set of new and historical event data every time a new event is added, which quickly becomes impractical. Second, k-ShapeStream uses a probabilistic time-series distance measure for clustering which has multiple benefits. This distance measure improves algorithm performance, enables anomaly detection, and enhances human interpretation of the results. It associates a probability distribution with each cluster, which can be interpreted as a confidence metric. By providing such a confidence measure rather than cut-and-dried answers, k-ShapeStream gives users more context, further engendering trust in the analytic tool.

k-ShapeStream is the progeny of k-Shape, a recently developed—but already popular—time series clustering algorithm [192], [193]. To understand k-ShapeStream, it is helpful to first understand k-Shape, briefly described next in Section 5.2.1. Then, Section 5.2.2 details how k-ShapeStream extends k-Shape to (i) operate over streaming data; (ii) produce probabilistic interpretable results; and (iii) separate outlier events. Finally, Sections 5.2.3-5.2.4 demonstrate the effectiveness of k-ShapeStream on events detected in μPMU voltage magnitude measurements from an operational distribution network in California.

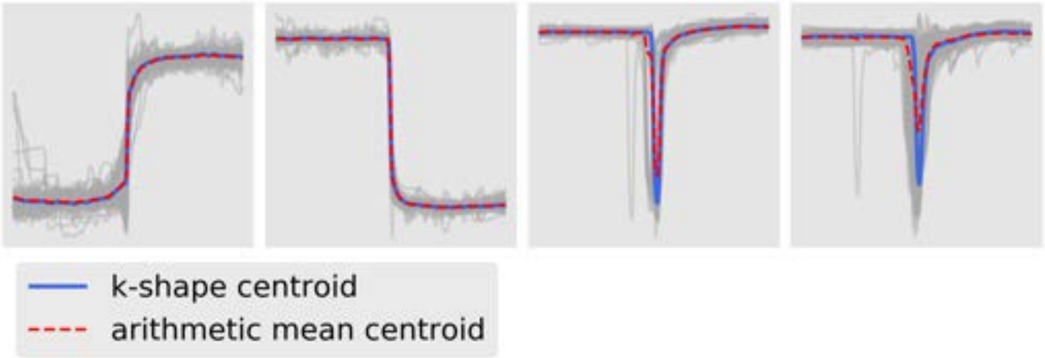


Figure 51: Visual comparison of cluster centroids computed using k-shape's eigendecomposition method versus the arithmetic mean. Gray lines show individual events in each cluster. The eigendecomposition better preserves the sharpest changes.

5.2.1 Background: k-Shape Time-Series Clustering

k-Shape is a time-series clustering algorithm that has been effectively applied to data from various domains [193], including the energy sector [194]–[196]. Similarly to the canonical k-means clustering algorithm [197], k-shape segregates input data points into k clusters—where k is specified by the user—by iteratively maximizing intra-cluster similarity. It ultimately returns cluster assignments for each input data point as well as a representative *centroid* for each cluster. A generic algorithm intended for use on data points in an arbitrary, multidimensional feature space, k-means uses Euclidean distance as the similarity measure between data points (lower distance means higher similarity) and computes the cluster centroids as the arithmetic mean of all member points. Here k-shape has a distinct difference, using an approach targeted to time series data. In the place of Euclidean distance, k-shape uses a normalized version of cross-correlation as the similarity measure between time series, termed *Shape-based Distance (SBD)*. In place of the arithmetic mean, k-shape applies an eigendecomposition-based method for centroid computation. Both modifications are especially suited to time series measurements. SBD is intuitive, robust to time series scaling and misalignment, and can be efficiently computed via the Fast Fourier Transform (FFT) [198]. While the arithmetic mean for centroid computation tends to have a low pass effect, eigendecomposition preserves sharp edges in time series signatures—as shown in

Fig. 51—and, therefore, produces more representative centroids. Together, these features make k-shape an attractive algorithm for clustering time series grid events.

5.2.2 Clustering Streams of Time Series

With all its compelling attributes, k-shape has one major weakness. To determine clusters, it requires access to the entire set of time series, making it prohibitively computationally expensive for use in streaming settings due to the need to re-cluster new and historical data. k-ShapeStream alleviates this shortcoming.

k-ShapeStream proceeds in *rounds*, handling one batch of data per round. In round r of clustering, the algorithm ingests n_r time series of length t, contained in the $n_r \times t$ data matrix X_r. Associated with each of the k clusters are nine parameters which evolve over rounds. For cluster j in round r, they are denoted and defined as follows:

- $m_r(j)$ is the scalar count of cumulative cluster members.
- $S_r(j)$ is the $t \times t$ cluster shape matrix.
- $u_r(j)$ is the $t \times 1$ time series cluster centroid.
- $\mu_r(j)$ is the scalar mean of the cluster distance distribution.
- $\delta_r(j)$ is the scalar squared mean of the cluster distance distribution.
- $\sigma_r(j)$ is the scalar standard deviation of the cluster distance distribution.

To enable a probabilistic clustering distance metric, the intra-cluster distances —between the time series members of a cluster and its centroid —are modeled as normally distributed, with mean and variance $\mu_r(j), \sigma_r(j)$. The scalar $\delta_r(j)$ enables a streaming update of the normal distribution parameters. All the listed parameters are efficiently updated in each round and are the *only* data carried forward between rounds. A length n_r list of indices indicating the cluster assignment of time series in X_r—denoted IDX_r—is returned after each round. Fundamental to the streaming approach is that shape matrices—from which cluster centroids are extracted via eigendecomposition—can be linearly updated with each new round of data, allowing centroids to reflect the entire, cumulative set of cluster members without accessing time series from prior rounds.

k-ShapeStream assigns each time series in data matrix X_r either to one of the k clusters, or to an *outlier* set, based on the normalized cross-correlation distance, or SBD, between the time series and the centroid of each cluster. The assignment depends on the probability of the distance between the time series and current cluster centroids, given the current intra-cluster distance distributions. As mentioned, these distributions are assumed normal, with mean $\mu_{r-1}(j)$ and standard deviation $\sigma_{r-1}(j)$ for cluster j. Therefore, in round r, time series i is assigned to a cluster or to the outlier set as follows:

$$\mathrm{dist}_i(j) \triangleq \frac{\mathrm{SBD}(X_r(i), u_r(j)) - \mu_{r-1}(j)}{\sigma_{r-1}(j)} \tag{70}$$

$$\mathrm{IDX}_r(i) = \begin{cases} \arg\min_j \mathrm{dist}_i(j) & \text{if } \min_j \mathrm{dist}_i(j) < \tau \\ \mathrm{outlier} & \text{otherwise} \end{cases} \tag{71}$$

τ is a user specified threshold of the number of σ's of permissible deviation for a time series to be assigned to a cluster. A typical choice—used in this work—is $\tau = 2$. By labeling outliers, k-ShapeStream allows unusual or unfamiliar events to be flagged for analysis and also avoids cluster contamination by outliers. Within round r, cluster memberships are iteratively refined, either until they have stabilized or until the maximum number of iterations has been reached.

Assume clustering has proceeded to the end of round r, having assigned each time series in X_r to one of the k clusters or the outlier set. $X_r(\mathrm{IDX}_r = j)$ is the set of time series from round r assigned to cluster j, which will be denoted X_r^j for brevity. At the end of the round, cluster parameters are updated based on the final time series assignments. The parameters of cluster j are updated according to the following equations:

$$m_r(j) = m_{r-1}(j) + |IDX_r = j|_1 \tag{72}$$

$$S_r(j) = X_r^j \cdot (X_r^j)^T + S_{r-1}(j) \tag{73}$$

$$u_r(j) = eig(Q^T \cdot S_r(j) \cdot Q) \text{ where } Q \triangleq I - \frac{1}{t}O \tag{74}$$

$$\mu_r(j) = \frac{m_{r-1}(j) \cdot \mu_{r-1}(j) + \sum_l SBD(u_r(j), X_r^j(l))}{m_{r-1}(j) + |IDX_r = j|_1} \tag{75}$$

$$\delta_r(j) = \frac{m_{r-1}(j) \cdot \delta_{r-1}(j) + \sum_l SBD(u_r(j), X_r^j(l))^2}{m_{r-1}(j) + |IDX_r = j|_1} \tag{76}$$

$$\sigma_r(j) = \alpha\sqrt{\delta_r(j) - \mu_r(j)^2} + (1 - \alpha) \tag{77}$$

The equations make clear why we track the mean of squared intra-cluster distances: it is needed in the standard deviation update of (77). Motivated by a maximum likelihood approach, a smoothing factor, denoted α, is used in the standard deviation update to capture increasing certainty in the distribution parameters with increasing number of cluster members [185]. The value of α is:

$$\alpha = \frac{m_{r-1}(j) + |IDX_r = j|_1}{1 + m_{r-1}(j) + |IDX_r = j|_1} \tag{78}$$

Importantly, the cluster parameters are updated using only the previous parameters but none of the member time series from prior rounds. Therefore, once all cluster parameters have been updated at the end of round r, X_r can be completely discarded; all pertinent information for the next round is captured in the cluster parameters. This is the fundamental benefit of the streaming approach which makes the method sustainable for application to ever-growing streams of indefinite duration.

Pseudocode for k-ShapeStream is provided in A7-9. The SBD function called in the pseudocode returns the shape based distance and aligned time series. SBD is fully described in [192] and has achieved state-of-the-art accuracy and runtime performance [199]. The input time series must initially be z-normalized, as described in [200]. Pseudocode for the update of the intra-cluster distance distribution parameters are in A9.

```
A 7: $[IDX_r, C_r] = \text{k-ShapeStream}(X_r, C_{r-1})$
Input  : $X_r$ is an $n_r$-by-t matrix containing $n_r$ z-normalized time series of
         length t.$C_{r-1}$ contains cluster parameters from the prior round.
Output : $IDX_r$ is an $n_r$-by-1 vector containing the assignment of $n_r$ time
         series to k clusters or outlier set. $C_r$ contains cluster parameters
         at the end of this round.
$u_r \leftarrow C_{r-1}.u,\ m_{r-1} \leftarrow C_{r-1}.m$ // prior centroids, cluster member counts
$\mu_{r-1},\ \delta_{r-1},\ \sigma_{r-1} \leftarrow C_{r-1}.\mu,\ C_{r-1}.\delta,\ C_{r-1}.\sigma$ // prior dist params
$\mu_r,\ \delta_r,\ \sigma_r \leftarrow \mu_{r-1},\ \delta_{r-1},\ \sigma_{r-1}$ // init new dist params
$IDX'_r \leftarrow [], S_r \leftarrow []$ // initial assignment & shape matrices
$mindist \leftarrow \mathbf{0}$ // $n_r$-by-1 zeros vector, distance to nearest centroid
while $IDX'_r \neq IDX_r$ & $iter = 0 : 100$ do
    $IDX'_r \leftarrow IDX_r$
    for $j \leftarrow 1$ to $k$ do
        $X' \leftarrow []$
        for $i \leftarrow 1$ to $n_r$ do
            if $IDX_r(i) = j$ then
                $X' \leftarrow [X', X_r(i)]$
            end
        end
        $[u_r(j), S_r(j)] \leftarrow \text{ShapeExtraction}(X', u_{r-1}(j), S_{r-1}(j))$
    end
    for $i \leftarrow 1$ to $n_r$ do
        $mindist(i) \leftarrow \infty$
        for $j \leftarrow 1$ to $k$ do
            $[d, x'] \leftarrow SBD(u_r(j),\ X_r(i))$
            $dist \leftarrow \frac{|d - \mu_{r-1}(j)|}{\sigma_{r-1}(j)}$ // Eq.70
            if $dist < mindist(i)$ then
                $mindist(i),\ IDX_r(i) \leftarrow dist,\ j$ // Eq.71
            end
        end
        if $mindist(i) > \tau$ then
            $IDX_r(i) \leftarrow k + 1$ // Outlier, Eq.71
        end
    end
    $[m_r, \mu_r, \delta_r, \sigma_r] \leftarrow \text{Update}(X_r, IDX_r, mindist, m_{r-1}, \mu_{r-1}, \delta_{r-1}, \sigma_{r-1})$
end
$C_r.m, C_r.S, C_r.u, C_r.\mu, C_r.\delta, C_r.\sigma \leftarrow m_r, S_r, u_r, \mu_r, \delta_r, \sigma_r$// New cluster
  params
```

```
A 8: $[u, S'] = \text{ShapeExtraction}(X, u, S)$
  Input  : $X$ is an $n$-by-$t$ matrix of $z$-normalized time series
           $u$, $S$ are prior centroid and shape matrix of cluster.
  Output : $u'$ is new $t$-by-1 centroid.
           $S'$ is new $t$-by-$t$ shape matrix.
  // Align time series to prior centroid
  $\hat{X} \leftarrow []$
  for $i \leftarrow 1$ to $n$ do
      $[\text{dist}, \hat{x}] \leftarrow \text{SBD}(u, X(i))$
      $\hat{X} \leftarrow [\hat{X}, \hat{x}]$
  end
  $S' \leftarrow \hat{X}^T \cdot \hat{X} + S$ // incrementally updated Eq.73
  $Q \leftarrow I - \frac{1}{t} \cdot O$ // I, O are identity & ones matrices respectively
  $M \leftarrow Q^T \cdot S' \cdot Q$
  $u' \leftarrow \text{eig}(M, 1)$// Eq.74
```

```
A 9: $[m_r, \mu_r, \delta_r, \sigma_r] = \text{Update}(IDX_r, \text{mindist}, m_{r-1}, \mu_{r-1}, \delta_{r-1}, \sigma_{r-1})$
  Input  : $X_r$, $IDX_r$, mindist as defined in Alg. 7
           $m_{r-1}$, $\mu_{r-1}$, $\delta_{r-1}$, $\sigma_{r-1}$ prior cluster parameters
  Output : $m_r$, $\mu_r$, $\delta_r$, $\sigma_r$ are new cluster parameters
  for $j \leftarrow 1$ to $k$ do
      $\text{count}, s, ss \leftarrow 0, 0, 0$
      for $i \leftarrow 1$ to $n_r$ do
          if $IDX_r(i) = j$ then
              $\text{count} = \text{count} + 1$
              $s = s + \text{mindist}(i)$
              $ss = ss + \text{mindist}(i)^2$
          end
      end
      $m_r(j) \leftarrow m_{r-1}(j) + \text{count}$ // Eq.72
      $\mu_r(j) \leftarrow \frac{m_{r-1}(j) \cdot \mu_{r-1}(j) + s}{m_{r-1}(j) + \text{count}}$ // Eq.75
      $\delta_r(j) \leftarrow \frac{m_{r-1}(j) \cdot \delta_{r-1}(j) + ss}{m_{r-1}(j) + \text{count}}$ // Eq.76
      $\alpha \leftarrow \frac{m_{r-1}(j) + \text{count}}{1 + m_{r-1}(j) + \text{count}}$ // Smoothing factor Eq.78
      $\sigma_r(j) \leftarrow \alpha \sqrt{\delta_r(j) - \mu_r(j)^2} + (1 - \alpha)$ // Eq.77
  end
```

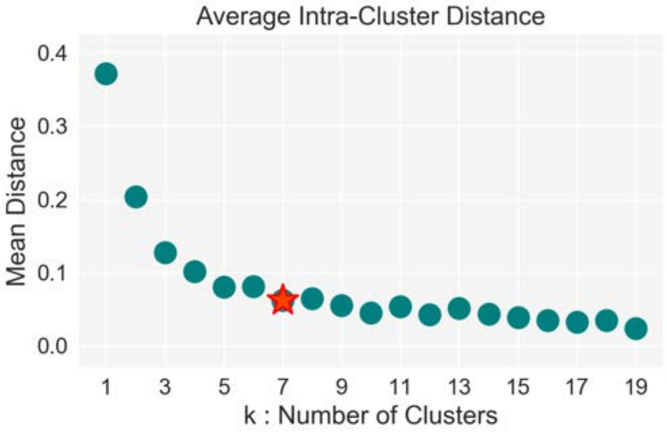


Figure 52: Average intra-cluster distance for different choices of k. This analysis on the *first* batch of data is useful for choosing k: here we choose $k = 7$, just after the "knee" of the curve.

5.2.3 Demonstration

This section demonstrates k-ShapeStream applied to open-access voltage magnitude measurements from a single µPMU on an operational distribution feeder in California, obtained through the NI4AI project platform[2] [201]. For this demonstration, similar to [16], [79], events are found by searching for sharp, significant changes in voltage magnitude streams. A window of 2 seconds (240 samples) around each event point is extracted from the measurement stream. These time series chunks are the inputs to k-ShapeStream. Note that k-ShapeStream is for post-detection event analysis and is agnostic to the preceding event detection algorithm. We use a simple approach, but there are a multitude of other options, such as baydetect, or the methods in [73], [82], [93], [94].

To emulate a streaming situation, the full data set of over 700 events found across four months of measurement is split into batches of 30 events each. k-ShapeStream ingests and clusters one batch at a time. To perform the clustering, the user must select k: the number of clusters to find. This can be a thorny step. For realism, we choose k based only on the first batch of data by clustering it with several values of k and considering the average intra-cluster distance—the average of the SBD between each time series

---

[2] https://ni4ai.org/

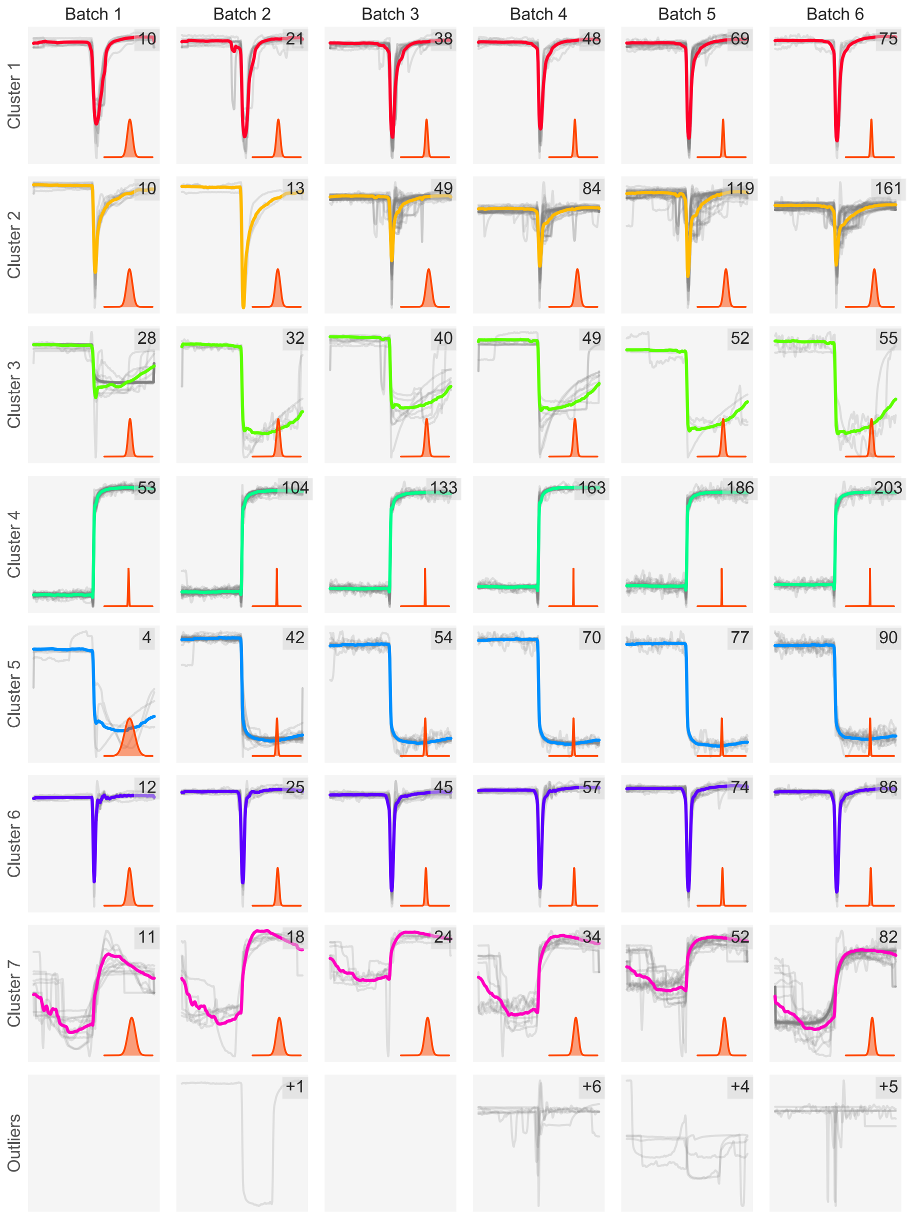


Figure 53: Seven clusters and the set of outliers visualized over six rounds of clustering. Gray lines indicate individual events. Colored lines show the cluster centroids. The inset number shows the number of events added to the cluster in each round. The inset distribution visualizes how intra-cluster distance distributions evolve: notice the narrowing distribution representing increasing certainty in the event's signature.

and its centroid—for each choice of k. The results are shown in Fig. 52, based on which we choose $k = 7$, as it lies just after the "knee" of the curve. Fig. 53 shows the resulting clusters and outliers, over six batches or rounds of clustering (columns correspond to rounds, rows to each cluster). The colored lines show cluster centroids, while gray lines show the individual events added to each cluster. The number in the top right corner indicates the number of events added to the cluster in that round. The inset distributions in orange visualize the distribution of intra-cluster distances, which evolve with each round. The distributions clearly and compellingly inform on the quality and nature of the clusters. Some clusters have very narrow distributions—indicating a highly regular, recurring signature—while others have wider distributions—reflecting a more irregular or perhaps poorer quality cluster. The outlier events differ emphatically from the centroids.

5.2.4 Example Use Case

To showcase the utility of k-ShapeStream, we perform analyses on some recognizable clusters from the full set of seven. These examples are not meant to present technically novel methods for system monitoring. Instead, we hope to illustrate how k-ShapeStream enables identification and analysis of recurring grid events and can be easily integrated into an analysis workflow to support a human analyst.

Transformer Tap Events

Load tap changing transformers (LTCs), common at distribution substations between medium and low voltage, mechanically adjust the effective turns ratio between their primary and secondary coils. They periodically "tap" the voltage up or down to compensate for changing voltage drop due to load variation, thus maintaining customer voltages within permissible limits. LTC failures can be costly and highly disruptive, motivating transformer monitoring [203]. Analysis of LTC operation based on PMU data has been *manually* demonstrated in the past [204]. k-ShapeStream can be used to automatically identify LTC tap events. Two clusters found in the data showing sharp step changes in voltage clearly correspond to LTCs operating to step voltage up and down (Fig. 54(a)-(b)). Notice the narrowness of the intra-cluster distance distributions indicating the high regularity of the LTC signatures. Once the signatures are temporally isolated, different event features can be analyzed. We consider the pre-event voltage (Fig. 54(c)), voltage change during the event (Fig. 54(d)), and the time of event occurrence (Fig. 54(e)). For this set of LTC operations, all these features seem normal: magnitudes are generally lower preceding a tap up operation than a tap down, the size of the voltage step is highly regular, and tap ups tend to occur later in the day while tap downs occur earlier (as we would expect under a typical residential feeder load profile). Such an analysis could reveal irregular transformer behavior. For example, the intra-cluster distance distribution found by k-ShapeStream could be used to reveal an anomalous LTC signature that might indicate incipient failure. Note that no prior knowledge whatsoever about LTCs was required for the algorithm to suggest the relevant clusters.

Voltage Sag Events

Voltage sags are large transient dips in a network voltage magnitude that can last from less than a cycle to several seconds. They may be caused by motor starts, equipment misoperation, or faults [205], including dangerous high-impedance faults that fail to trip overcurrent protection. Recurrent sags could be caused by repeated vegetation contact and indicate a fire hazard. Large, long, or frequent voltage sags are also problematic in that they may cause sensitive loads and solar PV inverters to trip offline. Monitoring voltage sags is therefore important for maintaining safety and reliability. We find one cluster containing a sharp, transient voltage drop that corresponds to a recurring voltage sag signature (55(a)). Notice from the intra-cluster distance distribution that while this event signature is quite regular, it is less so than the LTC tap signatures, as expected when comparing a random event to equipment operation. A voltage sag feature with implications for reliability is sag magnitude: the minimum voltage magnitude attained during the event. A result in the literature based on a simplified, generic model of fault-induced sags posits that the normalized frequency of sags with magnitude M will be proportional to $\frac{M}{1-M}$ [202]. Fig. 55(b) compares this model to the empirical distribution of sag magnitudes in the cluster found by k-ShapeStream. The model appears to describe the empirical distribution quite well, indicating its efficacy for prediction and monitoring on this feeder. Again, k-ShapeStream produces a characterization of voltage sag type and frequency, and thereby generates possible insights into physical occurrences, in an entirely unsupervised learning process.

5.2.5 Conclusion

The analyses of Section 5.2.4 highlight the efficacy of k-ShapeStream for identifying recurring and unusual (“outlier") event signatures in grid data. Once identified by k-ShapeStream, these signatures can then be analyzed further to identify issues, understand system behaviour, and improve overall situational awareness. Without such a streaming clustering approach, event signatures would have to be identified manually, which is always time consuming and sometimes impossible. Furthermore, k-ShapeStream generates highly intuitive results including a distribution that reflects the degree of confidence in a given cluster. These features make the algorithm particularly suitable for assisting and collaborating with a human user, which is essential in the electric grid context.

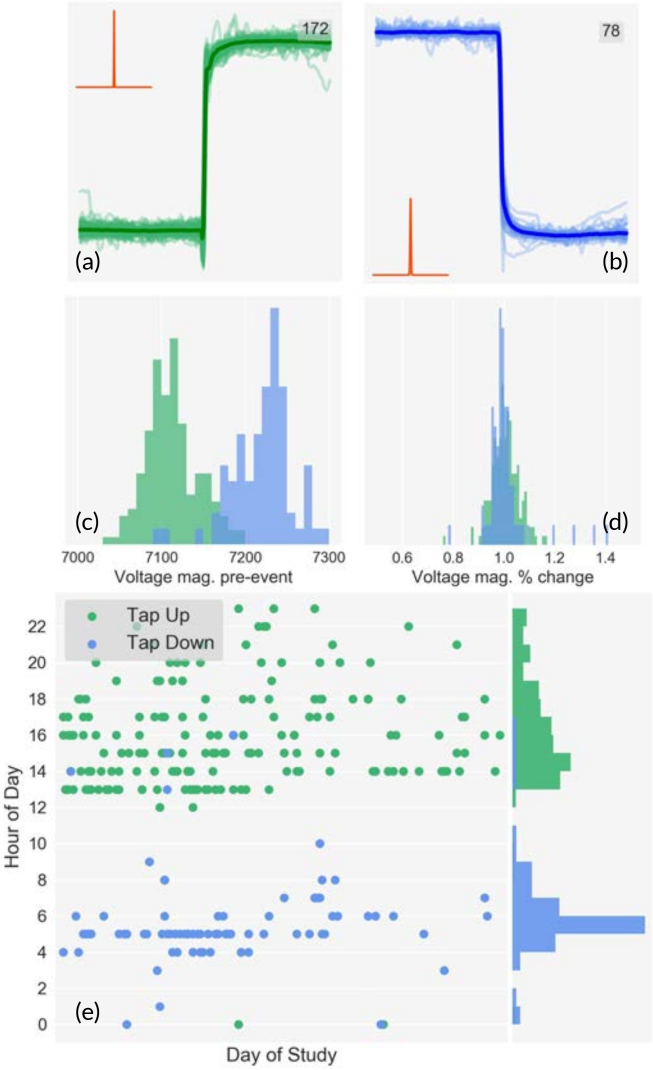


Figure 54: Analyzing clusters containing LTC operation events. Tap up (a) and down (b) signatures clustered together across several months of data and multiple rounds of clustering. Distribution insets are very narrow indicating highly regular event signatures. (c) Voltage magnitude preceding tap event, showing lower magnitudes for tap up events and higher magnitudes for tap down events. (d) Histogram of percent change in voltage during event showing highly regular step size. (e) Occurrence of tap up and down events over study period, with histograms showing hourly distribution. Tap up events tend to occur later in the day while tap down events tend to occur earlier, as is expected under a typical residential load profile.

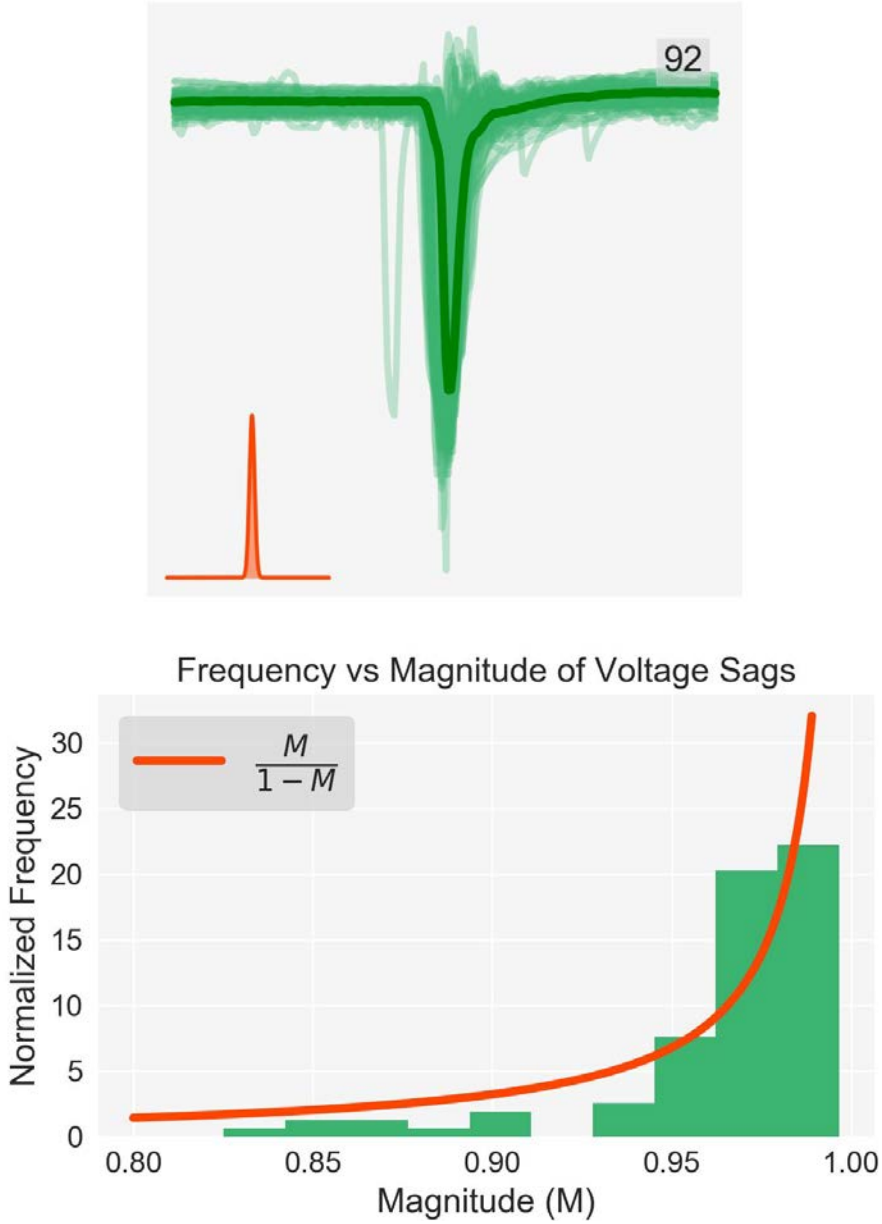


Figure 55: Analyzing cluster containing voltage sags. (a) A cluster containing 92 voltage sag events found over several months of data and multiple rounds of clustering. The inset distribution of intra-cluster distances is wider than for the tap events in Fig. 54(a),(b) indicating less consistent event signatures. (b) Comparison of the empirical distribution of sag magnitudes in the cluster to a theoretical model proposed in [202].

6 PRINCIPLES FOR USABLE TOOLS

The previous chapters covered much ground in defining and describing usable tools for grid monitoring. The tool algorithms in this thesis targeted two areas: topology monitoring and event detection and classification. Much works remains in expanding the grid operator and engineer's usable toolkit, not just in these areas but for a plethora of other use cases. Chapter 3 defined a wish-list of usability features, but a wish-list does not comprise a plan. How can we actually build more usable tools across applications? Drawing on the experience of developing the tools described in prior chapters, this final chapter presents some broad principles that facilitate the creation of tools to meet the criteria of usability, highlighting in particular the enabling features of high resolution, time synchronized measurements in getting there.

6.1 LINEAR MODELS

Linear models were crucial in the design of several of the topology tools in Chapter 4. They are tractable models capturing both the system and measurements from which we can derive physically justified tools that deal in intuitive physical quantities. Linear models also enable the principled study of the impacts of noise and other real world non-idealities on tool results.

There are many choices of linear models to use in the power systems space. Single phase, phasor quantities naturally obey a linear relationship parametrized by the network model, expressed in the now familiar matrix version of Ohm's Law:

$$\mathbf{I} = \mathbf{Y}\mathbf{V}$$

This model was the foundation of the physics-based topology estimation algorithm of Section 4.3. An extension, with a more elaborate $\mathbf{Y}$ matrix, was the basis of the three-phase topology estimation algorithm detailed in Section 4.4. In both the single and three phase case, it was the use of a linear model with an explicit, intuitive, well elucidated structure—recall the intuitive interpretation of the values in the Kron reduced impedance matrix—that was essential for successful tool design.

In the absence of phasor measurements, power system physics are generally non-linear. For example, real and reactive power injections (p and q) are related to nodal voltage magnitudes and angles (v and θ) through the famously non-linear and non-convex power flow equations.

$$p_k = \sum_{j=1}^{n} v_k v_j \big(g_{kj} \cos(\theta_k - \theta_j) + b_{jk} \sin(\theta_k - \theta_j) \big)$$

$$q_k = \sum_{j=1}^{n} v_k v_j \big(g_{kj} \sin(\theta_k - \theta_j) - b_{jk} \cos(\theta_k - \theta_j) \big)$$

Even so, there are numerous linearizations of these equations—based on well-defined assumptions and accurate across many scenarios—that can be useful starting points for tool design [65].

The coupling of high resolution, time aligned measurements with linear measurement models is especially enabling for the design of usable tools, and in particular for allowing low sensor coverage requirements. Consider the following toy example, which aims to localize the source of an injection change to the nearest node in a radial network, using only nodal voltages.

Toy Example: Localizing an Injection Change

Ohm's law for a network mapping from current injections to nodal voltages is:

$$\mathbf{V}(t) = \mathbf{Z}\mathbf{I}(t)$$

Here, $\mathbf{Z}$ is the system impedance matrix and $\mathbf{V}$ and $\mathbf{I}$ are $n \times 1$ vectors of the nodal voltage and current phasors across the network at one point in time t. Differentiating voltage and current in time produces the following relationship between changes in voltage and changes in current, which we compress into a more convenient notation

$$\mathbf{V}(t+1) - \mathbf{V}(t) = \mathbf{Z}\Big(\mathbf{I}(t+1) - \mathbf{I}(t) \Big) \rightarrow \mathbf{V}^{(\delta t)} = \mathbf{Z}\mathbf{I}^{(\delta t)}$$

Assume that we only have access to the left hand side of this model, specifically the nodal voltage changes $\mathbf{V}^{(\delta t)}$. Suppose there is a rapid, significant change in the current injection at a single node k between time t and $t+1$. If the measurement time resolution is low—i.e. t to $t+1$ is a long duration—many other injection changes at other nodes will appear coincident from the measurement perspective. However, with high enough measurement resolution—such as that delivered by PMUs—it is reasonable to assume that the change at node k is the only change occurring within this time step. In this case, $\mathbf{I}^{(\delta t)}$ will be sparse, with a single non-zero value at the k-th position. The impedance matrix $\mathbf{Z}$ is non-sparse, therefore the lone non-zero value in $\mathbf{I}^{(\delta t)}$ will ripple across $\mathbf{V}^{(\delta t)}$ to produce many non-zero voltage changes, eliminating the direct indication of the source. However, here the structural properties of $\mathbf{Z}$ are crucial. It can be shown that the diagonal elements of $\mathbf{Z}$ have the largest magnitude in each row [173]. Therefore, the largest element of $\mathbf{V}^{(\delta t)}$ will in fact be the one corresponding to the source node of the injection change, allowing the successful localization of the change only from voltage data.

The simple localization tool in this example relies on a linear model with clear structure, combined with high resolution measurement data that must also be time synchronized to be at all useful in the model. All together, these elements enable a localization tool that deals in physically meaningful quantities, is theoretically well-founded, and requires only voltage measurements—i.e. a very usable one.

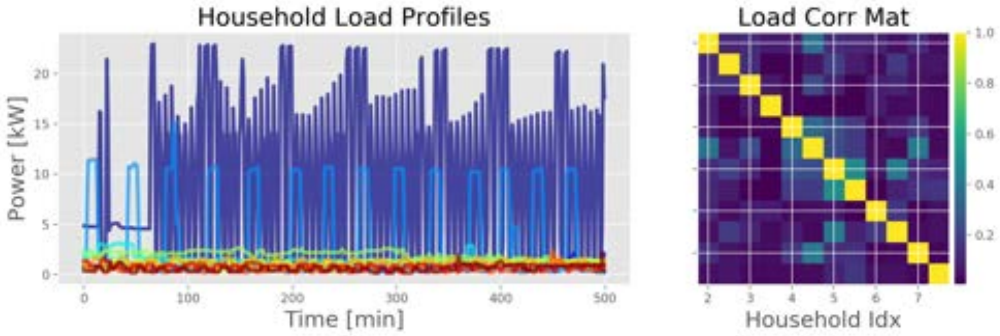


Figure 56: Studying correlations in household real power demand using minute resolution demand data from the Pecan Street project. Individual load profiles for 12 homes over several hours are shown at left. The absolute value of the Pearson correlation coefficient between every pair of loads is visualized in the matrix on the right. A value of 1 means two load profiles are highly positively or negatively correlated, while a 0 indicates no correlation.

6.2 STATISTICAL ASSUMPTIONS

Statistical assumptions can be extremely useful for making the problem of usable tool design tractable. Assumptions on the statistics of current injections (and implicitly demand), were vital for creating the three phase and justified heuristic topology tools in Chapter 4. Assuming that loads are uncorrelated—and therefore that current injections are uncorrelated—is a common and highly effective assumption. While aggregate load behavior or load variation on long time scales is highly correlated, the uncorrelated assumption often holds very well on individual loads or in high resolution data like that from PMUs. Fig. 56 visually validates this assumption on minute resolution household demand data from the Pecan Street data set[1]. The correlation matrix visualized in the right of Fig. 56 contains the absolute value of the Pearson correlation coefficients between household load time series. Its strong diagonal and relatively small off-diagonal elements support the claim of individual loads being weakly correlated and bolsters the validity of the uncorrelated load assumption.

The following toy example illustrates how the correlation assumption en-

---

[1] https://www.pecanstreet.org/

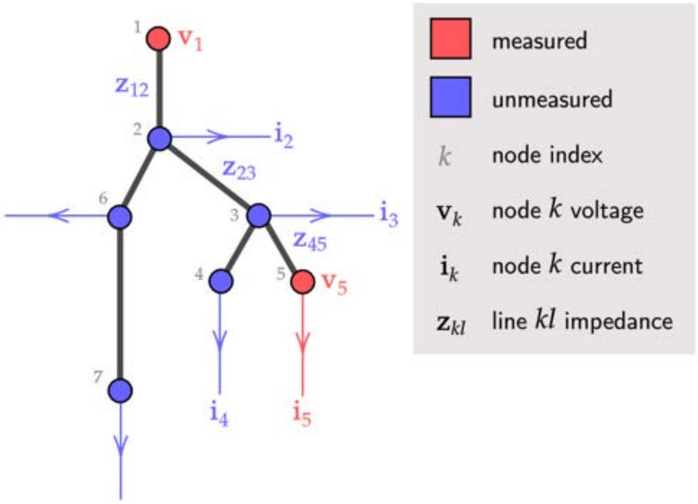


Figure 57: The setup of the toy example for estimating a path impedance using limited phasor measurements and the assumption of uncorrelated current extractions.

ables the development of usable tools in lower sensor coverage scenarios.

Toy Example: Estimating a Path Impedance

Consider an arbitrary radial network with only a voltage phasor measurement at the substation, and a measurement of the voltage and current injection phasors at a single leaf or edge node. This setup is visualized in Fig. 57, where the substation is node 1 and the measured leaf node is 5. The relevant voltage, current, and impedance quantities are explicitly labeled, and measured quantities are colored in red while unmeasured quantities are colored blue. Suppose we are interested in $\mathbf{z}_{15}$, the impedance of the path between the measured substation and leaf node. This quantity can be written as the sum of individual line impedances:

$$\mathbf{z}_{15} = \mathbf{z}_{12} + \mathbf{z}_{23} + \mathbf{z}_{35}$$

Conventionally, we would need to estimate each of these line impedances individually—requiring measurements from both ends of each line—in order to determine $\mathbf{z}_{15}$. However, if we make the assumption that current extractions are uncorrelated, the problem becomes tractable with the available data.

By Ohm's Law, the voltage difference between the substation and leaf 5 can be written in terms of line impedances and current extractions as follows:

$$\mathbf{v}_1 - \mathbf{v}_5 = \mathbf{z}_{12}\left(\sum_{k=2}^{5} \mathbf{i}_k\right) + \mathbf{z}_{23}\left(\sum_{k=3}^{5} \mathbf{i}_k\right) + \mathbf{z}_{45}\mathbf{i}_5$$

Then the correlation between voltage difference $\mathbf{v}_1 - \mathbf{v}_5$ and current extraction $\mathbf{i}_5$ can be written as:

$$(\mathbf{v}_1 - \mathbf{v}_5)^H \mathbf{i}_5 = \mathbf{z}_{12}\left(\sum_{k=2}^{5} \mathbf{i}_k^H \mathbf{i}_5\right) + \mathbf{z}_{23}\left(\sum_{k=3}^{5} \mathbf{i}_k^H \mathbf{i}_5\right) + \mathbf{z}_{45}(\mathbf{i}_5^H \mathbf{i}_5) \tag{79}$$

Since current extractions are uncorrelated between nodes, several terms on the right hand side of (79) disappear, leaving:

$$(\mathbf{v}_1 - \mathbf{v}_5)^H \mathbf{i}_5 = (\mathbf{z}_{12} + \mathbf{z}_{23} + \mathbf{z}_{45})\mathbf{i}_5^H \mathbf{i}_5$$

This then allows for the determination of the path impedance of interest using only the three available phasor measurements:

$$\mathbf{z}_{15} = \frac{(\mathbf{v}_1 - \mathbf{v}_5)^H \mathbf{i}_5}{\mathbf{i}_5^H \mathbf{i}_5}$$

6.3 STATISTICAL BASELINES

Statistical baselines were used for event detection in Chapter 5. More broadly, we can envision them as adding a final layer of robustness to tools, and thereby improving usability. The linear model and statistical assumptions involved in the derivation of a tool will rarely hold exactly, leading to noisy variation in the tool output. An overlaid statistical baseline can distinguish this noisy variation from truly significant changes that demand urgent human attention. This can address issues around excessive alarms and the resulting cognitive overload that currently challenges grid operators [206]. At the same time, statistical baseline distributions are highly intuitive and do not diminish tool transparency.

6.4 A VISION FOR TOOLS

We have reached the end of this thesis' journey towards usable tools. There is still far to go to achieve the real-time grid visibility needed to meet the challenges posed by a transforming energy system. My hope is that the usability criteria, tool examples, and design strategies proposed in this thesis provide a road map for the creation of future grid tools that work in successful collaboration with humans to safely, efficiently, reliably, and renewably deliver energy to all the world's people.

BIBLIOGRAPHY

[1] N. E. Observatory. (2012). “City lights 2012,” [Online]. Available: `https://commons.wikimedia.org/w/index.php?curid=77474429`.

[2] J. Wirfs-Brock. (2015). “How long is your blackout?” [Online]. Available: `http://insideenergy.org/2015/03/20/ie-questions-how-long-is-your-blackout` (visited on 02/14/2021).

[3] P. Fairley, “Germany’s grid: Renewables-rich and rock-solid,” *IEEE Spectrum*, vol. 28, 2014.

[4] S. von Roon, M. Sutter, F. Samweber, and K. Wachinger, “Network expansion in germany,” *What are new power grids needed for ″0*, 2014.

[5] O. Edenhofer, R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, *et al.*, “Contribution of working group iii to the fifth assessment report of the intergovernmental panel on climate change,” *Climate change*, pp. 1–11, 2014.

[6] I. G. E. Outlook, *Entering the decade of electric drive*, 2020.

[7] C. Bataille, M. Åhman, K. Neuhoff, L. J. Nilsson, M. Fischedick, S. Lechtenböhmer, B. Solano-Rodriquez, A. Denis-Ryan, S. Stiebert, H. Waisman, *et al.*, “A review of technology and policy deep decarbonization pathway options for making energy-intensive industry production consistent with the paris agreement,” *Journal of Cleaner Production*, vol. 187, pp. 960–973, 2018.

[8] A. Arif, Z. Wang, J. Wang, B. Mather, H. Bashualdo, and D. Zhao, “Load modeling—a review,” *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5986–5999, 2017.

[9] Z. Wang, X. Guo, J. Li, and X. Wang, “Impact of voltage sags on electric-vehicle charger and critical voltage sag determination,” *IEEE Transactions on Power Delivery*, vol. 31, no. 3, pp. 1397–1399, 2015.

[10] R. Guttromson and M. Behnke, "Momentary cessation: Improving dynamic performance and modeling of utility-scale inverter based resources during grid disturbances.," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2020.

[11] G. Wong-Parodi, "When climate change adaptation becomes a "looming threat" to society: Exploring views and responses to california wildfires and public safety power shutoffs," *Energy Research & Social Science*, vol. 70, p. 101 757, 2020.

[12] M. O. Román, E. C. Stokes, R. Shrestha, Z. Wang, L. Schultz, E. A. S. Carlo, Q. Sun, J. Bell, A. Molthan, V. Kalb, *et al.*, "Satellite-based assessment of electricity restoration efforts in puerto rico after hurricane maria," *PloS one*, vol. 14, no. 6, e0218883, 2019.

[13] S. Littlechild and L. Kiesling, "Hayek and the texas blackout," *The Electricity Journal*, vol. 34, no. 6, p. 106 969, 2021.

[14] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, 2016.

[15] S. Murphy, K. Jones, T. Laughner, M. Bariya, and A. von Meier, "Accelerating artificial intelligence on the grid," in *2020 Clemson University Power Systems Conference (PSC)*, IEEE, 2020, pp. 1–7.

[16] M. Bariya, S. Murphy, K. Jones, T. Laughner, and M. Anderson, "Analytics at warp speed – from prototypes to production," in *CIGRE Grid of the Future Symposium*, CIGRE, 2018.

[17] M. Bariya, K. Moffat, and A. von Meier, "Empirical noise estimation in distribution synchrophasor measurements," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, IEEE, 2019, pp. 1–7.

[18] T. Hong and M. Shahidehpour, "Load forecasting case study," *EISPC, US Department of Energy*, 2015.

[19] S. Bennett, *A history of control engineering, 1930-1955*, 47. IET, 1993.

[20] M. B. Cain, R. P. O'neill, A. Castillo, *et al.*, "History of optimal power flow and formulations," *Federal Energy Regulatory Commission*, vol. 1, pp. 1–36, 2012.

[21] R. C. Lanphier, *Electric meter history and progress*. Sangamo Electric Company, 1925.

[22] H. L. Smith, "A brief history of electric utility automation systems," *Electric Energy T&D Magazine*, vol. 14, pp. 39–44, 2010.

[23] A. Phadke and J. Thorp, "History and applications of phasor measurements," in *2006 IEEE PES Power Systems Conference and Exposition*, IEEE, 2006, pp. 331–335.

[24] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 33–43, 2012.

[25] A. Silverstein, "Diagnosing equipment health and mis-operations with pmu data," May 2015.

[26] B. K. Sovacool, P. Kivimaa, S. Hielscher, and K. Jenkins, "Vulnerability and resistance in the united kingdom's smart meter transition," *Energy Policy*, vol. 109, pp. 767–781, 2017.

[27] F. D. Garcia, F. P. Marafão, W. A. de Souza, and L. C. P. da Silva, "Power metering: History and future trends," in *2017 Ninth Annual IEEE Green Technologies Conference (GreenTech)*, IEEE, 2017, pp. 26–33.

[28] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, "Identifying topology of low voltage distribution networks based on smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5113–5122, 2017.

[29] L. Peretto, "The role of measurements in the smart grid era," *IEEE Instrumentation & Measurement Magazine*, vol. 13, no. 3, pp. 22–25, 2010.

[30] P. Bornard, J. Tesseron, J. Bastide, and M. Nourris, "Field experience of digital fault recorders and distance relay in ehv substations," *IEEE transactions on power apparatus and systems*, no. 1, pp. 133–139, 1984.

[31] S. Chakrabarti, E. Kyriakides, T. Bi, D. Cai, and V. Terzija, "Measurements get together," *IEEE Power and Energy Magazine*, vol. 7, no. 1, pp. 41–49, 2008.

[32] T. S. Sidhu, X. Zhang, and V. Balamourougan, "A new half-cycle phasor estimation algorithm," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1299–1305, 2005.

[33] J. Thorp, A. Phadke, and K. Karimi, "Real time voltage-phasor measurement for static state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 11, pp. 3098–3106, 1985.

[34] H. Kirkham and A. Riepnieks, "Measurement of phasor-like signals," *Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-25643*, 2016.

[35] H. Kirkham and A. Riepnieks, "Dealing with non-stationary signals: Definitions, considerations and practical implications," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, 2016, pp. 1–5.

[36] A. Riepnieks and H. Kirkham, "An introduction to goodness of fit for pmu parameter estimation," *IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp. 2238–2245, 2016.

[37] M. Weiss, A. Silverstein, F. Tuffner, and Y.-S. Li-Baboud, "The use and challenges of precise time in electric power synchrophasor systems," in *Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting*, 2017, pp. 351–366.

[38] S. Mukhopadhyay, "Indian experience of smart grid applications in transmission and distribution system," in *2014 6th IEEE Power India International Conference (PIICON)*, IEEE, 2014, pp. 1–6.

[39] A. G. Phadke, "Synchronized phasor measurements - a historical overview," in *IEEE/PES transmission and distribution conference and exhibition*, IEEE, vol. 1, 2002, pp. 476–479.

[40] Y. Liu, W. Yao, D. Zhou, L. Wu, S. You, H. Liu, L. Zhan, J. Zhao, H. Lu, W. Gao, *et al.*, "Recent developments of fnet/grideye—a situational awareness tool for smart grid," *CSEE Journal of Power and Energy Systems*, vol. 2, no. 3, pp. 19–27, 2016.

[41] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, *Micro-synchrophasors for distribution systems*. IEEE, 2014.

[42] A. von Meier, E. Stewart, A. McEachern, M. Andersen, and L. Mehrmanesh, "Precision micro - synchrophasors for distribution systems: A summary of applications," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2926–2936, 2017.

[43] A. Shahsavari, M. Farajollahi, E. Stewart, C. Roberts, and H. Mohsenian-Rad, "A data-driven analysis of lightning-initiated contingencies at a distribution grid with a pv farm using micro-pmu data," in *2017 North American Power Symposium (NAPS)*, IEEE, 2017, pp. 1–6.

[44] A. Shahsavari, M. Farajollahi, E. Stewart, A. von Meier, L. Alvarez, E. Cortez, and H. Mohsenian-Rad, "A data-driven analysis of capacitor bank operation at a distribution feeder using micro-pmu data," in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2017, pp. 1–5.

[45] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in pmu data," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, 2016, pp. 1–5.

[46] F. F. Wu and W.-H. Liu, "Detection of topology errors by state estimation (power systems)," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 176–183, 1989.

[47] A. Gomez-Exposito and A. Abur, "Power system state estimation: Theory and implementation," *CRC press*, 2004.

[48] K. R. Shih and S. J. Huang, "Application of a robust algorithm for dynamic state estimation of a power system," *IEEE Power Engineering Review*, vol. 22, p. 70, 2002.

[49] A. Sinha and J. Mondal, "Dynamic state estimator using ann based bus load prediction. ieee transactions on power systems," *IEEE Transactions on Power Systems*, vol. 14, pp. 1219–1225, 1999.

[50] W. Kersting, "Radial distribution test feeders," *Power Engineering Society Winter Meeting, 2001. IEEE*, vol. 2, pp. 908–912, 2001.

[51] C. M. Roberts, C. M. Shand, K. W. Brady, E. M. Stewart, A. W. McMorran, and G. A. Taylor, "Improving Distribution Network Model Accuracy using Impedance Estimation from Micro-Synchrophasor Data," pp. 1–5, 2016.

[52] P. Pegoraro, P. Castello, C. Muscas, K. Brady, and A. von Meier, "Handling instrument transformers and pmu errors for the estimation of line parameters in distribution grids," *2017 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, Sep. 2017.

[53] M. P. Andersen, S. Kumar, C. Brooks, A. von Meier, and D. E. Culler, "Distil: Design and implementation of a scalable synchrophasor data processing system," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2015, pp. 271–277.

[54] J. H. Enslin, S. G. Whisenant, and R. Hadidi, "Third egrid workshop maps the grid of the future: Attendees engage to examine the role of power electronic applications in modern electric power systems," *IEEE Power Electronics Magazine*, vol. 6, no. 1, pp. 48–55, 2019.

[55] A. Liao, E. Stewart, and E. Kara, "Micro-synchrophasor data for diagnosis of transmission and distribution level events," in *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, May 2016, pp. 1–5.

[56] M. of Information Photo Division Photographer. (1944). "Pastry making demonstration at the national training college of domestic science, westminster, london, 1944.," [Online]. Available: `http://media.iwm.org.uk/iwm/mediaLib//44/media-44335/large.jpg`.

[57] Unknown. (1875). "A charkha or spinning wheel, jammu," [Online]. Available: `http://digitallibrary.usc.edu/cdm/ref/collection/p15799coll123/id/64857`.

[58] ——, (1910). "Elsa von kalmár (1876–1956)," [Online]. Available: `https://commons.wikimedia.org/w/index.php?curid=74250876`.

[59] D. w. (2012). "Ercotoperator 2," [Online]. Available: `https://commons.wikimedia.org/w/index.php?curid=257312931`.

[60] A. von Meier, "Occupational cultures as a challenge to technological innovation," *IEEE Transactions on engineering management*, vol. 46, no. 1, pp. 101–114, 1999.

[61] O. Ardakanian, Y. Yuan, R. Dobbe, A. von Meier, S. Low, and C. Tomlin, "Event detection and localization in distribution grids with phasor measurement units," in *2017 IEEE Power & Energy Society General Meeting*, IEEE, 2017, pp. 1–5.

[62] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato, "Identification of power distribution network topology via voltage correlation analysis," in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 1659–1664.

[63] K. G. Ravikumar and A. K. Srivastava, "Zero-setting algorithm for high-speed open line detection using synchrophasors," in *2018 IEEE PES Transmission and Distribution Conference and Exposition (T&D)*, IEEE, 2018, pp. 1–9.

[64] J. E. Tate and T. J. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.

[65] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2015, pp. 402–409.

[66] W. Ju, N. Nayak, C. Vikram, H. Silva-Saravia, K. Sun, and G. Zu, "Indices for automated identification of questionable generator models using synchrophasors," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2020, pp. 1–5.

[67] R. Arghandeh, M. Gahr, A. von Meier, G. Cavraro, M. Ruh, and G. Andersson, "Topology detection in microgrids with micro - synchrophasors," in *2015 IEEE Power & Energy Society General Meeting*, IEEE, 2015, pp. 1–5.

[68] M. Farajollahi, A. Shahsavari, E. M. Stewart, and H. Mohsenian-Rad, "Locating the source of events in power distribution systems using micro-pmu data," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6343–6354, 2018.

[69] Y. Ge, A. J. Flueck, D.-K. Kim, J.-B. Ahn, J.-D. Lee, and D.-Y. Kwon, "Power system real-time event detection and associated data archival reduction based on synchrophasors," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2088–2097, 2015.

[70] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally placed mupmu sensors in distribution grids," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3611–3623, 2017.

[71] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, "A novel event detection method using pmu data with high precision," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 454–466, 2018.

[72] T. A. Ferryman and B. G. Amidan, “Investigation of phase angle differences using statistical analysis of real world state estimator data,” in *2012 45th Hawaii International Conference on System Sciences*, IEEE, 2012, pp. 2062–2071.

[73] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, “Wavelet-based event detection method using pmu data,” *IEEE Transactions on Smart grid*, vol. 8, no. 3, pp. 1154–1162, 2015.

[74] S. M. H. Rizvi, S. K. Sadanandan, and A. K. Srivastava, “Real-time zip load parameter tracking using adaptive window and variable elimination with realistic synthetic synchrophasor data,” in *2020 IEEE Industry Applications Society Annual Meeting*, IEEE, 2020, pp. 1–8.

[75] E. C. Kara, C. M. Roberts, M. Tabone, L. Alvarez, D. S. Callaway, and E. M. Stewart, “Towards real - time estimation of solar generation from micro-synchrophasor measurements,” *arXiv:1607.02919*, 2016.

[76] S. Kantra, H. A. Abdelsalam, and E. B. Makram, “Application of pmu to detect high impedance fault using statistical analysis,” in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, 2016, pp. 1–5.

[77] T. A. Ferryman and B. G. Amidan, “Statistical analysis of abnormal electric power grid behavior,” in *2010 43rd Hawaii International Conference on System Sciences*, IEEE, 2010, pp. 1–8.

[78] K. Kirihara, K. E. Reinhard, A. K. Yoon, and P. W. Sauer, “Investigating synchrophasor data quality issues,” in *2014 Power and Energy Conference at Illinois (PECI)*, IEEE, 2014, pp. 1–4.

[79] D. B. Arnold, C. Roberts, O. Ardakanian, and E. M. Stewart, “Synchrophasor data analytics in distribution grids,” in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2017, pp. 1–5.

[80] S. A. R. Konakalla and R. A. de Callafon, “Feature based grid event classification from synchrophasor data,” *Procedia Computer Science*, vol. 108, pp. 1582–1591, 2017.

[81] O. P. Dahal, S. M. Brahma, and H. Cao, “Comprehensive clustering of disturbance events recorded by phasor measurement units,” *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1390–1397, 2013.

[82] Y. Zhou, R. Arghandeh, I. Konstantakopoulos, S. Abdullah, A. von Meier, and C. J. Spanos, "Abnormal event detection with high resolution micro-pmu data," in *2016 Power Systems Computation Conference (PSCC)*, IEEE, 2016, pp. 1–7.

[83] A. Ahmed, V. V. Krishnan, S. A. Foroutan, M. Touhiduzzaman, C. Rublein, A. Srivastava, Y. Wu, A. Hahn, and S. Suresh, "Cyber physical security analytics for anomalies in transmission protection systems," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 6313–6323, 2019.

[84] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee, "Ensemble-based algorithm for synchrophasor data anomaly detection," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2979–2988, 2018.

[85] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over pmu data streams," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, IEEE, 2015, pp. 1–8.

[86] S. Basumallik, R. Ma, and S. Eftekharnejad, "Packet-data anomaly detection in pmu-based state estimator using convolutional neural network," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 690–702, 2019.

[87] H. Li, Y. Weng, E. Farantatos, and M. Patel, "A hybrid machine learning framework for enhancing pmu-based event identification with limited labels," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, IEEE, 2019, pp. 1–8.

[88] T. Guo and J. V. Milanović, "Online identification of power system dynamic signature using pmu measurements and data mining," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1760–1768, 2015.

[89] J. Jiang, X. Zhao, S. Wallace, E. Cotilla-Sanchez, and R. Bass, "Mining pmu data streams to improve electric power system resilience," in *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, 2017, pp. 95–102.

[90] I. Niazazari and H. Livani, "Disruptive event classification using pmu data in distribution networks," in *2017 IEEE Power & Energy Society General Meeting*, IEEE, 2017, pp. 1–5.

[91] D. O. Amoateng, R. Yan, and T. K. Saha, “A deep unsupervised learning approach to pmu event detection in an active distribution network,” in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2020, pp. 1–5.

[92] A. Aligholian, A. Shahsavari, E. Cortez, E. Stewart, and H. Mohsenian-Rad, “Event detection in micro-pmu data: A generative adversarial network scoring method,” in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2020, pp. 1–5.

[93] L. Xie, Y. Chen, and P. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2784–2794, 2014.

[94] T. Xu and T. Overbye, “Real-time event detection and feature extraction using pmu measurement data,” in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2015, pp. 265–270.

[95] R. Meier, E. Cotilla-Sanchez, B. McCamish, D. Chiu, M. Histand, J. Landford, and R. B. Bass, “Power system data management and analysis using synchrophasor data,” in *2014 IEEE Conference on Technologies for Sustainability (SusTech)*, IEEE, 2014, pp. 225–231.

[96] R. Meier, B. McCamish, E. Cotilla-Sanchez, J. Landford, R. B. Bass, and D. Chiu, “Event detection using correlation within arrays of streaming pmu data,” in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2018, pp. 1–5.

[97] P. Gravois, *Synchrophasor based oscillation detection in ercot operations*, NASPI, Springfield, Massachusetts: ERCOT, Sep. 2017.

[98] “Forced oscillation monitoring & mitigation,” *North American Electric Reliabilty Corporation: Atlanta, GA, USA*, 2017.

[99] North American Synchrophasor Initiative, *"naspi oscillation detection and voltage stability tools technical workshop - houston, tx," oct. 22, 2014*. [Online]. Available: `https://www.naspi.org/node/440`.

[100] ——, “Using synchrophasor data for phase angle monitoring,” Tech. Rep. NASPI-2016-TR-003, 2016. [Online]. [Online]. Available: `https://www.naspi.org/sites/default/files/reference_documents/0.pdf`.

[101] North American Synchrophasor Initiative Control Room Solutions Task Team, "Using synchrophasor data during system islanding events and blackstart restoration," Tech. Rep., 2015. [Online]. [Online]. Available: https://www.naspi.org/sites/default/files/reference_documents/10.pdf?fileID=1537.

[102] Z. Huang, P. Du, D. Kosterev, and S. Yang, "Generator dynamic model validation and parameter calibration using phasor measurements at the point of connection," *IEEE transactions on power systems*, vol. 28, no. 2, pp. 1939–1949, 2013.

[103] M. H. Bollen, "Understanding power quality problems," in *Voltage sags and Interruptions*, IEEE press, 2000.

[104] G. Zenobi and P. Cunningham, "Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error," in *European Conference on Machine Learning*, Springer, 2001, pp. 576–587.

[105] J. D. Follum, F. K. Tuffner, L. A. Dosiek, and J. W. Pierre, "Power system oscillatory behaviors: Sources, characteristics, & analyses," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2017.

[106] North American Synchrophasor Initiative, "Diagnosing equipment health and mis-operations with pmu data," Tech. Rep., 2015. [Online]. [Online]. Available: https://www.naspi.org/sites/default/files/reference_documents/14.pdf?fileID=1530.

[107] K. D. Jones, *Getting beyond base camp: Scaling your synchrophasor data mountain*, NASPI, Philadelphia, Pennsylvania: Dominion, Oct. 2018. [Online]. Available: https://naspi.org/sites/default/files/2018-11/dominion_jones_pmu_scalability_20181024.pdf.

[108] E. Flandin. (1840). "Kizil hauzen bridge," [Online]. Available: https://commons.wikimedia.org/wiki/File:Kizil_Hauzen_Bridge_by_Eugene_Flandin.jpg.

[109] E. L. Lehmann, "The fisher, neyman-pearson theories of testing hypotheses: One theory or two?" *Journal of the American statistical Association*, vol. 88, no. 424, pp. 1242–1249, 1993.

[110] R. E. Mayer and R. Moreno, "Nine ways to reduce cognitive load in multimedia learning," *Educational psychologist*, vol. 38, no. 1, pp. 43–52, 2003.

[111] R. Klump, R. E. Wilson, and K. E. Martin, "Visualizing real-time security threats using hybrid scada/pmu measurement displays," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, IEEE, 2005, pp. 55c–55c.

[112] S. Walton, E. Maguire, and M. Chen, "A visual analytics loop for supporting model development," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, IEEE, 2015, pp. 1–8.

[113] M. Bariya, A. von Meier, A. Ostfeld, and E. Ratnam, "Data-driven topology estimation with limited sensors in radial distribution feeders," in *2018 IEEE Green Technologies Conference (GreenTech)*, IEEE, 2018, pp. 183–188.

[114] M. Bariya, K. Moffat, and A. von Meier, "Unsupervised impedance and topology estimation of distribution networks—limitations and tools," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 846–856, 2019.

[115] K. Moffat, M. Bariya, and A. von Meier, "Network impedance estimation for microgrid control using noisy synchrophasor measurements," in *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*, IEEE, 2018, pp. 1–6.

[116] M. Bariya, D. Deka, and A. von Meier, "Guaranteed phase & topology identification in three phase distribution grids," *IEEE Transactions on Smart Grid*, 2021.

[117] M. Bariya, K. Moffat, and A. von Meier, "Physically meaningful grid analytics on voltage measurements using graph spectra," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2020, pp. 1–5.

[118] G. Cavraro, R. Arghandeh, K. Poolla, and A. von Meier, "Data-driven approach for distribution network topology detection," in *2015 IEEE power & energy society general meeting*, IEEE, 2015, pp. 1–5.

[119] R. A. Sevlian, J. Yu, Y. Liao, X. Chen, Y. Weng, E. C. Kara, M. Tabone, S. Badri, C.-W. Tan, D. Chassin, *et al.*, "Vader: Visualization and analytics for distributed energy resources," *arXiv preprint arXiv:1708.09473*, 2017.

[120] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

[121] A. Monticelli, "Fast decoupled state estimator," in *State Estimation in Electric Power Systems*, Springer, 1999, pp. 313–342.

[122] G. N. Korres and N. M. Manousakis, "A state estimation algorithm for monitoring topology changes in distribution systems," in *2012 IEEE Power and Energy Society General Meeting*, IEEE, 2012, pp. 1–8.

[123] M. Liserre, F. Blaabjerg, and R. Teodorescu, "Grid impedance estimation via excitation of `lcl`-filter resonance," *IEEE Transactions on Industry Applications*, vol. 43, no. 5, pp. 1401–1407, 2007.

[124] G. Cavraro and V. Kekatos, "Graph algorithms for topology identification using power grid probing," *IEEE control systems letters*, vol. 2, no. 4, pp. 689–694, 2018.

[125] Y. Yuan, S. Low, O. Ardakanian, and C. Tomlin, "Inverse power flow problem," *arXiv preprint arXiv:1610.06631*, 2016.

[126] O. Ardakanian, V. W. Wong, R. Dobbe, S. H. Low, A. von Meier, C. J. Tomlin, and Y. Yuan, "On identification of distribution grids," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 950–960, 2019.

[127] J. Yu, Y. Weng, and R. Rajagopal, "Patopa: A data-driven parameter and topology joint estimation framework in distribution grids," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4335–4347, 2017.

[128] S. Park, D. Deka, S. Backhaus, and M. Chertkov, "Learning with end-users in distribution grids: Topology and parameter estimation," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1428–1440, 2020.

[129] D. Deka, S. Backhaus, and M. Chertkov, "Learning topology of the power distribution grid with and without missing data," in *2016 European Control Conference (ECC)*, IEEE, 2016, pp. 313–320.

[130] ——, "Structure learning in power distribution networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1061–1074, 2017.

[131] ——, "Estimating distribution grid topologies: A graphical learning based approach," in *2016 Power Systems Computation Conference (PSCC)*, IEEE, 2016, pp. 1–7.

[132] Y. Weng, Y. Liao, and R. Rajagopal, "Distributed energy resources topology identification via graphical modeling," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2682–2694, 2016.

[133] Y. Liao, Y. Weng, G. Liu, Z. Zhao, C.-W. Tan, and R. Rajagopal, "Unbalanced multi-phase distribution grid topology estimation and bus phase identification," *IET Smart Grid*, vol. 2, no. 4, pp. 557–570, 2019.

[134] D. Deka, M. Chertkov, and S. Backhaus, "Topology estimation using graphical models in multi-phase power distribution grids," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 1663–1673, 2019.

[135] W. Li, M. Wang, and J. H. Chow, "Fast event identification through subspace characterization of pmu data in power systems," in *2017 IEEE Power & Energy Society General Meeting*, IEEE, 2017, pp. 1–5.

[136] M. Wang, J. H. Chow, P. Gao, X. T. Jiang, Y. Xia, S. G. Ghiocel, B. Fardanesh, G. Stefopolous, Y. Kokai, N. Saito, *et al.*, "A low-rank matrix approach for the analysis of large amounts of power system synchrophasor data," in *2015 48th Hawaii International Conference on System Sciences*, IEEE, 2015, pp. 2637–2644.

[137] P. Gao, R. Wang, M. Wang, and J. H. Chow, "Low-rank matrix recovery from noisy, quantized, and erroneous measurements," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2918–2932, 2018.

[138] K. Mahapatra, N. R. Chaudhuri, and R. Kavasseri, "Bad data detection in pmu measurements using principal component analysis," in *2016 North American Power Symposium (NAPS)*, IEEE, 2016, pp. 1–6.

[139] Z. Mao, T. Xu, and T. J. Overbye, "Real-time detection of malicious pmu data," in *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*, IEEE, 2017, pp. 1–6.

[140] Z. Wang, Y. Zhang, and J. Zhang, "Principal components fault location based on wams/pmu measure system," in *2011 IEEE Power and Energy Society General Meeting*, IEEE, 2011, pp. 1–5.

[141] P. A. Pegoraro, P. Castello, C. Muscas, K. Brady, and A. von Meier, "Handling instrument transformers and pmu errors for the estimation of line parameters in distribution grids," in *2017 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, IEEE, 2017, pp. 1–6.

[142] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2015.

[143] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, no. 1, pp. 43–57, 1985.

[144] M. Choi, V. Tan, A. Anandkumar, and A. Willsky, "Learning latent tree graphical models," *Journal of Machine Learning Research*, May 2011.

[145] S. Park, D. Deka, and M. Chertkov, "Exact topology and parameter estimation in distribution grids with minimal observability," *2018 Power Systems Computation Conference (PSCC)*, Jun. 2018.

[146] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I*, vol. 60, pp. 150–163, 1 Jan. 2013.

[147] L. Lovasz, "Eigenvalues of graphs," 2007. [Online]. Available: `http://web.cs.elte.hu/~lovasz/eigenvals-x.pdf`.

[148] G. Kron, "Tensor analysis of networks," 1939.

[149] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, 1 Feb. 2011.

[150] J. F. Grcar, "Optimal sensitivity analysis of linear least squares.," *Lawrence Berkeley National Laboratory, Report*, 2003.

[151] D. Robinson and L. Foulds, "Comparison of phylogenetic trees," *Mathematical Biosciences*, vol. 53, pp. 131–147, 1-2 Feb. 1981.

[152] J. Verboomen, D. Van Hertem, P. H. Schavemaker, W. L. Kling, and R. Belmans, "Phase shifting transformers: Principles and applications," in *2005 International Conference on Future Power Systems*, IEEE, 2005, 6–pp.

[153] R. C. Dugan and T. E. McDermott, "An open source platform for collaborating on smart grid research," in *2011 IEEE Power and Energy Society General Meeting*, IEEE, 2011, pp. 1–7.

[154] J.-S. Li and X.-D. Zhang, "On the laplacian eigenvalues of a graph," *Linear algebra and its applications*, vol. 285, no. 1-3, pp. 305–307, 1998.

[155] F. Olivier, A. Sutera, P. Geurts, R. Fonteneau, and D. Ernst, "Phase identification of smart meters by clustering voltage measurements," in *2018 Power Systems Computation Conference (PSCC)*, IEEE, 2018, pp. 1–8.

[156] H. Pezeshki and P. J. Wolfs, "Consumer phase identification in a three phase unbalanced lv distribution network," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, IEEE, 2012, pp. 1–7.

[157] L. Blakely, M. J. Reno, and W.-c. Feng, "Spectral clustering for customer phase identification using ami voltage timeseries," in *2019 IEEE Power and Energy Conference at Illinois (PECI)*, IEEE, 2019, pp. 1–7.

[158] W. Wang, N. Yu, B. Foggo, J. Davis, and J. Li, "Phase identification in electric power distribution systems by clustering of smart meter data," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2016, pp. 259–265.

[159] T. A. Short, "Advanced metering for phase identification, transformer identification, and secondary modeling," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 651–658, 2012.

[160] K. Brady, "Uses and limitations of micro-synchrophasor measurements in distribution grid management," *PhD diss., Masters thesis, EECS Department, University of California, Berkeley*, 2016.

[161] W. H. Kersting, "Radial distribution test feeders," *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975–985, 1991.

[162] M. Escobar, D. Bienstock, and M. Chertkov, "Learning from power system data stream," in *2019 IEEE Milan PowerTech*, IEEE, 2019, pp. 1–6.

[163] F. R. Chung and F. C. Graham, *Spectral graph theory*, 92. American Mathematical Soc., 1997.

[164] J. A. Taylor, *Convex optimization of power systems*. Cambridge University Press, 2015.

[165] L. Liu, M. Esmalifalak, and Z. Han, "Detection of false data injection in power grid exploiting low rank and sparsity," in *2013 IEEE international conference on communications (ICC)*, IEEE, 2013, pp. 4461–4465.

[166] P. N. McGraw and M. Menzinger, "Laplacian spectra as a diagnostic tool for network structure and dynamics," *Physical Review E*, vol. 77, no. 3, p. 031 102, 2008.

[167] A. Mukherjee, R. Vallakati, V. Lachenaud, and P. Ranganathan, "Using phasor data for visualization and data mining in smart-grid applications," in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, IEEE, 2015, pp. 13–18.

[168] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[169] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[170] A. C. Adewole and R. Tzoneva, "Synchrophasor-based online coherency identification in voltage stability assessment," *Advances in Electrical and Computer Engineering*, 2015.

[171] A. Tascikaraoglu and B. M. Sanandaji, "Short-term residential electric load forecasting: A compressive spatio-temporal approach," *Energy and Buildings*, vol. 111, pp. 380–392, 2016.

[172] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[173] K. Sajan, M. Bariya, S. Basak, A. Srivastava, A. Dubey, A. von Meier, and G. Biswas, "Realistic synchrophasor data generation for anomaly detection and event classification," in *2020 8th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, IEEE, 2020, pp. 1–6.

[174] M. Bariya, J. Paparrizos, A. von Meier, and M. J. Franklin, "K - shapestream: Probablistic streaming clustering for electric grid events," in *2021 PowerTech*, IEEE, 2021.

[175] E. F. Fuchs and M. A. Masoum, "Power quality in power systems and electrical machines book• 2008," 2015.

[176] Y. Zhou, R. Arghandeh, and C. J. Spanos, "Partial knowledge data-driven event detection for power distribution networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5152–5162, 2017.

[177] S. Pandey, A. K. Srivastava, and B. G. Amidan, "A real time event detection, classification and localization using synchrophasor data," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4421–4431, 2020.

[178] H. Li, Y. Weng, E. Farantatos, and M. Patel, "An unsupervised learning framework for event detection, type identification and localization using pmus without any historical labels," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2019, pp. 1–5.

[179] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory protection and automated event diagnosis using pmu data," *IEEE Transactions on power delivery*, vol. 31, no. 4, pp. 1855–1863, 2016.

[180] A. Shahsavari, M. Farajollahi, E. M. Stewart, E. Cortez, and H. Mohsenian - Rad, "Situational awareness in distribution grid using micro-pmu data: A machine learning approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6167–6177, 2019.

[181] R. Yadav, S. Raj, and A. K. Pradhan, "Real-time event classification in power system with renewables using kernel density estimation and deep neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6849–6859, 2019.

[182] S. S. Negi, N. Kishor, K. Uhlen, and R. Negi, "Event detection and its signal characterization in pmu data stream," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3108–3118, 2017.

[183] M. Biswal, Y. Hao, P. Chen, S. Brahma, H. Cao, and P. De Leon, "Signal features for classification of power system disturbances using pmu data," in *2016 Power Systems Computation Conference (PSCC)*, IEEE, 2016, pp. 1–7.

[184] D. Phillips and T. Overbye, "Distribution system event detection and classification using local voltage measurements," in *2014 Power and Energy Conference at Illinois (PECI)*, IEEE, 2014, pp. 1–4.

[185] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," *def*, vol. 1, no. 2σ2, p. 16, 2007.

[186] I. Dabbagchi, "Ieee 14 bus power flow test case," *American Electric Power System, Golden CO*, 1962.

[187] S.-l. Yang, C. Shen, *et al.*, "A review of electric load classification in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 103–110, 2013.

[188] G. Fan, K. Shi, T. Zheng, L. Feng, and Z. Li, "Cluster analysis of grid-connected large scale wind farms," *Power System Technology*, vol. 11, pp. 62–66, 2011.

[189] A. Maurya, A. S. Akyurek, B. Aksanli, and T. S. Rosing, "Time-series clustering for data analysis in smart grid," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2016, pp. 606–611.

[190] A. Aligholian, A. Shahsavari, E. Stewart, E. Cortez, and H. Mohsenian - Rad, "Unsupervised event detection, clustering, and use case exposition in micro-pmu measurements," *arXiv preprint arXiv:2007.15237*, 2020.

[191] E. Klinginsmith, R. Barella, X. Zhao, and S. Wallace, "Unsupervised clustering on pmu data for event characterization on smart grid," in *2016 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, IEEE, 2016, pp. 1–8.

[192] J. Paparrizos and L. Gravano, "K-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1855–1870.

[193] ——, "Fast and accurate time-series clustering," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 2, pp. 1–49, 2017.

[194] S. N. Fallah, R. C. Deo, M. Shojafar, M. Conti, and S. Shamshirband, "Computational intelligence approaches for energy load forecasting in smart energy management grids: State of the art, future challenges, and research directions," *Energies*, vol. 11, no. 3, p. 596, 2018.

[195] J. Yang, C. Ning, C. Deb, F. Zhang, D. Cheong, S. E. Lee, C. Sekhar, and K. W. Tham, "K-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement," *Energy and Buildings*, vol. 146, pp. 27–37, 2017.

[196] F. Fahiman, S. M. Erfani, S. Rajasegarar, M. Palaniswami, and C. Leckie, "Improving load forecasting based on deep learning and k-shape clustering," in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 4134–4141.

[197] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[198] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[199] J. Paparrizos, C. Liu, A. J. Elmore, and M. J. Franklin, "Debunking four long-standing misconceptions of time-series distance measures," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1887–1905.

[200] D. Q. Goldin and P. C. Kanellakis, "On similarity queries for time-series data: Constraint specification and implementation," in *International Conference on Principles and Practice of Constraint Programming*, Springer, 1995, pp. 137–153.

[201] S. Murphy, K. Jones, T. Laughner, M. Bariya, and A. von Meier, "Accelerating artificial intelligence on the grid," in *2020 Clemson University Power Systems Conference (PSC)*, IEEE, 2020, pp. 1–7.

[202] M. H. Bollen, "Voltage sags: Effects, mitigation and prediction," *Power Engineering Journal*, vol. 10, no. 3, pp. 129–135, 1996.

[203] R. Jongen, P. Morshuis, J. Smit, A. Janssen, and E. Gulski, "A statistical approach to processing power transformer failure data," in *19th International Conference on Electricity Distribution*, 2007, p. 4.

[204] C. Roberts, A. Scaglione, M. Jamei, R. Gentz, S. Peisert, E. M. Stewart, C. McParland, A. McEachern, and D. Arnold, "Learning behavior of distribution system discrete control devices for cyber-physical security," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 749–761, 2019.

[205] J. V. Milanovic, M. T. Aung, and C. Gupta, "The influence of fault distribution on stochastic prediction of voltage sags," *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 278–285, 2005.

[206] N. Ball, L. Kiernan, K. Warwick, E. Cahill, D. Esp, and J. Macqueen, "Neural networks for power systems alarm handling," *Neurocomputing*, vol. 4, no. 1-2, pp. 5–8, 1992.

[207] (). "PSL PQube specifications," Power Standards Laboratory, [Online]. Available: `http://PQube3.com`.

[208] A. von Meier, R. Arghandeh, K. Brady, M. Brown, G. Cotter, D. Deka, H. Hooshyar, M. Jamei, H. Kirkham, A. McEachern, L. Mehrmanesh, T. Rizy, A. Scaglione, J. Schuman, Y. Seyedi, A. Shahvasari, A. Silverstein, E. Stewart, L. Vanfretti, and L. Zhan, "Synchrophasor monitoring for distribution systems: Technical foundations and applications, a white paper by the NASPI distribution task team," North American SynchroPhasor Initiative (NASPI), Report, Jan. 2018. [Online]. Available: `www.naspi.org/node/688`.

[209] H. Kirkham, "A conceptual framework for measurement (with emphasis on phasor measurement)," Pacific Northwest National Laboratory (PNNL), Report, 2015.

[210] ——, "Measurement of phasor-like quantities," in *DOE/OE Transmission Reliability Program, CERTS*, Washington, DC, 2015.

[211] A. Silverstein, M. Weimar, and J. Petersen, "The value proposition for synchrophasor technology: Itemizing and calculating the benefits from synchrophasor technology use, version 1.0," North American SynchroPhasor Initiative (NASPI), Report, Oct. 2015. [Online]. Available: `www.naspi.org/sites/default/files/reference_documents/5.pdf`.

[212] O. Ardakanian, Y. Yuan, R. Dobbe, A. von Meier, S. Low, and C. Tomlin, "Event detection and localization in distribution grids with phasor measurement units," *arXiv*, Nov. 2016.

[213] M. Bariya, A. von Meier, A. Ostfeld, and E. Ratnam, "Data-driven topology estimation with limited sensors in radial distribution feeders," *IEEE Green Technologies Conference (GreenTech), 2018*, pp. 183–188, Apr. 2018.

[214] A. Riepnieks and H. Kirkham, “An introduction to goodness of fit for pmu parameter estimation,” *IEEE Transactions on Power Delivery*, vol. 32, pp. 2238–2245, 5 Oct. 2017.

[215] H. Kirkham and A. Riepnieks, “Students’ simple method for the determining the parameters of an ac signal,” *2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, Oct. 2016.

[216] J. A. de la O Serna, “Dynamic phasor estimates for power system oscillations,” *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 1648–1657, 5 2007.

[217] B. M. Sanandaji, E. Bitar, K. Poolla, and T. L. Vincent, “An abrupt change detection heuristic with applications to cyber data attacks on power systems,” in *2014 American Control Conference*, IEEE, 2014, pp. 5056–5061.

[218] J. W. Mitchell, “Power lines and catastrophic wildland fire in southern california,” in *Proceedings of the 11th International Conference on Fire and Materials*, Citeseer, 2009, pp. 225–238.

[219] R. Arghandeh, M. Brown, A. Del Rosso, G. Ghatikar, E. Stewart, A. Vojdani, and A. von Meier, “The local team: Leveraging distributed resources to improve resilience,” *IEEE Power and Energy Magazine*, vol. 12, no. 5, pp. 76–83, 2014.

[220] A. Silverstein, “Naspi and synchrophasor technology progress,” in *NERC OC-PC Meetings*, 2013.

[221] T. Doggett, “Overcoming barriers to smart grids & new energy services,” in *Proceedings of UT Smart Grid Conference, The University of Texas, Austin, ERCOT*, vol. 7, 2011.

[222] I. Niazazari and H. Livani, “A pmu-data-driven disruptive event classification in distribution systems,” *Electric Power Systems Research*, vol. 157, pp. 251–260, 2018.

[223] Y. Cai and M.-Y. Chow, “Exploratory analysis of massive data for distribution fault diagnosis in smart grids,” in *2009 IEEE Power & Energy Society General Meeting*, IEEE, 2009, pp. 1–6.

[224] M. Mishra and P. K. Rout, "Detection and classification of micro-grid faults based on hht and machine learning techniques," *IET Generation, Transmission & Distribution*, vol. 12, no. 2, pp. 388–397, 2017.

[225] D. Deka, S. Talukdar, M. Chertkov, and M. V. Salapaka, "Graphical models in meshed distribution grids: Topology estimation, change detection & limitations," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4299–4310, 2020.

[226] S. C. Johnson, D. J. Papageorgiou, D. S. Mallapragada, T. A. Deetjen, J. D. Rhodes, and M. E. Webber, "Evaluating rotational inertia as a component of grid reliability with high penetrations of variable renewable energy," *Energy*, vol. 180, pp. 258–271, 2019.