Least Absolute Value State Estimation

Student
Ashwin Venkatraman

Supervisors
Dmitry Shchetinin

Examiner
Prof. Dr. Gabriela Hug

Project Number
1815

Report Date
June 30, 2018

Power Systems Laboratory
ETH Zurich
Abstract

The most common methods used to solve the state estimation problem are the Weighted Least Squares (WLS) and Least Absolute Value (LAV). WLS is computationally efficient but it is not robust to outliers. On the other hand, LAV is robust to outliers but it is computationally demanding. This project presents a new LAV-based algorithm that is robust and fast. The LAV problem is formulated as an unconstrained non-linear optimization problem that can be solved using gradient-based approaches. The motivation is to combine the robustness of LAV with the computational efficiency of WLS. The proposed algorithm is then compared with the traditional WLS and LAV algorithms on the basis of computational time and robustness. The algorithms are tested using 30 to 13000 bus power systems.
Contents

Abstract ii
List of Acronyms iv
List of Symbols v
1 Introduction 1

2 Overview of State Estimation Formulations 4
  2.1 Introduction ..................................................... 4
  2.2 WLS ............................................................. 5
    2.2.1 Gauss Newton solution algorithm for WLS ............... 5
    2.2.2 Bad Data Processing ........................................ 6
  2.3 LAV ............................................................... 8
    2.3.1 Leverage Measurements ............................... 10

3 Proposed Algorithm 12
  3.1 Mathematical Formulation .................................... 12
  3.2 Algorithm ..................................................... 15

4 Experimental Results 17
  4.1 Simulation Setup .............................................. 17
  4.2 Without Bad Measurements .................................... 18
  4.3 With Bad Measurements ....................................... 19
    4.3.1 IP-LAV vs Proposed LAV .................................. 19
    4.3.2 Test system: 1888 rte ..................................... 22
    4.3.3 Test system: 1354pegase ................................. 29
    4.3.4 Comparison of Computational time ....................... 35

5 Conclusion 37

Bibliography 38
List of Acronyms

SE  State Estimation
SCADA  Supervisory Control and Data Acquisition
PMU  Phasor Measurement Unit
WLS  Weighted Least Squares
LAV  Least Absolute Value
LP  Linear Programming
IP  Interior Point
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>State vector</td>
</tr>
<tr>
<td>$z$</td>
<td>Measurement vector</td>
</tr>
<tr>
<td>$\hat{z}$</td>
<td>Estimated measurement vector</td>
</tr>
<tr>
<td>$e$</td>
<td>Measurement noise vector</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>Measurement function</td>
</tr>
<tr>
<td>$J(\cdot)$</td>
<td>Objective function</td>
</tr>
<tr>
<td>$W$</td>
<td>Weight Matrix</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
<td>Measurement Jacobian matrix</td>
</tr>
<tr>
<td>$g(\cdot)$</td>
<td>Gradient of Objective function</td>
</tr>
<tr>
<td>$G(\cdot)$</td>
<td>Gain matrix</td>
</tr>
<tr>
<td>$r$</td>
<td>Measurement residual</td>
</tr>
<tr>
<td>$r^N$</td>
<td>Normalized measurement residual</td>
</tr>
<tr>
<td>$R$</td>
<td>Measurement error covariance matrix</td>
</tr>
<tr>
<td>$K$</td>
<td>Hat matrix</td>
</tr>
<tr>
<td>$S$</td>
<td>Residual sensitivity matrix</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Residual covariance matrix</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Step size</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Modern power systems are highly complex and interconnected. The operating state of a given system needs to be constantly monitored, which is essential to ensure secure operation of the system. However, this is not an easy task due to the high complexity of the network. The monitoring is done with the help of a large number of measurements that are acquired by the Supervisory Control and Data Acquisition (SCADA) system. Traditionally, these measurements include voltage magnitudes at buses, active and reactive line power flows, line current flows and active and reactive power injection at buses. A reliable estimate of the system state is required for a lot of applications including Optimal Power Flow, contingency analysis and redispatch. The acquired measurements are used to determine the operating state of the system, which can be represented by the voltage magnitude and the voltage angle at every bus. This may seem like a straightforward task but the measurements available from the SCADA system pose the following challenges:

- It may not be possible to obtain the states directly from the available measurements. For example, usually, the voltage phasor measurements are not available.
- The measurements may be corrupted with noise or in some cases, may include gross errors.

This can be due to noises occurring during the communication phase, faulty meters etc. It is also possible that since the power system is dynamic, the measurement taken at a particular time is no longer useful because, in the meantime, the state of the system has changed.

In 1970, these problems were resolved when Fred C. Schweppe and J. Wildes published a two-part paper [1, 2] proposing the idea of state estimation. According to them, static state estimator is a data processing algorithm that uses the redundant measurements and other relevant information to arrive at an estimate of the system state vector. Typically a large number of measurements are available. The idea of state estimation is then to use the redundancy in the measurements to filter the errors present in the measurements and provide a reasonable estimate of the system state.

Apart from providing a reliable estimate of the system state, a state estimation program may also include other features such as observability analysis, topology processor, and post-processor for detecting errors in measurements (bad data), parameters of network elements and, network topology [3]. Observability analysis is an important component of a state estimator because only if the system is observable, i.e. enough measurements are available, the state estimation algorithm can work. An additional feature of the state estimator
can be to split the network into observable and unobservable modes and to identify the sources of the “unobservability” such as unobservable branches or islands. A key function of a state estimator is the ability to arrive at a reasonable estimate of the state even in the presence of errors in its input parameters. These errors can be broadly categorized as:

- **Measurement errors**: Gross errors (bad data) can be present in measurements due to various reasons including poor condition of measurement devices and human error. If there is enough redundancy in the system, the bad data processor can eliminate or even correct these measurements.

- **Parameter errors**: There can be errors in the values of network parameters such as line resistances and reactances used in the state estimation algorithm due to factors such as temperature. Correcting this type of errors is more difficult compared to correcting errors in measurements.

- **Topology errors**: Errors can also be present in the topology of the system. For example, a circuit breaker that is off can be wrongly categorized as on.

A reliable state estimator must make sure that these errors do not affect the solution of the state estimation algorithm.

In general, state estimation is a highly non-linear optimization problem. Hence, most of the solution methods available for solving the state estimation problem are iterative. However, in the last few years, there has been an increased use of Phasor Measurement Units (PMU) based measurements. PMUs provide measurements of bus voltages and line currents and it can be shown that the state estimation becomes linear in this case and hence, a direct solution can be obtained \cite{4}. However, PMUs are not yet ubiquitous and only a fraction of collected measurements are obtained with these devices. Therefore, in this project, PMU based measurements were not used and the state estimation problem remains non-linear.

The most common solution approaches used for solving the state estimation problem are the Weighted Least Squares (WLS) and Least Absolute Value (LAV). The WLS algorithm iteratively computes the state that minimizes the square of difference between the actual measurements and the estimated measurements. It is widely used prominently due to its high computational efficiency. However, the major drawback is that it is not robust to outliers. The estimates provided by WLS can be significantly affected by the presence of gross errors in measurements. In order to correct the estimates, post-processing needs to be done, which is computationally demanding.

The shortcomings of the WLS algorithm called for more robust estimators such as LAV. The LAV approach iteratively computes the state that minimizes the $L_1$ norm of the difference between the actual measurements and the estimated measurements. It is typically solved using standard solution techniques such as Interior Point and Simplex algorithms. The attractive property of this approach is its inherent ability to discard gross errors present in the measurements. On the other hand, it is much more computationally demanding compared to WLS. Moreover, this approach is vulnerable when there are errors present in “leverage” measurements as will be explained in Chapter 2. In this report, a new solution algorithm based on the LAV method is proposed. This algorithm models the LAV based state estimation as a smooth non-linear optimization problem. The main purpose of doing this is to be able to use gradient-based methods to solve the problem. This algorithm aims to combine the inherent ability of LAV to discard gross measurement errors and the computational efficiency of WLS. The proposed algorithm is compared...
with the traditional WLS and LAV algorithms on the basis of computational time and robustness.

The report is organized as follows. Chapter 2 presents the mathematical formulation and solution algorithm of the WLS and LAV methods for solving the state estimation problem. Chapter 3 presents the formulation of the proposed algorithm and Chapter 4 contains the simulation results of all presented approaches.
Chapter 2

Overview of State Estimation Formulations

In this chapter, the WLS and LAV solution approaches are discussed in detail. The mathematical formulation, solution methodology, and the advantages and disadvantages of both the approaches are presented.

2.1 Introduction

Consider a power system that contains \( n \) buses and \( m \) measurements. The measurements can be corrupted with noise and it is assumed that the measurement noise has a Gaussian distribution. Let \( h(\cdot) \) be the non-linear measurement function that relates the state vector \( x \) to the measurement vector \( z \). If \( e \) represents the measurement error vector,

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

(2.1)

It is important to note that the expected value of the measurement noise is assumed to be 0 and the measurement noises are assumed to be independent.

The state estimation problem can then be formulated as the following optimization problem:

\[
\min_x J(z - h(x))
\]

s.t. \( g(x) = 0 \)

\( c(x) \leq 0 \)

where \( x \in \mathbb{R}^{2n-1} \) is the state vector, \( z \in \mathbb{R}^m \) is the measurement vector, \( h(\cdot) : \mathbb{R}^{2n - 1} \rightarrow \mathbb{R}^m \) is the non-linear measurement function, \( g(\cdot) \) and \( c(\cdot) \) are the constraint functions, and \( J(\cdot) \) is the objective function that depends on the type of state estimation used to solve the problem.

The difference between the WLS and LAV approaches is the choice of the objective function \( J(\cdot) \). WLS based algorithms typically have \( J(\cdot) \) as a quadratic function whereas the LAV based approaches have \( J(\cdot) \) in the form of \( L_1 \) norm.
CHAPTER 2. OVERVIEW OF STATE ESTIMATION FORMULATIONS

2.2 WLS

The WLS problem can be formulated as the following optimization problem:

$$\min_x \ (z - h(x))^T W (z - h(x))$$

(2.2)

where $W \in \mathbb{R}^{m \times m}$ denotes the diagonal weight matrix and $z - h(x) \in \mathbb{R}^m$ is the measurement residual vector.

The weight matrix is related to the accuracy of the particular measurement. It is common to use $R^{-1}$ as the weight matrix, where $R$ is the measurement error covariance matrix.

At the solution of this problem, the first order optimality conditions have to be satisfied.

$$g(x) = \frac{\partial J(x)}{\partial x} = -H(x)^T W (z - h(x)) = 0$$

(2.3)

where $J(x) = (z - h(x))^T W (z - h(x))$ is the objective function and $H(x) = \frac{\partial h(x)}{\partial x}$ is the Jacobian of the measurement function $h(x)$.

To get a solution for (2.2), which is highly non-linear due to the non-linearity of the measurement function $h(\cdot)$, the system of equations represented by (2.3) is solved. One of the most common methods used to solve such problems is the Gauss-Newton method, which is briefly described below.

2.2.1 Gauss Newton solution algorithm for WLS

Gauss-Newton algorithm is the most commonly used approach for solving the WLS problem. It is an iterative procedure where the function $g(\cdot)$ is linearized by taking the Taylor expansion of the function around the current solution. The Taylor expansion of $g(\cdot)$ around a point $x_i$ neglecting the higher order terms gives:

$$g(x) = g(x_i) + \frac{\partial g(x_i)}{\partial x} (x - x_i) = 0$$

This can be solved by iteratively computing the series of updates $x - x_i$ until a point close enough to satisfying (2.3) is reached.

$$G(x_i) \cdot (x_{i+1} - x_i) = H(x)^T W (z - h(x))$$

(2.4)

where, $G(x) = \frac{\partial g(x)}{\partial x} \approx H(x)^T W H(x)$ is called the Gain Matrix.

The gain matrix is usually sparse and provided the system is observable, it is also symmetric and positive definite [3]. However, it is important to note that $G$ is only an approximation.

$$G(x) = \frac{\partial g(x)}{\partial x} = H(x)^T W H(x) - \frac{\partial h(x)^T}{\partial x} W (z - h(x))$$

The second term can be neglected here because $\frac{\partial h(x)^T}{\partial x}$ is quite small and the value of $z - h(x)$ is also close to zero near the solution.

The states of the system can then be obtained by iteratively solving (2.4) until the updates become smaller than a certain tolerance level.

The algorithm can be summarized as follows:

1. Set iteration counter as 0.
2. Initialize the state vector $x_0$ (Usually, a flat start is chosen).

3. Compute $h(x_i)$, $H(x_i)$, $g(x_i)$ and $G(x_i)$.

4. Solve (2.4) to obtain $x_{i+1}$.

5. Check if $x_{i+1} - x_i$ is less than the tolerance limit. If yes exit, else update the iteration counter and go to step 3.

To reduce the computational burden associated solving (2.4), $G$ can be decomposed into triangular matrices using well-established methods such as Cholesky or LU decomposition and the system of equations can be solved using a simple forward and backward sweep algorithm. Even though it is faster than simply taking the inverse of $G$, the decomposition needs to be carried out in every iteration. This is the main motivation behind the development of the decoupled approach for solving the state estimation problem \cite{5, 6}. Similar to the decoupled power flow algorithms, the pairs $P - \theta$ and $Q - V$ are strongly coupled. Moreover, $G$ is assumed to be constant throughout the course of the estimation and hence, the decomposition of $G$ is computed only once. The measurement function $h$ is separated into active (active power flows and active power injections) and reactive measurements (reactive power flows, reactive power injections and voltage magnitudes). The measurement Jacobian $H$ is also separated in this form and essentially the state estimation problem is split into two smaller sub-problems, one for active measurements and one for reactive measurements. This approximation helps in increasing the computational speed, but in some cases, this approach may fail to converge. Another drawback of using the decoupled formulation is that it cannot process current measurements \cite{3}.

As discussed in the previous chapter, the biggest drawback of the WLS approach is the fact that it is not robust to outliers. The measurements obtained from the SCADA system can be corrupted by gross errors, and in such a case the WLS estimates will be heavily biased. To overcome this problem, post-processing needs to be done to identify and eliminate such bad data. The general method to detect bad data is briefly discussed below.

### 2.2.2 Bad Data Processing

The problem of bad data processing in WLS-based state estimation was tackled in \cite{7} and \cite{8} in the early 1970s. Since then there have been significant improvements in the field, with methods for effectively identifying errors in parameters and topologies being developed. To limit the scope of this project, only errors in measurements are considered. The basic idea behind bad data processing is to use the redundancy present in the system to produce an estimate that is independent of the corrupted measurement. For that, the first step will be to identify the erroneous measurement. This step is possible only if the erroneous measurement is redundant, i.e. eliminating the measurement will not lead to a loss of observability. The basic techniques used for detection and identification are the $\chi^2$ test and the normalized residual test.

#### $\chi^2$ test

The main idea behind the application of $\chi^2$ test in state estimation, as proposed in \cite{8}, is that the objective function $J(x)$ can be approximated as a $\chi^2$ distribution with at most $(m - (2n - 1))$ degrees of freedom, i.e the difference between the number of measurements and the number of states. A confidence interval, say 95 $\%$, can be chosen by the user
and the value of $J(x)$ can be compared against the value corresponding to this confidence interval and degrees of freedom from the $\chi^2$ distribution table. If the value of $J(x)$ is greater, then there is a bad data present in the measurement set. This method has a very low computational cost, but it cannot identify the measurement containing the bad data. Moreover, since it is an approximation, it may fail to detect bad data in some cases.

**Normalized Residual Test**

In order to overcome the problems faced by the $\chi^2$ test, an alternative method to accurately detect and identify bad data was presented in [9]. This test is based on normalizing the measurement residuals $r$ and comparing its maximum value against a pre-defined threshold. It can be shown that

$$r_i^N = \frac{|r_i|}{\sqrt{\Omega_i}} = \frac{|r_i|}{\sqrt{R_{ii}S_{ii}}}$$

where, $r_i^N$ is the normalized residual for measurement $i$, $r_i$ is the residual for measurement $i$, $\Omega = SR$ is the residual covariance matrix, $R$ is the measurement error covariance matrix and $S$ is the residual sensitivity matrix, which is defined as

$$S = I - K = I - HG^{-1}H^T W$$

where $K = HG^{-1}H^T W$ is commonly known as the “hat” matrix.

Once the normalized residuals have been computed, the largest value of the $r_i^N$, say $r_k^N$ can be compared with a threshold (typically chosen as 3 or 4). If $r_k^N$ is greater than the threshold, then the measurement corresponding to that residual is the one with bad data. It can either be removed or corrected and the state estimation must be run once again and this procedure is repeated until all the bad data have been eliminated or corrected.

**Drawbacks**

The approach described above has certain drawbacks. First, the normalized residual test fails to give the correct estimate if the erroneous measurement is either critical (removal of this measurement will lead to loss of observability) or is a part of a critical pair (removal of both the measurements at the same time will lead to loss of observability). It is also interesting to analyze the performance of the normalized residual test in the presence of multiple bad data. It has been established that if the bad data are interacting and conforming, the normalized residual test fails [3].

Assuming that the bad data present in the measurements can be successfully detected and identified, the computational burden is considerably high. For every bad data present, the residual covariance matrix $\Omega$ needs to be computed, which is computationally intensive. Once the bad data have been identified, the state estimation must be repeated again with the updated set of measurements. Similar approaches are available for detecting errors in network parameters and topology, which are computationally demanding as well. The state estimation process along with bad data identification is depicted in the flowchart in Figure 2.1.

Even though the WLS algorithm is fast, its lack of robustness means that post-processing techniques are required, which can significantly slow down the overall procedure. This calls for more robust state estimation algorithms where such expensive post-processing methods are not required.
CHAPTER 2. OVERVIEW OF STATE ESTIMATION FORMULATIONS

2.3 LAV

The LAV approach for solving the state estimation problem was introduced as a way to make the solution algorithms more robust to outliers so that the need for post-processing techniques can be eliminated. The LAV problem attempts to find the estimate of the state by minimizing the $L_1$ norm of the difference between the acquired measurements and the estimated measurements. The LAV state estimation problem can be formulated as

$$
\min_x \quad c^T |r|
$$

s.t. \quad z - h(x) = r

(2.5)

where $c$ is a vector of ones and $r$ is the vector of measurement residuals.

The robustness of the LAV method is primarily due to the fact that the objective function minimizes the absolute value of the measurement residuals rather than the squares of the residuals, which makes it less sensitive to outliers or bad data. Moreover, this method finds the estimate by using the minimum number of data points required (equal to the total number of states), thereby filtering out the measurements which are outliers. On the other hand, this comes at an additional computational cost. In the WLS approach, it is possible to arrive at an analytic solution for minimizing the objective function $J(x)$ by solving $\frac{\partial J(x)}{\partial x} = 0$. Observing (2.5), it is clear that the objective function is not differentiable at 0. Hence, it is not possible to find the minima by setting the gradient of the objective function as 0. This requires other special solution techniques which are computationally expensive compared to WLS. Another drawback is the fact that the estimates are heavily biased when there is bad data present in leverage measurements which will be discussed.
By adding slack variables and manipulating the equations, the problem in (2.5) can be reformulated in a way that it can be solved by gradient-based methods.

Let $\zeta$ be defined such that $|r_i| \leq \zeta_i \forall i = 1$ to $m$. Then, the objective function of (2.5) can also be written as $c^T \times \zeta$. Now, additional slack variables can be introduced to change the inequality constraint $|r_i| \leq \zeta_i$ into an equality constraint. Let $l, k, u$ and $v$ be defined such that they are non-negative and

$$\begin{align*}
r_i + l_i &= \zeta_i \\
r_i - k_i &= -\zeta_i \\
u_i &= \frac{1}{2} k_i \\
v_i &= \frac{1}{2} l_i
\end{align*}$$

Clearly,

$$\begin{align*}
u_i + v_i &= \zeta_i \\
u_i - v_i &= r_i
\end{align*}$$

Hence, (2.5) can be rewritten as

$$\begin{align*}
\min_x c^T (u + v) \\
\text{s.t. } h(x) + u - v &= z \\
u, v &\geq 0
\end{align*} \tag{2.6}$$

where vectors $u$ and $v$ of dimensions $m \times 1$ are appended to the state vector $x$, which is now of dimensions $(2n - 1 + 2m) \times 1$.

Another popular approach to solve the LAV problem is to formulate it as a Linear Programming problem. The equivalence between LAV and LP was pointed out in [10, 11]. The problem in (2.6) can be converted into the form of a LP problem by linearizing the system around the operating point. The linearized version of (2.1) is given by

$$\Delta z = H \Delta x + e$$

Let $\Delta x^u$ and $\Delta x^l$ be defined such that they are non-negative and

$$\Delta x = \Delta x^u - \Delta x^l$$

Equation (2.6) can now be transformed into

$$\begin{align*}
\min_x c^T (u + v) \\
\text{s.t. } H \left( \Delta x^u - \Delta x^l \right) + u - v &= z \\
u, v, \Delta x^u, \Delta x^l &\geq 0
\end{align*}$$

Hence, the LAV state estimation problem can also be formulated as a series of LP problems which can then be solved using Simplex Algorithm or IP methods. Unfortunately, the Simplex approach suffers from scaling problems as it is not fast enough for large systems and hence is not a viable option for real time state estimation. On the other hand, with constant improvements being made in the use of IP methods for large scale problems, the IP based solution algorithm for (2.6) is becoming increasingly more attractive. The application of IP based solution algorithms has also been extensively studied in the literature.
[12, 13, 14, 15]. However, the mathematical details of solving the LAV state estimation using IP methods will not be presented here. More details on an IP solution method using a logarithmic barrier function can be found in [3].

### 2.3.1 Leverage Measurements

As discussed earlier, the solution of LAV state estimation becomes highly biased when gross errors are present in leverage measurements. Leverage measurements can have an undue influence on the obtained estimates and hence, if they are corrupted by gross errors, it may lead to a bad estimate. It is thus interesting to analyze the characteristics of these leverage measurements.

The hat matrix \( K \) defined as \( HG^{-1}HTW \) relates the true measurement \( z \) and the estimated measurement \( \hat{z} \) through

\[
\hat{z} = Kz
\]

Hence, the influence of measurement \( i \) on its estimate depends on the diagonal term \( K_{ii} \). Since \( K_{ii} = H_iG^{-1}H_i^TW \), the influence of measurement \( i \) on its estimate depends on \( H_i \), which is the \( i^{th} \) row of \( H \). Thus, in general, leverage measurements are those measurements that correspond to rows in the \( H \) matrix that have significantly higher values as compared to the other rows. When there is an error present in these measurements, the LAV method will try to fit this point in the solution, leading to a bad estimate. This can also be explained as follows. The residual \( r \) is related to the error \( e \) through

\[
r = Se = (I - K)e
\]

If the of \( K_{ii} \) corresponding to measurement \( i \) is large, its residual value will be low even when there is a large error present. Typically, leverage points are measurements with the following properties [16]:

- Flows and injections associated with short lines
- Injections at buses with a large number of incident branches

The inherent bad data rejection capability of LAV fails in the case of leverage measurements. On the other hand, the bad data processing of WLS using normalized residual test can detect single bad data in leverage measurements but is susceptible to some cases when there are multiple bad data in leverage measurements [17]. However, there has been a lot of research in the area of identification and elimination of leverage points. One way of negating the influence of leverage points is to identify and eliminate those measurements offline before the estimation is done. A simple way to identify leverage measurements is to compare the corresponding diagonal terms in \( K \) with a threshold, usually set as \( \frac{2(2n-1)}{n} \), and classify all measurements with \( K_{ii} \) greater than the threshold as leverage measurement. But this method can fail when there is a cluster of leverage measurements present. A more robust approach was presented in [18], which made use of projection statistics to identify and eliminate leverage measurements. Other methods include using estimators that are insensitive to leverage points such as least median of squares estimator [16] or to use linear transformations and scaling to make sure that there are no outliers present in the \( H \) matrix [19].

With the recent increase in the number of PMU based measurements available in the system, it is interesting to note that the elimination of leverage measurements can be done by simply scaling the \( H \) matrix [4]. This is due to the fact that the state estimation problem is linear and \( H \) matrix remains constant when the entire system is measured.
through PMUs. However, as PMUs were not used in this project, the system is highly non-linear. Hence, despite the attractive bad data removal properties of LAV-based approaches, the computational time is not competitive with WLS-based approaches especially for large systems.
Chapter 3

Proposed Algorithm

The most commonly encountered trade-off when choosing a state estimation algorithm is between robustness and computational time. For example, the LAV estimator is more robust to outliers compared to WLS, but it takes longer to compute the estimates. The goal of the proposed algorithm is to combine the convergence properties of WLS with the bad data removal properties of LAV. The idea is to cast the LAV state estimation problem as a smooth non-linear optimization problem that can be solved using existing gradient-based methods. This chapter outlines the mathematical formulation, solution methodology and the advantages and limitations of the proposed approach.

3.1 Mathematical Formulation

Instead of formulating the LAV problem in the form of (2.5), an alternative formulation is proposed

$$\min_x \sum_{i=1}^{m} \sqrt{(z_i - h_i(x))^2 + \epsilon}$$  \hspace{1cm} (3.1)

where $\epsilon$ is made progressively smaller and is close to zero at the solution.

Note that the objective function $J(x)$ is no longer non-differentiable at zero. This means an analytic solution for minimizing $J(x)$ can be obtained by solving $\frac{\partial J(x)}{\partial x} = 0$. The first order optimality conditions must be satisfied by the solution

$$g(x) = \frac{\partial J(x)}{\partial x} = \frac{\partial \left( \sum_{i=1}^{m} \sqrt{(z_i - h_i(x))^2 + \epsilon} \right)}{\partial x} = 0$$

Clearly, $g \in \mathbb{R}^{2n-1}$. Now, consider the $j$-th term of $g$ given by $g_j$.
g_j(x) = \frac{\partial J(x)}{\partial x_j} \\
= \sum_{i=1}^{m} \frac{1}{2\sqrt{(z_i - h_i(x))^2 + \epsilon}} \cdot 2(z_i - h_i(x)) \cdot \left( -\frac{\partial h_i(x)}{\partial x_j} \right) \\
= -\left( \frac{\partial h_1(x)}{\partial x_j} \cdots \frac{\partial h_m(x)}{\partial x_j} \right) \cdot \left( \begin{array}{c}
\frac{z_1-h_1(x)}{\sqrt{(z_1-h_1(x))^2 + \epsilon}} \\
\vdots \\
\frac{z_m-h_m(x)}{\sqrt{(z_m-h_m(x))^2 + \epsilon}} 
\end{array} \right) \\
Hence, \\
g(x) = \frac{\partial J(x)}{x} \\
= \left( \begin{array}{c}
\frac{\partial J(x)}{\partial x_1} \\
\vdots \\
\frac{\partial J(x)}{\partial x_{2n-1}} 
\end{array} \right) \\
= -\left( \begin{array}{ccc}
\frac{\partial h_1(x)}{\partial x_1} & \cdots & \frac{\partial h_m(x)}{\partial x_1} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_1(x)}{\partial x_{2n-1}} & \cdots & \frac{\partial h_m(x)}{\partial x_{2n-1}} 
\end{array} \right) \cdot \left( \begin{array}{c}
\frac{z_1-h_1(x)}{\sqrt{(z_1-h_1(x))^2 + \epsilon}} \\
\vdots \\
\frac{z_m-h_m(x)}{\sqrt{(z_m-h_m(x))^2 + \epsilon}} 
\end{array} \right) \\
= -H(x)^T \bar{h}(x) \\
where \( H^T \in \mathbb{R}^{2n-1 \times m} \) is the transpose of the measurement Jacobian matrix defined in Chapter 2 and \( \bar{h} \in \mathbb{R}^m \) is defined such that \\
\bar{h}(x) = \left( \begin{array}{c}
\frac{z_1-h_1(x)}{\sqrt{(z_1-h_1(x))^2 + \epsilon}} \\
\vdots \\
\frac{z_m-h_m(x)}{\sqrt{(z_m-h_m(x))^2 + \epsilon}} 
\end{array} \right) \\
The first order optimality conditions of (3.1) can now be written in a compact form as \\
g(x) = -H^T(x) \cdot \bar{h}(x) = 0 \quad (3.2) \\
Similar to WLS, to obtain a solution for (3.1), the system of equations in (3.2) can be solved using the Gauss Newton algorithm. Linearizing \( g(\cdot) \) by taking its Taylor approximation around a point \( x_i \) and neglecting the higher order terms, \\
g(x) = g(x_i) + \frac{\partial g(x_i)}{\partial x} (x - x_i) = 0 \\
This can be solved by iteratively computing the series of updates \( x - x_i \) until a point close enough to satisfying (3.2) is reached. \\
G(x_i) \cdot (x_{i+1} - x_i) = H^T(x) \cdot \bar{h}(x) \quad (3.3) \\
where, \( G(x) = \frac{\partial g(x)}{\partial x} \) the Gain Matrix.
The second term can be neglected as the value of \( \frac{\partial H}{\partial x} \) is quite small.

\( H^T \) has dimension of \((2n - 1) \times m\) and \( \frac{\partial h_i(x)}{\partial x} \) is a matrix of dimension \( m \times (2n - 1)\).

Hence, the gain matrix \( G \) has dimension \((2n - 1) \times (2n - 1)\). Consider the term \( \frac{\partial h_i(x)}{\partial x} \)

\[
\frac{\partial h_i(x)}{\partial x_j} = \frac{1}{(z_i - h_i(x))^2 + \epsilon} \left( \sqrt{(z_i - h_i(x))^2 + \epsilon} \cdot \left( -\frac{\partial h_i(x)}{\partial x_j} \right) \right)
- \frac{1}{(z_i - h_i(x))^2 + \epsilon} \cdot \frac{(z_i - h_i(x))}{2\sqrt{(z_i - h_i(x))^2 + \epsilon}} \cdot \left( -\frac{\partial h_i(x)}{\partial x_j} \right)
\]

\[
\Rightarrow \frac{\partial h_i(x)}{\partial x_j} = \frac{1}{(z_i - h_i(x))^2 + \epsilon} \left( \sqrt{(z_i - h_i(x))^2 + \epsilon} - \frac{(z_i - h_i(x))^2}{\sqrt{(z_i - h_i(x))^2 + \epsilon}} \right) \cdot \left( -\frac{\partial h_i(x)}{\partial x_j} \right)
= \frac{\epsilon}{(z_i - h_i(x))^2 + \epsilon}^{3/2} \cdot \left( -\frac{\partial h_i(x)}{\partial x_j} \right)
\]

Hence, \( \frac{\partial h_i(x)}{\partial x} \) can be written as

\[
\frac{\partial h_i(x)}{\partial x} = \begin{pmatrix}
\frac{\epsilon}{(z_i - h_i(x))^2 + \epsilon}^{3/2} \cdot \frac{\partial h_1(x)}{\partial x_1} & \cdots & \frac{\epsilon}{(z_i - h_i(x))^2 + \epsilon}^{3/2} \cdot \frac{\partial h_{n-1}(x)}{\partial x_{2n-1}} \\
\cdots & \ddots & \cdots \\
\frac{\epsilon}{(z_i - h_i(x))^2 + \epsilon}^{3/2} \cdot \frac{\partial h_{n-1}(x)}{\partial x_1} & \cdots & \frac{\epsilon}{(z_i - h_i(x))^2 + \epsilon}^{3/2} \cdot \frac{\partial h_{n-1}(x)}{\partial x_{2n-1}}
\end{pmatrix}
\]

\[
= -\mathcal{G}(x) \cdot H(x)
\]

where \( \mathcal{G}(x) \) is a diagonal matrix defined as

\[
\mathcal{G}(x) = \begin{pmatrix}
\frac{\epsilon}{(z_1 - h_1(x))^2 + \epsilon}^{3/2} \\
\vdots \\
\frac{\epsilon}{(z_{n-1} - h_{n-1}(x))^2 + \epsilon}^{3/2}
\end{pmatrix}
\]

Finally, the gain matrix \( G \) can now be written as

\[
G(x) = H^T(x) \cdot \mathcal{G}(x) \cdot H(x)
\]

(3.4)
3.2 Algorithm

Equation (3.3) can now be solved iteratively by decomposing $G$ into triangular matrices using Cholesky or LU decomposition and obtaining the updates via backward/forward substitution. An important factor to be considered is the step size along the search direction. Due to the formulation of the problem, it is possible that the updates obtained in (3.3) is large and might lead to divergence in some systems. To avoid this unwanted scenario, the updates could be adjusted by

$$x_{i+1} = x_i + \alpha_i d_i$$  \hspace{1cm} (3.5)

where $d_i$ is the search direction or the update obtained from (3.3) and $\alpha_i$ is the step length.

There are several methods available to compute $\alpha$, one of which is the Line Search algorithm. It ensures that the step size is large enough to obtain a sufficient decrease in the objective function. It can be formulated as

$$\min_{\alpha} f(x_i + \alpha d_i)$$

where $f(\cdot)$ is the objective function, $x_i$ is the current solution, $d_i$ is the search direction and $\alpha$ is the step size. Now, finding an exact solution for this problem is computationally expensive. Hence, an “inexact” line search can be performed, where the objective function value is sufficiently minimized. One of the inexact line search methods is the backtracking line search, which is based on the Armijo-Goldstein condition \cite{20} given by

$$f(x_i + \alpha_i d_i) \leq f(x_i) + c_1 \alpha_i d_i^T \Delta f(x_i)$$

where $\Delta f(x_i)$ is the gradient of the objective function at the current solution and $c_1$ is chosen such that $0 < c_1 < 1$. This condition ensures that there is a sufficient decrease in the objective function. An initial value of $\alpha$ is chosen and it is iteratively reduced until the condition is satisfied.

In (3.1) it was mentioned that the value of $\epsilon$ is progressively reduced such that it becomes close to zero at the solution. Choosing a proper initial value for $\epsilon$ is crucial for the convergence pattern. A low initial value may lead to longer time for convergence and, in some systems, may also lead to divergence. In the algorithm, the initial value of $\epsilon$ was chosen based on the system. The convergence pattern also depends on the way $\epsilon$ is updated during the process. When the value of $\epsilon$ is changed, the optimization problem that is being solved changes. One strategy is to solve the state estimation for a particular value of $\epsilon$ and once it converges, $\epsilon$ can be reduced and this process will be continued until $\epsilon$ is below a threshold. Another strategy is to reduce the value of $\epsilon$ when there is a sufficient decrease in the value of the gradient of the objective function. In this algorithm, a combination of both of these strategies is used. $\epsilon$ is reduced by a factor of 10 when the gradient reduces by a factor of 10 or when the difference $x_{i+1} - x_i$ is smaller than a threshold. The algorithm can now be summarized as follows

1. Set iteration counter as 0.
2. Initialize $\epsilon$ and the state vector $x_0$.
3. Compute the measurement function $h(x_i)$ and measurement Jacobian $H(x_i)$
4. Compute the gradient $g(x_i)$ using (3.2) and gain matrix $G(x_i)$ using (3.4).
5. Compute the updates by solving (3.3).
6. Perform line search to determine the step size $\alpha$.

7. Update the states based on (3.5) and reduce the value of $\epsilon$ based on the conditions stated above.

8. Check if $x_{i+1} - x_i$ is less than the tolerance limit. Check if $\epsilon$ is smaller than the threshold. If both statements are true, exit, else update the iteration counter and go to step 3.

When the value of $\epsilon$ is small, (3.1) can be written as

$$
\lim_{\epsilon \to 0} \left( \min_x \sum_{i=1}^{m} \sqrt{(z_i - h_i(x))^2 + \epsilon} \right)
$$

$$
= \min_x \sum_{i=1}^{m} |z_i - h_i(x)|
$$

$$
= \min_x c^T |z - h(x)|
$$

where $c \in \mathbb{R}^m$ is a vector of ones. Clearly, the above equation is exactly equivalent to (2.5). This means that this algorithm will possess the inherent bad data detection property associated with the LAV formulation. On the other hand, similar to the LAV formulation, this algorithm is vulnerable to outliers present in leverage points. However, this issue can be addressed through the methods suggested in Chapter 2. Hence, when there is bad data present, the computational time of this method will be comparable or, in large systems, be better than that of WLS with the normalized residual test.

The major advantage of solving the formulation given in (3.1) over the traditional LAV formulation is the computational efficiency. Since the new formulation can be solved using gradient-based methods such as Gauss-Newton, it is expected to be faster than using IP methods to solve the LAV formulation in (2.6) especially for large systems.
Chapter 4

Experimental Results

4.1 Simulation Setup

Experiments were conducted on the test systems present in MATPOWER. A comprehensive list of test systems used is shown in table 4.1. The measurements were generated by running a power flow on the test system. Gaussian noise was added to the results of the power flow to simulate a more realistic scenario. The redundancy was then reduced by randomly removing measurements without affecting the observability based on the analysis presented in [21].

The WLS and the proposed LAV algorithm were implemented in MATLAB whereas, the IP based LAV algorithm was implemented using IPOPT as a core solver. The test systems were run both with and without bad measurements and the computational time of all the algorithms was compared. The solution accuracy of the algorithms, quantified

<table>
<thead>
<tr>
<th>Test System</th>
<th>Redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ieee30</td>
<td>2.64</td>
</tr>
<tr>
<td>case57</td>
<td>2.67</td>
</tr>
<tr>
<td>case118</td>
<td>1.99</td>
</tr>
<tr>
<td>case300</td>
<td>2.62</td>
</tr>
<tr>
<td>1354pegase</td>
<td>3.29</td>
</tr>
<tr>
<td>1888rte</td>
<td>3.12</td>
</tr>
<tr>
<td>2383wp</td>
<td>2.95</td>
</tr>
<tr>
<td>2736sp</td>
<td>2.93</td>
</tr>
<tr>
<td>2746wop</td>
<td>2.94</td>
</tr>
<tr>
<td>2848rte</td>
<td>3.10</td>
</tr>
<tr>
<td>2869pegase</td>
<td>3.46</td>
</tr>
<tr>
<td>3012wp</td>
<td>2.92</td>
</tr>
<tr>
<td>3120sp</td>
<td>2.91</td>
</tr>
<tr>
<td>3375wp</td>
<td>2.98</td>
</tr>
<tr>
<td>6470rte</td>
<td>3.19</td>
</tr>
<tr>
<td>6515rte</td>
<td>3.18</td>
</tr>
<tr>
<td>9241pegase</td>
<td>3.65</td>
</tr>
<tr>
<td>13659pegase</td>
<td>4.49</td>
</tr>
</tbody>
</table>
as $MSE$ (Mean Square Error), is defined as

$$MSE = \sqrt{\frac{1}{2n-1} \sum_{i=1}^{2n-1} (x_i^{estimated} - x_i^{true})^2}$$  \hspace{1cm} (4.1)$$

The true state $x_i^{true}$ was obtained by running the power flow solution for each system.

The following test cases were simulated:

- Case 1: No bad measurements
- Case 2: One bad measurement
- Case 3: Two bad measurements, with one of them being a leverage measurement
- Case 4: Three bad measurements
- Case 5: Five bad measurements

In case 2, the value of the bad measurement was set as 0. In case 3, the value of one measurement was set as 0 and the value of the other measurement was set to the negative of the actual value of the measurement. In case 4, the values of two of the measurements were set as 0 and the value of the other measurement was set to the negative of its original value. In case 5, values of three measurements were set as 0, the value of one measurement was set to the negative of its original value and value of one measurement was corrupted by an error of 15 times the standard deviation. The test cases with bad measurements were not run for large systems as the computational time for the WLS algorithm with bad data processing was prohibitively high.

### 4.2 Without Bad Measurements

The computational times of the different algorithms in the absence of bad measurements are shown in table 4.2. As expected, WLS is the fastest of the three algorithms when there is no bad data present. The key observation here is the fact that the proposed LAV algorithm is faster than the IP-based LAV algorithm. This difference becomes more pronounced as the system size increases. For the 13659 bus test system, the proposed LAV is more than 4 times faster than the IP-based LAV. This observation validates the expectation that solving the LAV problem as an unconstrained minimization problem using gradient techniques is faster than solving it using IP techniques. The values of $MSE$ indicates that the solution accuracy of all the approaches is similar.
### Table 4.2: Computational time without bad measurements

<table>
<thead>
<tr>
<th>Test System</th>
<th>Time (seconds)</th>
<th>MSE ($\times 10^{-5}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WLS</td>
<td>IP-LAV</td>
</tr>
<tr>
<td>ieee30</td>
<td>0.006</td>
<td>0.198</td>
</tr>
<tr>
<td>case57</td>
<td>0.009</td>
<td>0.317</td>
</tr>
<tr>
<td>case118</td>
<td>0.01</td>
<td>0.26</td>
</tr>
<tr>
<td>case300</td>
<td>0.03</td>
<td>0.44</td>
</tr>
<tr>
<td>154pegase</td>
<td>0.11</td>
<td>1.16</td>
</tr>
<tr>
<td>1888rte</td>
<td>0.38</td>
<td>1.84</td>
</tr>
<tr>
<td>2383wp</td>
<td>0.25</td>
<td>2.21</td>
</tr>
<tr>
<td>2736sp</td>
<td>0.21</td>
<td>4.70</td>
</tr>
<tr>
<td>2746wop</td>
<td>0.27</td>
<td>2.58</td>
</tr>
<tr>
<td>2848rte</td>
<td>0.51</td>
<td>4.45</td>
</tr>
<tr>
<td>2869pegase</td>
<td>0.29</td>
<td>3.50</td>
</tr>
<tr>
<td>3012wp</td>
<td>0.23</td>
<td>3.15</td>
</tr>
<tr>
<td>3120sp</td>
<td>0.24</td>
<td>3.02</td>
</tr>
<tr>
<td>3375wp</td>
<td>0.33</td>
<td>3.51</td>
</tr>
<tr>
<td>6470rte</td>
<td>2.01</td>
<td>10.5</td>
</tr>
<tr>
<td>6515rte</td>
<td>2.32</td>
<td>16.1</td>
</tr>
<tr>
<td>9241pegase</td>
<td>1.91</td>
<td>14.2</td>
</tr>
<tr>
<td>13659pegase</td>
<td>2.07</td>
<td>53.0</td>
</tr>
</tbody>
</table>

### 4.3 With Bad Measurements

#### 4.3.1 IP-LAV vs Proposed LAV

In this section, the similarities in the bad data removal properties of the IP-based LAV algorithm and the proposed LAV algorithm are presented. The 300 bus test system is considered and 5 measurements are corrupted by gross error. Figure 4.1 represents the residual plot at the solution of the SE using the IP-LAV method. The red dots indicate the measurements that were corrupted by gross errors. It can be observed that the values of residuals are large only for the measurements that were corrupted with bad data and the rest of the measurements have almost zero residuals. Clearly, the estimate provided by the algorithm is unbiased and it illustrates the inherent bad data detection ability of the LAV estimator. Figure 4.2 shows the residual plot at the solution of the SE using the proposed LAV method. It can be seen that it is similar to Figure 4.1 and it can be concluded that the proposed algorithm has the same inherent bad data detection properties. This does not come as a surprise because, at the solution, the proposed algorithm is exactly equivalent to the traditional LAV formulation as pointed out in the previous chapter. On the flip side, this also means that the proposed algorithm will contain some of the undesired properties of the traditional LAV, namely the vulnerability to leverage measurements. Figure 4.3 shows the residual plot of IP-LAV when the bad data is present in the leverage measurement. When bad data was added in a leverage measurement (measurement 1355), the algorithm failed to provide an unbiased estimate. The same properties can be observed in Figure 4.4 thereby showing that the proposed LAV algorithm also suffers from the same drawback.
CHAPTER 4. EXPERIMENTAL RESULTS

Figure 4.1: Residual plot of IP-LAV

Figure 4.2: Residual Plot of Proposed LAV
Figure 4.3: Residual Plot of IP-LAV: bad data at leverage point

Figure 4.4: Residual Plot of Proposed LAV: bad data at leverage point
4.3.2 Test system: 1888 rte

One Bad Data

The 1888 bus RTE system is tested. In the first test, one measurement is corrupted by a gross error. Figure 4.5 shows the residual plot of the WLS algorithm when no post-processing is done to remove bad data. Non-zero residuals are found for measurements that are not corrupted by bad data. This shows that WLS provides a biased estimate in the presence of outliers. On the other hand, Figure 4.6 shows that the proposed LAV is unbiased to the outlier present. Figure 4.7 shows the residual plot of the WLS algorithm with bad data processing. Here, the bad data is identified using the Normalized Residual test and the measurement containing the bad data is subsequently removed from the measurement set. This can be observed in Figure 4.7 as there is no bad measurement present and the estimate is now unbiased.

![Figure 4.5: Residual Plot of WLS without bad data processing](image-url)
Figure 4.6: Residual Plot of Proposed LAV

Figure 4.7: Residual Plot of WLS with bad data processing
Bad Data at Leverage Measurement

In this test, two measurements are corrupted by gross errors in such a way that one of the measurements corresponds to a leverage point. Figure 4.8 shows the residual plot of the WLS algorithm without bad data processing. Similar to Figure 4.5, measurements that are not corrupted by bad data have non-zero residuals. The difference between this test case and the previous can be seen in the residual plot of the proposed LAV method given in Figure 4.9. It can be seen that when the bad data is present in a leverage measurement, the proposed LAV algorithm fails to provide an unbiased estimate. On the other hand, the normalized residual test was able to identify the bad data present in the leverage measurement and as a result, the estimates are unbiased in Figure 4.10.

![Residual Plot of WLS without bad data processing](image)

**Figure 4.8: Residual Plot of WLS without bad data processing**
CHAPTER 4. EXPERIMENTAL RESULTS

Figure 4.9: Residual Plot of Proposed LAV

Figure 4.10: Residual Plot of WLS with bad data processing
Multiple Bad Data

Next, the effect of multiple bad data is studied. In this test, three and five measurements are corrupted by gross errors, with none of them being a leverage measurement. Figures 4.11 and 4.14 show the residual plot of the WLS algorithm without bad data processing under the presence of three and five bad data. The characteristics of these plots are similar to those observed earlier. Multiple measurements that are not corrupted by bad data have non zero residual, thereby showing that WLS fails to provide an unbiased estimate in the presence of bad data. Figures 4.12 and 4.15 show the residual plot of the proposed LAV algorithm. From these plots, it can be concluded that even in the presence of multiple bad data the algorithm provides an unbiased estimate as long as the bad data is not present in leverage measurements. Figures 4.13 and 4.16 show the residual plots of the WLS algorithm after the removal of bad data. The normalized residual test had to be run multiple times to successfully identify and remove all bad data and the final estimate obtained is unbiased.

Figure 4.11: Residual Plot of WLS without bad data processing: 3 bad measurements
Figure 4.12: Residual Plot of Proposed LAV: 3 bad measurements

Figure 4.13: Residual Plot of WLS with bad data processing: 3 bad measurements
Figure 4.14: Residual Plot of WLS without bad data processing: 5 bad measurements

Figure 4.15: Residual Plot of Proposed LAV: 5 bad measurements
4.3.3 Test system: 1354pegase

The 1354 bus system is tested. The tests run on this system are the same as the ones run on the 1888 bus system. Figures 4.17 to 4.19 show the residual plots when one measurement is corrupted by a gross error. Similar to the observations made in the previous case, WLS without bad data processing fails to provide an unbiased estimate whereas, the proposed LAV and WLS with bad data processing provide an unbiased estimate.
Figures 4.20 to 4.22 show the residual plots when two measurements are corrupted by gross errors in such a way that one of the measurements corresponds to a leverage point. Clearly, WLS without bad data processing fails to provide an unbiased estimate whereas WLS with bad data processing provides an unbiased estimate. On the other hand, the proposed LAV fails to provide an unbiased estimate due to the effect of bad data present in leverage points, thereby highlighting the method’s vulnerability to outliers present in leverage points.
Figure 4.20: Residual Plot of WLS without bad data processing

Figure 4.21: Residual Plot of Proposed LAV
Figures 4.23 to 4.25 show the residual plots when three measurements are corrupted by gross errors. It can be observed that WLS without bad data processing fails to provide an unbiased estimate whereas, the proposed LAV and WLS with bad data processing provide an unbiased estimate.

Figure 4.22: Residual Plot of WLS with bad data processing

Figure 4.23: Residual Plot of WLS without bad data processing
Figures 4.26 to 4.28 show the residual plots when five measurements are corrupted by gross errors. Similar to the previous case, WLS without bad data processing fails to provide an unbiased estimate whereas, the proposed LAV and WLS with bad data processing provide an unbiased estimate.
Figure 4.26: Residual Plot of WLS without bad data processing

Figure 4.27: Residual Plot of Proposed LAV


**Figure 4.28: Residual Plot of WLS with bad data processing**

### 4.3.4 Comparison of Computational time

Table 4.3 presents the computational time of WLS (with bad data processing), IP-based LAV and the proposed LAV in the presence of bad data. It can be observed that when there is no bad data present, WLS is the fastest of the three algorithms. However, the computational time of WLS increases with an increase in the number of bad data whereas the LAV methods are fairly insensitive to the number of bad data present in the system. This is due to the computational burden associated with the post-processing step that is required to eliminate the effects of bad data in WLS. The effect of multiple bad data becomes more pronounced as the system size increases. In the 1354pegase and 1888 RTE systems, the LAV methods are faster than WLS even in the presence of a single bad data. Another observation to be made is that the proposed LAV is faster than the IP-based LAV. This is due to the fact that the proposed algorithm formulates the LAV problem as an unconstrained minimization problem.
### Table 4.3: Computational time with bad measurements

<table>
<thead>
<tr>
<th>Test System</th>
<th>No. of Bad Data</th>
<th>Time (seconds)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>WLS</td>
<td>IP-LAV</td>
<td>Proposed LAV</td>
</tr>
<tr>
<td>ieee30</td>
<td>0</td>
<td>0.006</td>
<td>0.198</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.010</td>
<td>0.270</td>
<td>0.038</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.013</td>
<td>0.218</td>
<td>0.039</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.017</td>
<td>0.225</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.025</td>
<td>0.222</td>
<td>0.061</td>
</tr>
<tr>
<td>case57</td>
<td>0</td>
<td>0.009</td>
<td>0.317</td>
<td>0.059</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.014</td>
<td>0.325</td>
<td>0.081</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.025</td>
<td>0.302</td>
<td>0.088</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.032</td>
<td>0.296</td>
<td>0.067</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.049</td>
<td>0.304</td>
<td>0.081</td>
</tr>
<tr>
<td>case118</td>
<td>0</td>
<td>0.01</td>
<td>0.26</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.17</td>
<td>0.25</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.32</td>
<td>0.27</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.44</td>
<td>0.27</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.74</td>
<td>0.51</td>
<td>0.09</td>
</tr>
<tr>
<td>case300</td>
<td>0</td>
<td>0.03</td>
<td>0.44</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.17</td>
<td>0.38</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.30</td>
<td>0.42</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.45</td>
<td>0.90</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.73</td>
<td>0.67</td>
<td>0.33</td>
</tr>
<tr>
<td>1354pegase</td>
<td>0</td>
<td>0.11</td>
<td>1.16</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5.10</td>
<td>1.20</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10.1</td>
<td>1.28</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>15.2</td>
<td>1.33</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>24.9</td>
<td>2.25</td>
<td>1.41</td>
</tr>
<tr>
<td>1888rte</td>
<td>0</td>
<td>0.38</td>
<td>1.84</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9.55</td>
<td>2.21</td>
<td>1.89</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>18.9</td>
<td>2.40</td>
<td>2.30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>28.2</td>
<td>2.35</td>
<td>2.08</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>47.1</td>
<td>2.05</td>
<td>1.92</td>
</tr>
</tbody>
</table>
The main motivation of this project was to develop a fast yet robust algorithm to solve the state estimation problem. The existing approaches present a trade-off between computational efficiency and robustness. The WLS algorithm is fast but is not robust to outliers. On the other hand, the traditional IP-based LAV algorithms are robust but are computationally demanding compared to WLS. The proposed algorithm reformulates the LAV problem as a non-linear optimization problem so that it can be solved using existing gradient-based methods. The following can be concluded from the simulation results:

- The proposed LAV algorithm and the IP-based LAV algorithm possess identical bad data rejection properties.
- The proposed LAV algorithm computationally outperforms the IP-based LAV algorithm, and this difference is considerably high for large systems.
- When there is no bad data present, WLS is the fastest of the three approaches. However, in the presence of bad data in large systems, the proposed LAV becomes the most efficient of the three algorithms.

Since the bad data rejection properties of the IP-based LAV and the proposed LAV algorithms are identical, both approaches are vulnerable to outliers present in leverage points. Although there are methods available to reduce or eliminate the effect of leverage points, it remains to be seen whether those methods can be successfully adapted to the proposed algorithm.
Bibliography


