Partial State Estimation for Electricity Grids

Manuel Garcia†, Annarita Giani∗§, Kameshwar Poolla†

Abstract— Power system operators rely critically on state estimation for verification, fault detection and localization, and re-dispatch under contingency operations. In current practice, power system data within a control area such as voltages, phases, real and reactive power flows and injections, are relayed to the operator using SCADA systems. State estimation is formulated as an over-determined weighted nonlinear least squares problem and the solver of choice is the Newton-Raphson method. Two critical issues are: (a) estimate quality, due to data latency or convergence to false local minima, and (b) computation time, due to the large number of state variables involved.

In this paper, we explore techniques to accelerate state estimation by computing state estimates at a small subset of buses using limited measurements from the power subsystem of interest. These could be operator selected “important” buses which connect to “important” lines with significant real power transfer. Our techniques are inspired by uncertainty quantification methods. The influence of power flows from exogenous buses is treated as uncertainty which defines a feasible set of state variables consistent with available measurements. The state estimation problem can be cast as a non-convex optimization problem. We use a surrogate model relaxation and Shor’s rank relaxation to obtain state estimates and associated error bounds at user-defined confidence levels.

I. INTRODUCTION

The state of a power system consists of voltage magnitudes and phases at all buses. Knowledge of the state allows computation of real/reactive power flows on any line and real/reactive power injections at any bus. As the state is not directly measured, it must be estimated from measurements of voltage, real/reactive power injections at various buses, and real/reactive power flows on various transmission lines. State estimation requires solving an over-determined set of coupled nonlinear algebraic equations. Power system operators rely critically on state estimation for fault detection and localization, re-dispatch under contingency operations, system visibility, and situation awareness.

The electricity grid is evolving rapidly. Renewable generation is being added rapidly and adds significant variability to supply. Demand is more elastic and flexible loads are under direct or price proxy control. Cyber-security concerns have emerged from the increased use of information technology in grid operations. Managing grid resources in this brave new world presents an enormous challenge. As a result, there is an urgency for innovation in state estimation techniques which have remain unchanged for decades. In response, there is a push to make critical state estimation tools open-source, which will allow for innovations necessary for tomorrow’s uncertain grid to proliferate.

Existing state estimation methods suffer from two principal drawbacks:

(a) Total computation time. Traditional tools are designed to estimate the state of the entire transmission network (> 10⁴ for the California grid). State estimators require at least as many measurements as the state dimension, to solve an over-determined set of coupled nonlinear equations. Estimation is gated both by acquisition of these measurements and by algorithmic computation time. As a result, it may take several minutes for estimates to become available.

(b) Convergence issues. On occasion, the iterative methods used for state estimation converge to false local minima or even fail to converge. This is due to the enormous state dimension, and the large latency in the measurement stream. Indeed, the state of the system may have changed by the time sufficient measurements have been acquired. This concern is compounded with the increased supply variability from renewables.

In this paper, we explore techniques to potentially address these shortcomings in legacy state estimation. Our methods are inspired by uncertainty quantification research, and offer the promise of reduced computation time and improved convergence. In particular, we are concerned with the partial state estimation problem, where we wish to compute state estimates at a subset of buses using limited measurements from the power subsystem of interest. This problem is motivated by the fact that system operators are most often interested in state estimates at “important” buses. States at these select buses allow vigilant monitoring of significant power flows.

For partial state estimation problems, we treat power flows from exogenous buses as random variables and rely critically on a priori information of voltages and power flows within the subsystem of interest. This formulation naturally lends itself to uncertainty quantification which lets us provide error bounds on the estimated state vector at operator-defined confidence levels. These bounds are found by solving a series...
of convex optimization problems that can be solved quickly and have convergence guarantees. Our speed advantages are derived from the fact that we are often dealing with a small sub-grid of important buses. We emphasize that a consequence of using a subset of available measurements is that the uncertainty in state estimates increases. We illustrate this trade-off through synthetic examples.

II. BACKGROUND AND RELATED WORK

A. State Estimation Background

State estimation was first proposed by Schweppe 40 years ago in his seminal paper [12]. Today, it is a standard and essential tool for power systems operations. Monticelli offers a comprehensive survey of state estimation [2].

State estimation is applied to an observable subsystem (usually an entire control area). Before conventional state estimation is performed network topology checkers verify assumed logical data such as relay states and bad data detection subroutines pre-screen measurements for outliers.

We briefly describe the essential computational elements of state estimation that we will require for our exposition. Consider a power system with \( n \) buses in steady-state operation. The state of this system consists of the voltage magnitudes \( V_k \) and voltage angles \( \theta_k \) at each bus. A slack bus is designated to reference voltage angles. With this caveat, we write the state as

\[
x = \begin{bmatrix} V \\ \theta \end{bmatrix} \in \mathbb{R}^{2n}
\]

To estimate the state, we have available certain noisy measurements. These include voltages (magnitude and possibly phase), real/reactive power injections/extractions at select buses and real/reactive power flows on select lines. These measurements are acquired at various sampling rates and transmitted asynchronously to the EMS control center over a legacy SCADA network. We can gather all available measurements and write

\[
z = h(x) + e
\]

where \( z \) is the vector of measurements, \( h \) is a vector-valued nonlinear function of the state, and \( e \) represents measurement noise. The \( k^{th} \) components of the vector \( z \) and the function \( h(\cdot) \) are written \( z_k \) and \( h_k(\cdot) \). We assume the noise is zero-mean Gaussian, with covariance matrix \( \Sigma \). The standard state estimation problem is then

\[
\min_x \| \Sigma^{-\frac{1}{2}} (z - h(x)) \|_2^2 \quad \text{st:} \quad x \in \mathcal{P}
\]

where \( \mathcal{P} \) (defined formally in the next section) represents the prior information constraints such as direct state constraints and line capacities. This is a weighted nonlinear least-squares problem which is traditionally solved with the Newton-Raphson algorithm. The nonlinearities \( h \) cause this problem to be challenging, and state estimates may converge to false local minima, or even fail to converge.

B. Related Work on Fast State Estimation

Fast and partial state estimation has received some recent attention. A comprehensive treatment of uncertainty analysis in state estimation using sequential quadratic programming can be found in [6]. In some earlier work [3], the same authors use weighted least squares to compute expected values of the state variables as initial seeds, followed by a linear programming formulation to find the tight upper and lower error bounds. The use of fuzzy linear regression models, for modeling uncertainty in power system state estimation may be found in [4]. Fuzzy logic-based state estimation has also been proposed in [7].

The use of Phasor Measurements Unit (PMU) data for state estimation has received considerable attention. For example [8] incorporate PMU data in conventional state estimators [8]. A placement algorithm for optimally informative PMUs is proposed in [11]. This algorithm determine PMU placement using system topological structure, accuracy and redundancy.

III. FEASIBLE SET FOR THE STATE VECTOR

We will assume noise cross-covariances are zero, so we can write \( \Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_2^2) \). Define the feasible set \( \mathcal{F} \) to be

\[
(x, e) \in \mathcal{P}
\]

\[
z_i = h_i(x) + e_i, \quad i = [1, \ldots, n]
\]

\[
|e_i| \leq n_i \sigma_i, \quad i = [1, \ldots, n]
\]

\[
V_i^{\text{min}} \leq x_i \leq V_i^{\text{max}}, \quad i = [1, \ldots, n]
\]

\[
\delta_{ij} \leq x_i - x_j \leq \delta_{ij}^\text{max}, \quad \forall (i, j) \in \mathcal{L}
\]

\[
x \in \mathcal{C}
\]

The feasible set confines the state and the measurement noise based on available measurements \( z \), operator specified confidence bounds on \( e \), and prior state information. The equality constraints (2) correspond to the \( a \) available measurements, while constraints (3) capture the confidence bounds. Constraints (4) are simple bounding boxes for the voltage magnitudes while constraints (5) are bounding polytopes for the voltage angles. \( \mathcal{L} \) is the set of pairs \((i, j)\) such that a line exists connecting buses \( i \) and \( j \). \( V_i^{\text{min}}, V_i^{\text{max}}, \delta_{ij}^\text{min}, \) and \( \delta_{ij}^\text{max} \) are bounds chosen by the system operator \textit{a priori}. Any physical constraint known by the system operator (i.e. power flow constraints) can be encoded by (6). For simplicity we will ignore these physical constraints and omit constraint (6) in future sections.

Define the set \( \mathcal{M} \) to be the set of all \((x, e)\) that satisfy the measurement constraints (2) and (3). \( \mathcal{D} \) will be the set of all \( x \) that satisfy constraints (4) and (5). \( \mathcal{D} \) forms a polytope to which the state is \textit{a priori} confined. The prior information constraints are (4), (5), and (6), and we have \( \mathcal{P} = \mathcal{C} \cap \mathcal{D} \).

Define the feasible set of states \( \mathcal{X} \) to be the projection of the feasible set \( \mathcal{F} \) onto state space. Equivalently,

\[
\mathcal{X} = \{x : \exists e \text{ where } (x, e) \in \mathcal{F}\}
\]
1) System Operator Defined Confidence Levels: The values $n_i$ in constraints 3 can be adjusted by the system operator. He/she can choose $n_i$ so that

$$\prod_{i=1}^{n} \text{Prob} \left( |c_i| \leq n_i \sigma_i \right) = p$$

Assuming that constraints (2), (4), (5), and (6) all hold with certainty, the feasible set $F$ will contain the true state with confidence level $p$.

2) Model Functions $h_i(\cdot)$: Some components of $z$ could be direct state measurements (voltage magnitude or phase) for which the corresponding functions $h_i(\cdot)$ are trivial. Other components of $z$ could be real/reactive power injections at buses, or real/reactive power flows on lines. In this case we use the two-port $\pi$-model of a network from [17].

The standard formulae are:

Real and reactive power flow from bus $i$ to bus $j$:

$$h_i^P(x) = V_i^2(g_{si} + g_{ij}) - V_i V_j (g_{ij} \cos \delta_{ij} + b_{ij} \sin \delta_{ij})$$
$$h_i^Q(x) = -V_i^2(b_{si} + b_{ij}) - V_i V_j (g_{ij} \sin \delta_{ij} - b_{ij} \cos \delta_{ij})$$

Real and Reactive power injections at each bus:

$$h_i(x) = \sum_{j \in \mathcal{L}_i} h_{ij}^P(x), \quad h_i^S(x) = \sum_{j \in \mathcal{L}_i} h_{ij}^Q(x)$$

where $g_{ij} + jb_{ij}$ is the admittance of the series branch connecting buses $i$ and $j$, and $g_{si} + jb_{si}$ is the admittance of the shunt branch connected at bus $i$. $V_i$ is the voltage magnitude at bus $i$, and $\delta_{ij} = \theta_i - \theta_j$ is the phase difference between bus $i$ and bus $j$. The set $\mathcal{L}_i$ is the set of all buses that share a transmission line with bus $i$.

The following result allows us to better understand the feasible set $F$:

**Theorem 1:** If $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ is diagonal and $\mathcal{P}$ is non-empty, then the WLS problem

$$\min_x \| \Sigma^{-\frac{1}{2}} (z - h(x)) \|_2 \quad \text{st: } x \in \mathcal{P} \quad (8)$$

is equivalent to

$$\min_{x, n, e} \| n \|_2 \quad \text{s.t. } \left\{ \begin{array}{l}
|e_i| \leq \sigma_i n_i, \quad i = [1, \ldots, a] \\
x \in \mathcal{P}
\end{array} \right. \quad (9)$$

The decision variables are $x \in \mathbb{R}^N$, $e \in \mathbb{R}^a$, $n \in \mathbb{R}^a$. The optimal values are $n^*$, $x^*$ and $e^*$.

**Proof:** [21] provides a proof.

This result offers two important insights into the feasible set $F$. First, it shows that for over-determined systems, there is a lower bound on the two-norm of $n$. This implies that the system operator must be careful when choosing $n$ in defining the feasible set $F$. If he/she chooses $n$ such that $\|n\|_2 < \|n^*\|_2$ then the resulting feasible set, $F$, will be empty. Second, if the user chooses $n$ such that $n_i \geq n_i^* \forall i \in [1, \ldots, a]$, then the feasible set $F$ will contain the WLS optimal state estimate.

IV. MAIN RESULTS

A. Problem Formulation

We now formulate the state bounding problem which finds the smallest volume, normal hyper-rectangle that contains the feasible set of states $\mathcal{X}$. This hyper-rectangle is illustrated in Figure 1. The solution to this problem is termed the true bound of the state. Since the hyper-rectangle formed by the true bounds contains the feasible set $\mathcal{X}$, the true bounds must hold with a minimum confidence level of $p$.

The true bounds are found by solving two optimization problems for each state variable $x_i$. The true lower and true upper bounds on the $i$th state variable are denoted $\underline{x}_i$ and $\bar{x}_i$ respectively.

$$\underline{x}_i = \min_x x_i \quad \text{st: } x \in \mathcal{X} \quad (10)$$
$$\bar{x}_i = \max_x x_i \quad \text{st: } x \in \mathcal{X} \quad (11)$$

The problems (10) and (11) are difficult to solve because the constraints $x \in \mathcal{X}$ are non-convex. An approximate solution can be found using a Newton-Raphson algorithm; however, this algorithm may converge to a false local minimum and the resulting bounds are not guaranteed to contain the feasible set $\mathcal{X}$. This inner bounding hyper-rectangle is illustrated in Figure 1.

We now describe two relaxations of (10) and (11) that result in convex optimization problems. Solving the relaxed problems allows us to compute outer bounds on the feasible set as illustrated in Figure 1.

IV. MAIN RESULTS

A. Problem Formulation

We now formulate the state bounding problem which finds the smallest volume, normal hyper-rectangle that contains the feasible set of states $\mathcal{X}$. This hyper-rectangle is illustrated in Figure 1. The solution to this problem is termed the true bound of the state. Since the hyper-rectangle formed by the true bounds contains the feasible set $\mathcal{X}$, the true bounds must hold with a minimum confidence level of $p$.

The true bounds are found by solving two optimization problems for each state variable $x_i$. The true lower and true upper bounds on the $i$th state variable are denoted $\underline{x}_i$ and $\bar{x}_i$ respectively.

$$\underline{x}_i = \min_x x_i \quad \text{st: } x \in \mathcal{X} \quad (10)$$
$$\bar{x}_i = \max_x x_i \quad \text{st: } x \in \mathcal{X} \quad (11)$$

The problems (10) and (11) are difficult to solve because the constraints $x \in \mathcal{X}$ are non-convex. An approximate solution can be found using a Newton-Raphson algorithm; however, this algorithm may converge to a false local minimum and the resulting bounds are not guaranteed to contain the feasible set $\mathcal{X}$. This inner bounding hyper-rectangle is illustrated in Figure 1.

We now describe two relaxations of (10) and (11) that result in convex optimization problems. Solving the relaxed problems allows us to compute outer bounds on the feasible set as illustrated in Figure 1.

IV. MAIN RESULTS

A. Problem Formulation

We now formulate the state bounding problem which finds the smallest volume, normal hyper-rectangle that contains the feasible set of states $\mathcal{X}$. This hyper-rectangle is illustrated in Figure 1. The solution to this problem is termed the true bound of the state. Since the hyper-rectangle formed by the true bounds contains the feasible set $\mathcal{X}$, the true bounds must hold with a minimum confidence level of $p$.

The true bounds are found by solving two optimization problems for each state variable $x_i$. The true lower and true upper bounds on the $i$th state variable are denoted $\underline{x}_i$ and $\bar{x}_i$ respectively.

$$\underline{x}_i = \min_x x_i \quad \text{st: } x \in \mathcal{X} \quad (10)$$
$$\bar{x}_i = \max_x x_i \quad \text{st: } x \in \mathcal{X} \quad (11)$$

The problems (10) and (11) are difficult to solve because the constraints $x \in \mathcal{X}$ are non-convex. An approximate solution can be found using a Newton-Raphson algorithm; however, this algorithm may converge to a false local minimum and the resulting bounds are not guaranteed to contain the feasible set $\mathcal{X}$. This inner bounding hyper-rectangle is illustrated in Figure 1.

We now describe two relaxations of (10) and (11) that result in convex optimization problems. Solving the relaxed problems allows us to compute outer bounds on the feasible set as illustrated in Figure 1.

IV. MAIN RESULTS

A. Problem Formulation

We now formulate the state bounding problem which finds the smallest volume, normal hyper-rectangle that contains the feasible set of states $\mathcal{X}$. This hyper-rectangle is illustrated in Figure 1. The solution to this problem is termed the true bound of the state. Since the hyper-rectangle formed by the true bounds contains the feasible set $\mathcal{X}$, the true bounds must hold with a minimum confidence level of $p$.

The true bounds are found by solving two optimization problems for each state variable $x_i$. The true lower and true upper bounds on the $i$th state variable are denoted $\underline{x}_i$ and $\bar{x}_i$ respectively.

$$\underline{x}_i = \min_x x_i \quad \text{st: } x \in \mathcal{X} \quad (10)$$
$$\bar{x}_i = \max_x x_i \quad \text{st: } x \in \mathcal{X} \quad (11)$$

The problems (10) and (11) are difficult to solve because the constraints $x \in \mathcal{X}$ are non-convex. An approximate solution can be found using a Newton-Raphson algorithm; however, this algorithm may converge to a false local minimum and the resulting bounds are not guaranteed to contain the feasible set $\mathcal{X}$. This inner bounding hyper-rectangle is illustrated in Figure 1.

We now describe two relaxations of (10) and (11) that result in convex optimization problems. Solving the relaxed problems allows us to compute outer bounds on the feasible set as illustrated in Figure 1.

IV. MAIN RESULTS

A. Problem Formulation

We now formulate the state bounding problem which finds the smallest volume, normal hyper-rectangle that contains the feasible set of states $\mathcal{X}$. This hyper-rectangle is illustrated in Figure 1. The solution to this problem is termed the true bound of the state. Since the hyper-rectangle formed by the true bounds contains the feasible set $\mathcal{X}$, the true bounds must hold with a minimum confidence level of $p$.

The true bounds are found by solving two optimization problems for each state variable $x_i$. The true lower and true upper bounds on the $i$th state variable are denoted $\underline{x}_i$ and $\bar{x}_i$ respectively.

$$\underline{x}_i = \min_x x_i \quad \text{st: } x \in \mathcal{X} \quad (10)$$
$$\bar{x}_i = \max_x x_i \quad \text{st: } x \in \mathcal{X} \quad (11)$$

The problems (10) and (11) are difficult to solve because the constraints $x \in \mathcal{X}$ are non-convex. An approximate solution can be found using a Newton-Raphson algorithm; however, this algorithm may converge to a false local minimum and the resulting bounds are not guaranteed to contain the feasible set $\mathcal{X}$. This inner bounding hyper-rectangle is illustrated in Figure 1.
where $\epsilon_h(x)$ is the error of the approximation. Define the maximum absolute error

$$\epsilon_{h_{\text{absmax}}} = \max_{x \in \mathcal{D}} |\epsilon_h(x)|$$

[13] offers a method for determining an (empirical) two-norm quadratic surrogate model. This method minimizes the two-norm of the error of the surrogate function evaluated at points that are randomly sampled from some domain. It is more convenient to use $\mathcal{D}$ than $\mathcal{X}$ for this purpose. Indeed generating uniformly distributed samples from an irregular domain is a very difficult problem. The matrix $N_h$ is indefinite. The maximum absolute error $\epsilon_{h_{\text{absmax}}}$ is determined by evaluating the difference between the model function $h_i(x)$ and the surrogate function $x^T N_h x$ at many randomly sampled points in $\mathcal{D}$.

To obtain a quadratic programming formulation the surrogate relaxation transforms the set $\mathcal{F}$ into $\mathcal{S}$ which is defined as:

$$\{x, e\} \text{ st: } z_i \leq x^T N_h x + e_i + \epsilon_{h_{\text{absmax}}}, \quad i \in \{1, \ldots, a\}$$

$$z_i \geq x^T N_h x + e_i - \epsilon_{h_{\text{absmax}}}, \quad i \in \{1, \ldots, a\}$$

$$- \sigma_i n_i \leq e_i \leq \sigma_i n_i, \quad i \in \{1, \ldots, a\}$$

$$V_i^{\text{min}} \leq x_i \leq V_i^{\text{max}}, \quad i \in \{1, \ldots, n\}$$

$$\delta_{ij}^{\text{min}} \leq x_{n+i} - x_{n+j} \leq \delta_{ij}^{\text{max}}, \quad \forall (i, j) \in \mathcal{L}$$

Simple algebraic manipulation will show that the set $\mathcal{S}$ can be written using quadratic constraints. As a result, the following relaxed problems are Non-convex Quadratically Constrained Quadratic Programs (NQCQPs).

$$\min x_i \text{ st: } (x, e) \in \mathcal{S}, \quad \max x_i \text{ st: } (x, e) \in \mathcal{S}$$

but the feasible set $\mathcal{S}$ is still non convex.

### C. Shor’s Rank Relaxation

Shor’s rank relaxation [15] [16] allows us to convexify Non-convex Quadratically Constrained (NQC) sets. Our relaxed set $\mathcal{S}$ can be written generally as

$$w \text{ st: } \begin{bmatrix} 1 \\ w \end{bmatrix}^T Z_i \begin{bmatrix} 1 \\ w \end{bmatrix} \leq 0 \quad \forall i$$

where $w = [x^T e^T]^T$.

By taking the trace of each inequality this set can be rewritten as follows. The formal proof is provided in [14].

$$w \text{ such that } \exists M \in \mathbb{R}^{n+m+a} \text{ where: }$$

$$\text{Tr} \begin{bmatrix} Z_i & w^T \\ w & M \end{bmatrix} \leq 0 \quad \forall i$$

$$\begin{bmatrix} 1 & w^T \\ w & M \end{bmatrix} \succ 0$$

$$\text{rank} \begin{bmatrix} 1 & w^T \\ w & M \end{bmatrix} = 1$$

Shor’s rank relaxation is performed by removing the rank constraint. The resulting rank relaxed set is convex and will be denoted $\mathcal{R}$. The following relaxed optimization problems are simple Semi-Definite Programs (SDPs). Outer bounds can be found easily by solving these problems. $\mathcal{R}_i$ and $\mathcal{R}_i^o$ are the outer lower and outer upper bounds respectively.

$$\mathcal{R}_i = \min x_i \text{ st: } (x, e) \in \mathcal{R}$$

$$\mathcal{R}_i^o = \max x_i \text{ st: } (x, e) \in \mathcal{R}$$

### D. Partial State Estimation

In practice the system operator is often interested in partial state estimation. Here, there is a small but critical portion of the grid that requires vigilant monitoring. We would like to perform frequent uncertainty quantification analysis of this subgrid. Partial state estimation accelerates our computational effort in two ways. First, less measurements are used and thus data latency becomes less of an issue. Second, it allows for the elimination of decision variables in SDP’s (21) and (22).

By bounding all state variables using the polytope constraint $x \in \mathcal{D}$, the feasible set $\mathcal{X}$ is bounded. As a result each SDP (21) and (22) is feasible and each state can be bounded. Further, when solving the SDPs for state variable $x_i$, state variable $x_j$ need not appear in the polytope constraint if it does not appear in the measurement constraints (2) and (3). In fact all state variables $x_i$ that satisfy this condition can be eliminated from SDPs (21) and (22).

In Section V-B, we offer an example that selectively chooses measurements to eliminate hundreds of decision variables and isolates a small portion of the grid.

### E. Iterative Algorithm

The iterative algorithm addresses the problem of choosing constants $V_i^{\text{min}}$, $V_i^{\text{max}}$, $\delta_{ij}^{\text{min}}$, and $\delta_{ij}^{\text{max}}$ in the feasible set definition. We assume the system operator contains reasonable yet generous estimates of the hard bounds described by constraints (4) and (5). The objective of the iterative algorithm is to use this prior knowledge of the system to attain even tighter bounds that will replace (4) and (5).

During each iteration, the algorithm formulates and solves SDPs (21) and (22) for all voltage magnitudes. Similarly, bounds are calculated for the voltage angle differences across each line. This can be done by solving an SDP with feasible set $\mathcal{R}$ and linear objective function $x_j - x_i$. The resulting bounds are then used as constraints (4) and (5) in the next iteration.

Each iteration can be viewed as a mapping $f(\cdot)$ from a polytope in state space to a smaller polytope. This mapping fits surrogate models on a domain specified by its argument (as explained in section IV-B), uses these surrogate models to formulate SDPs, and then solves the SDPs to compute
the new bounding polytope. This procedure is executed iteratively, using the resulting polytope as the bounds on the state in the second iteration (i.e., \( f(D) \subseteq f(D) \)). With successive iterations, surrogate models will be fit on smaller domains improving the surrogate relaxation. In practice we can continue this iterative process until the change in volume of the polytope is less than some tolerance.

This iterative process is shown in Figure 2. We define the initial polytope (the prior knowledge of the system operator) to be \( D_0 = D \). We iteratively calculate \( D_{i+1} = f(D_i) \). We would expect \( D_i \) to converge to some polytope, however, this is difficult to prove because the fitting technique involves random sampling of the surrogate fitting domain. We continue this iterative process until the change in volume of the polytope is less than some tolerance.

\[
\text{Given state constraints } D \text{ from the system operator}
\]

\begin{verbatim}
0 Define \( D_0 = D \)
1 \( D_1 = f(D_0) \)
2 set \( i = 0 \)

while volume(\( D_i \)) - volume(\( D_{i+1} \)) \( \geq \) tolerance

3 set \( i = i + 1 \)
4 \( D_{i+1} = f(D_i) \)
end
\end{verbatim}

Fig. 2: Iterative bounding algorithm

V. Empirical Results

A. Iterative Algorithm Example

We implemented our methodology on the IEEE 9-Bus test case shown in Figure 3. The “true” state was calculated using the basic power flow function of the Matpower Toolbox and lies in the interval

\[
V_i \in [1.01, 1.07] \quad \forall i, \quad \delta_j \in [-0.05, 0.16] \quad \forall j
\]

The initial constraints on the state variables (constraints (4) and (5) from Section III) are

\[
V_i \in [0.85, 1.15] \quad \forall i, \quad \delta_j \in [-0.3, 0.3] \quad \forall j
\]

Independent measurement noise was synthetically added at a standard deviation of \( \sigma = .01pu \). The measurements consist of real power injections and voltage magnitude at generator buses (PV buses), real and reactive power injections at load buses (PQ buses), real and reactive power flows at one end of each line. Real and reactive power injections at null buses are zero and are treated as equality constraints. For convenience we do not utilize the power injection measurement at bus 1, this is further explained in section V-B.

We ran simulations that enforced (a) \( 4\sigma \), and (b) \( 3\sigma \) limits. \( 4\sigma \) limits resulted in a feasible set \( F \) that holds with 99.82% confidence. \( 3\sigma \) limits resulted in a feasible set \( F \) that holds with 92.46% confidence. 20 iterations of the algorithm were performed in both cases. The results are shown in Figure 4 and Figure 5.

Figure 4 shows the polytope bounds that result from the final iteration of the algorithm. The upper plot shows the bounds on voltage magnitudes at each bus, while the lower plot shows the bounds on the voltage angle difference across each line. The bounds that resulted from \( 3\sigma \) (\( 4\sigma \)) error constraints are shown in gray (black). The true values are shown as a dot, which should lie within the error bounds. Notice that \( 3\sigma \) error constraints result in tighter bounds as compared to \( 4\sigma \) constraints. This makes sense intuitively because the feasible set \( F \) is smaller in this case.

Figure 5 shows the volume of the polytope at each iteration of the algorithm. The volume drops much more rapidly when using \( 3\sigma \) bounds, and it also converges to a smaller value. Using CVX software on a standard dual-core desktop computer, each SDP required an average of 3.41 seconds to solve. Each iteration solves 36 SDPs. Solving each of these SDPs in series for all 20 iterations required 40 minutes 55 seconds. However, parallelizing these computations could reduce the time to roughly 68 seconds on 40 processors, or 272 seconds on 10 processors.

B. Partial State Estimation

Our method allows isolation of a small portion of the grid. For example, consider again the IEEE 9 bus test case in Figure 3. Imagine bus 1 being connected by a tie-line to another 1000 bus grid. Since the power injection at bus 1 is unmeasured, the constraints that define our feasible set \( F \) from the previous example remain unchanged. We can eliminate all optimization decision variables corresponding to the states in the 1000 bus grid. This will result in significant computational savings.
surrogate model error [21]. We will also consider branch and bound methods [13, 14] to tighten error bounds. Using rectangular coordinates for state representation allows direct formulation of a NQCQP without using the surrogate model relaxation; however, it might be difficult to find appropriate initial bounds in this case.

REFERENCES


VI. CONCLUSIONS AND FUTURE WORK

We have offered a state estimation technique that exploits prior information, can be used for partial state estimation, and provides error-bounds at user-specified confidence levels. Our method has the potential of accelerating (or at least augmenting) legacy state estimation tools. We suggest our method be used in parallel to conventional point state estimation tools, augmenting) legacy state estimation tools. We suggest our method be used in parallel to conventional point state estimation tools. We suggest our method be used in parallel to conventional point state estimation tools. We suggest our method be used in parallel to conventional point state estimation tools.

Our results represent an initial attempt. We will continue this research using rational quadratic surrogate models to reduce