A New Method of Determining the Transmission Line Parameters of an Untransposed Line using Synchrophasor Measurements

Bradley Shayne Lowe

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of:

Master of Science

In

Electrical Engineering

Jaime De La Reelopez, Chair

Virgilio A. Centeno

Robert M. Gardner

August 28, 2015

Blacksburg, Virginia

Keywords: synchronized measurements, line parameters, estimation, least squares, noise

A New Method of Determining the Transmission Line Parameters of an Untransposed Line using Synchrophasor Measurements

Bradley Shayne Lowe

ABSTRACT

Transmission line parameters play a significant role in a variety of power system applications. The accuracy of these parameters is of paramount importance. Traditional methods of determining transmission line parameters must take a large number of factors into consideration. It is difficult and in most cases impractical to include every possible factor when calculating parameter values. A modern approach to the parameter identification problem is an online method by which the parameter values are calculated using synchronized voltage and current measurements from both ends of a transmission line.

One of the biggest problems facing the synchronized measurement method is line transposition. Several methods have been proposed that demonstrate how the line parameters of a transposed line may be estimated. However, the present case of today's power systems is such that a majority of transmission lines are untransposed. While transposed line methods have value, they cannot be applied in real-world scenarios. Future efforts of using synchronized measurements to estimate transmission line parameters must focus on the development and refining of untransposed line methods.

This thesis reviews the existing methods of estimation transmission line parameters using synchrophasor measurements and proposes a new method of estimating the parameters of an untransposed line. After the proposal of this new method, a sensitivity analysis is conducted to determine its performance when noise is present in the measurements.

Acknowledgements

I would like to offer my appreciation to Dr. De La Reelopez for the guidance and encouragement he has provided me with throughout my years as an undergraduate and graduate student.

I would like to thank those who I have met while working for Dominion Virginia Power. The company has provided me with numerous opportunities and its employees have played an important role in furthering my interest in power systems.

I would like to thank my friends in the Electrical Engineering Department at Virginia Tech. The knowledge I have gained from working with you all has been invaluable. I would not be half the engineer I am today without your support.

I would like to thank the G.E.R.M.A.N Club of Virginia Tech. I will hold on to the teachings of the ritual wherever I go in life. From M – "It is not what you have, but what you are that expresses your worth as a man."

I would like to thank my girlfriend, Amanda Brown. Your love and friendship helped carry me through graduate school. Thank you for being there for me in the times I needed it the most. I cannot wait to see what the future has in store for us!

Finally, I offer a special thanks to my family. Specifically my parents Dale Lowe, Melanie Holman, and Johnny Holman, my grandparents Katrinka Burton and Hoyle Burton, and my siblings Katrina Lowe, Patrick Holman, Blaire Holman, and Johnny Ray Holman. You have always provided me with unconditional love and support. It is a blessing to have each one of you in my life.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Transmission Line Parameters and their Applications

The electrical performance of a transmission line is governed by four fundamental parameters [**1**]. These parameters are resistance, inductance, capacitance, and conductance. The resistance and inductance parameters form the series impedance of the line. The capacitance and conductance parameters form the shunt admittance. The symbol abbreviations and units for each of the parameters are listed in Table 1.1-1.

Quantity	**Symbol**	**Unit**
Resistance	R	Ohm (Ω)
Inductance	L	Henry (H)
Capacitance	C	Farad (F)
Conductance	G	Siemens (S or Ω^{-1})

Table 1.1-1: Symbols and Units for Transmission Line Parameters

The conductance parameter accounts for leakage current in transmission tower insulators. Under normal operating conditions, leakage current is minimal [**2**]. For this reason, the conductance parameter is usually ignored. In this thesis, the conductance parameter will NOT be considered. However, it will be included in the definitions presented throughout this section.

The series impedance and shunt admittance quantities of a transmission line can be expressed as either a total value for the entire line or as a function of unit length. The total value expression is referred to as the lumped parameter expression. The per unit length expression is referred to as the distributed parameter expression. Each of these representations is presented in Table 1.1-2.

Quantity	**Expression**	**Unit**
Series Impedance (per unit length)	$z = R + j\omega L$	Ω/m
Shunt Admittance (per unit length)	$y = G + j\omega C$	S/m
Total Series Impedance	$Z = zl$	Ω
Total Shunt Admittance	$Y = yl$	S
Operating Frequency	ω	radians/sec
Line Length	l	meter (m)

Table 1.1-2: Expressions for Series Impedance and Shunt Admittance

From Table 1.1-2, we see that both the series impedance and the shunt admittance quantities are complex valued. The resistance parameter is the real part of the series impedance and the conductance parameter is the real part of the shunt admittance. The imaginary components of both quantities are dependent on the operating frequency of the transmission line. The name given to the imaginary part of the series impedance is reactance. The name given to the imaginary part of the shunt admittance is susceptance. The symbols and units for these parameters are given in Table 1.1-3.

Quantity	Expression	Unit
Reactance	$X = j\omega L$	Ohm (Ω)
Susceptance	$B = j\omega C$	Siemens (S or Ω^{-1})

Table 1.1-3: Symbols and Units for Reactance and Susceptance

Transmission line parameters are extremely important to the field of power systems engineering. They have several uses and play a large role in how we design, operate, and protect transmission lines. If parameter values are calculated incorrectly, they can cause problems for the applications that use them. Some of the major applications include:

1. Power System Protection Relays
2. Transmission Planning Models
3. State Estimation

Power System Protection Relays

The two main types of relays used for power system protection are overcurrent and impedance (distance) relays. Of the two, impedance relays are likely more sensitive to inaccurate parameter values.

Overcurrent relays operate by monitoring the current through a transmission line and tripping a circuit breaker when the current seen exceeds a preset value. The pickup setting of an overcurrent relay is selected by examining the maximum load and minimum fault currents associated with a line segment [**3**]. In order to calculate these current values, knowledge of a transmission line's phase and sequence impedances are required. An inaccurate parameter value may introduce error to the trip setting of an overcurrent relay and cause improper operation.

An impedance relay operates by taking voltage and current measurements provided by instrument transformers and computing a ratio. This ratio represents the impedance seen by the relay. If the impedance seen by the relay falls below a set threshold, the relay sends a tripping signal to isolate the faulted line from the system.

Transmission line parameter values directly affect the reach of an impedance relay. If a zone setting is created with a line impedance value that is smaller than the actual value, the reach of the relay will be shorter than desired. If a zone setting is created with a line impedance value that is larger than the actual value, the reach of the relay will extend further than desired. Under and over reaching of impedance relays interferes with protection schemes and degrades power system reliability.

Transmission Planning Models

The goal of transmission expansion planning is to ensure that forecasted loads will be completely supplied during both normal and contingency conditions. Normal conditions are when all elements (lines, transformers, generators, etc.) that comprise the transmission system are in service. A contingency condition arises when one or more elements are taken out of service.

The North American Reliability Corporation (NERC) standards TPL- 001 through TPL-004 outline the requirements for network performance during different states of operation. To be in compliance with each standard, transmission system planners must show that all elements within their interconnection obey applicable thermal and voltage ratings.

Transmission system planners evaluate the performance of their interconnection using power system simulation software. To model contingency conditions, planners remove elements and run load flow analyses. This process is repeated until every possible contingency has simulated. If at any point during a contingency study, it is discovered that the system fails to meet the NERC requirements, the transmission planner must develop a solution that installs new transmission infrastructure and corrects the problem.

Line parameter values are extremely important to the transmission planning process An incorrect parameter value may jeopardize the integrity of a contingency study. For example, it could cause a line to appear within its thermal ratings when it reality it is overloaded and in violation of NERC standards. The opposite scenario is equally as probable. In which case the utility would invest valuable money and energy into fixing a problem that didn't actually exist.

State Estimation

The goal of state estimation is to provide a real-time view of power system conditions. When performing state estimation, the estimator requires measurements of real and reactive power flows across the network, the system topology, and parameters values for the lines, transformers, and generators that compose the system. When all of these quantities are available, an estimation technique is applied to yield the desired state variables. Usually the state variables are the voltage magnitude and phase at every node in the system.

When estimations are performed in succession, changes in node voltages and other variables of interest can be observed over time. This information allows system operators to better monitor and control the network. They are able to take action in critical moments in order to maintain network stability and ensure reliable service.

Line parameter values impact the results of the state estimation process. An inaccurate parameter value can result in a bad estimation of the network conditions. System operators who use these results are prone to make poor technical and economic decisions. For example, a state estimation performed with inaccurate parameter values may lead the operator to believe the system is in a stable condition, when in actuality it is not. If the operator does not take the appropriate action, parts of the system may experience failure.

1.2 Methods of Determining Line Parameters

Transmission line parameter values can be determined by one of the following methods:

1. Theoretical Calculation
2. Estimation using Synchronized Measurements

Theoretical Calculation

Traditionally, transmission line parameter values are calculated by analyzing the physical properties of the transmission line conductors, the tower configuration, and the environment in which the line is operating [**4**]. Various magnetic and electric field principles are applied. References [**1**], [**2**], and [**5**] can be consulted for detailed descriptions of line parameter calculation techniques. Some of the major factors that are taken into consideration when calculating each parameter are listed here:

- Resistance is influenced by conductor material, conductor spiraling, the operating frequency of the line, and the ambient temperature. Today, resistance values are obtained from experimental tables provided by conductor manufacturers. These tables document the resistances of different conductors based on the factors above and simplify the conductor selection process.

- Inductance is calculated by applying Ampere's law to solve for the internal and external flux linkages between the conductors that compose the transmission line. These linkages are influenced by the spacing between the conductors, conductor bundling, conductor stranding, and transposition.

- Capacitance is calculated by applying Gauss's law to solve for the electric field strength between the conductors that compose the transmission line. The electric field strength is influenced by the spacing between the conductors and conductor bundling. In some capacitance calculations, the effect of the earth is taken into consideration.

Estimation using Synchronized Measurements

As Phasor Measurement Units (PMUs) are further integrated into today's power systems, real-time computation of transmission line parameters is becoming a viable option [**6**]. There are various methods that propose how this can be achieved, but the basic approach is the same across the board. That is, GPS time-tagged voltage and current measurements are taken from both ends of a line and manipulated to estimate the line parameter values.

The method of estimating line parameters by way of synchronized measurements possess several advantages over the theoretical calculation method. For example, the exact configuration and properties of the line do not have to be known. Line parameter values can be estimated without taking into consideration any of the complex factors that influence them. Another advantage worth noting is that parameters values can be re-estimated as frequently as the phasor sampling rate permits. This allows the parameters values to be updated across the various applications that use them to reflect the changing conditions of the transmission network.

One of the key factors limiting the widespread use of the synchronized measurement method is line transposition. A majority of the transmission lines that compromise today's power system are not transposed. However, almost all of the estimation methods that have been proposed are for transposed lines! In fact, after an extensive literature review, only one untransposed line method was found. While transposed methods still have value, they cannot be applied in real-world scenarios. As stated, future efforts of using synchronized measurements to estimate transmission line parameters must focus on the development and refining of untransposed line methods.

1.3 Motivation and Objective

This thesis has several objectives. The first is to briefly outline what has been done in the area of studying concerning line parameter identification using PMU measurements. All existing transposed and untransposed line methods will be discussed. Next, a new untransposed line method with be presented and validated. A sensitivity analysis will be conducted to compare the performance of untransposed line methods when noise is present in the PMU measurements.

1.4 Thesis Outline

The organization of this thesis is as follows. Chapter 2 provides background information on the transmission line models that will be used throughout the work. Chapter 3 introduces all of the existing estimation methods for transposed and untransposed lines. It also presents and validates a new untransposed line method. Chapter 4 presents the sensitivity analysis for untransposed line methods. It discusses the factors that influence the accuracy of estimated parameter values, the type and origins of noise in synchrophasor measurements, and provides several observations in regards to the performance of each method. Chapter 5 draws conclusion on the research performed and offers suggestions for future work.

2 Nominal Pi Model

Models are developed for transmission lines based on their length. The three categories of line lengths are short, medium, and long. Short lines are normally considered lines that are less than 80 km long. Medium lines fall between 80 and 250 km in length. Long lines are classified has having lengths greater than 250 km. [**5**]

The short line model assumes that the shunt admittance is open-circuited uses lumped parameters. Medium lines are represented using the nominal pi model and use lumped parameters. Long lines are represented using the equivalent pi model and use distributed parameters.

This section will present the nominal pi model as it is the only model that will be used and referred to throughout this thesis. For more information regarding short and long line models, consult references [**2**] and [**5**].

2.1 Representation for General Three-Phase Line

The nominal pi-model is formed using a two-port network. The model for a general three phase transmission line is given by Figure 2.1-1. Note that the shunt admittance is divided into two equal parts and placed on the ends of the line. The series impedance is placed in middle of the line.

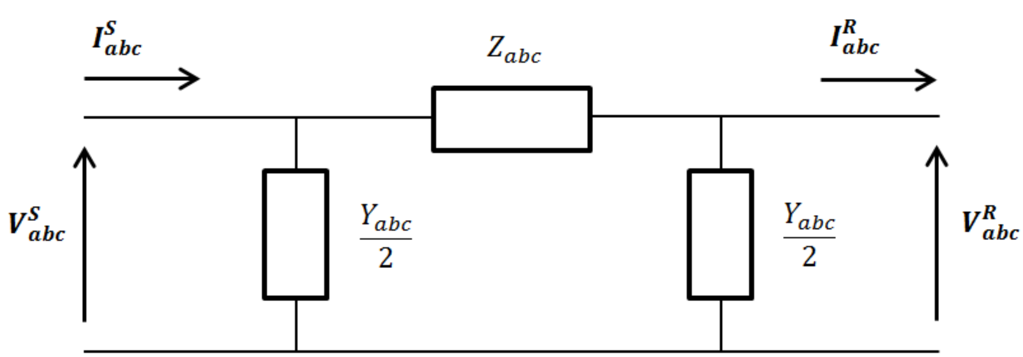

Figure 2.1-1: Nominal Pi Model for a Three-Phase Transmission Line

In Figure 2.1-1, V_{abc}^S and I_{abc}^S are the three phase, sending-end voltage and current phasors. Similarly, V_{abc}^R and I_{abc}^R are the three phase, receiving-end voltage and current phasors. The expression for each of these quantities is given below:

$$V_{abc}^S = \begin{bmatrix} V_a^S \\ V_b^S \\ V_c^S \end{bmatrix} \qquad V_{abc}^R = \begin{bmatrix} V_a^R \\ V_b^R \\ V_c^R \end{bmatrix} \qquad I_{abc}^S = \begin{bmatrix} I_a^S \\ I_b^S \\ I_c^S \end{bmatrix} \qquad I_{abc}^R = \begin{bmatrix} I_a^R \\ I_b^R \\ I_c^R \end{bmatrix}$$

For a three phase line, the phase impedance and admittance quantities are expressed as 3x3 matrices. The diagonal terms in the impedance matrix are the self-impedances of each phase. The off-diagonal terms are the mutual impedances between the phases. This notation also applies to the admittance matrix. The phase impedance and admittance matrices are given below:

$$Z_{abc} = \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix}$$

$$Y_{abc} = j\begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix}$$

Note that each entry in the phase impedance matrix has the following form:

$$Z_x = R_x + jX_x \qquad \text{where: } (x = a, b, c, ab, ac, or\ bc)$$

2.2 Representation for Transposed Lines

The three-phase model described in the previous section can be used to represent either transposed or untransposed lines. However, transposed lines are more often represented using only positive sequence quantities. In order show this representation, the phase quantities presented above need to be converted to sequence quantities.

To convert to the sequence domain, we apply the phase-to-sequence transformation matrix given by equation (2.2.1).

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \quad \text{where: } a = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \tag{2.2.1}$$

Equation (2.2.2) shows the relationship between the phase voltages and currents and the sequence voltages and currents.

$$\begin{aligned} V_{012}^S &= A^{-1} V_{abc}^S \qquad & V_{012}^R &= A^{-1} V_{abc}^R \\ I_{012}^S &= A^{-1} I_{abc}^S \qquad & I_{012}^R &= A^{-1} I_{abc}^R \end{aligned} \tag{2.2.2}$$

where:

$$V_{012}^S = \begin{bmatrix} V_0^S \\ V_1^S \\ V_2^S \end{bmatrix} \qquad V_{012}^R = \begin{bmatrix} V_0^R \\ V_1^R \\ V_2^R \end{bmatrix} \qquad I_{012}^S = \begin{bmatrix} I_0^S \\ I_1^S \\ I_2^S \end{bmatrix} \qquad I_{012}^R = \begin{bmatrix} I_0^R \\ I_1^R \\ I_2^R \end{bmatrix}$$

Equation (2.2.3) shows the relationship between the phase impedance/admittance matrices and the sequence impedance/ admittance matrices. Note that for both of the sequence matrices, the diagonal terms are the self-impedances of each sequence component and the off-diagonal terms are the mutual impedances between the sequence components.

$$\begin{aligned} Z_{012} &= A^{-1} Z_{abc} A \\ Y_{012} &= A^{-1} Y_{abc} A \end{aligned} \tag{2.2.3}$$

where:

$$Z_{012} = \begin{bmatrix} Z_0 & Z_{01} & Z_{02} \\ Z_{01} & Z_1 & Z_{12} \\ Z_{02} & Z_{12} & Z_2 \end{bmatrix}$$

$$Y_{012} = j\begin{bmatrix} B_0 & B_{01} & B_{02} \\ B_{01} & B_1 & B_{12} \\ B_{02} & B_{12} & B_2 \end{bmatrix}$$

For a completely transposed line, the phase impedance matrix has the characteristic such that each of its diagonal (self) terms are equal and each of its off-diagonal (mutual) terms are equal. The phase admittance matrix shares the same characteristic. When the phase to sequence transformation is applied to these matrices, the result are sequence matrices consisting of only diagonal terms. The transformation between the phase and sequence quantities of a transposed line is illustrated below:

$$Z_{abc} = \begin{bmatrix} Z_{\text{self}} & Z_{\text{mutual}} & Z_{\text{mutual}} \\ Z_{\text{mutual}} & Z_{\text{self}} & Z_{\text{mutual}} \\ Z_{\text{mutual}} & Z_{\text{mutual}} & Z_{\text{self}} \end{bmatrix} \quad \rightarrow \quad Z_{012} = \begin{bmatrix} Z_0 & 0 & 0 \\ 0 & Z_1 & 0 \\ 0 & 0 & Z_2 \end{bmatrix}$$

$$Y_{abc} = j \begin{bmatrix} B_{\text{self}} & B_{\text{mutual}} & B_{\text{mutual}} \\ B_{\text{mutual}} & B_{\text{self}} & B_{\text{mutual}} \\ B_{\text{mutual}} & B_{\text{mutual}} & B_{\text{self}} \end{bmatrix} \quad \rightarrow \quad Y_{012} = j \begin{bmatrix} B_0 & 0 & 0 \\ 0 & B_1 & 0 \\ 0 & 0 & B_2 \end{bmatrix}$$

This transformation implies that the sequence components are fully decoupled. This means the individual sequence voltages and currents are only determined by their corresponding sequence impedance and admittance quantities [**7**]. The simplifications made by the phase to sequence transformation allows us to represent a fully transposed line using only positive sequence quantities. The nominal pi model for transposed lines is found in Figure 2.2-1. In this figure, the following abbreviations are made:

$$Z = Z_1 = R_1 + jX_1$$
$$Y = Y_1 = jB_1$$

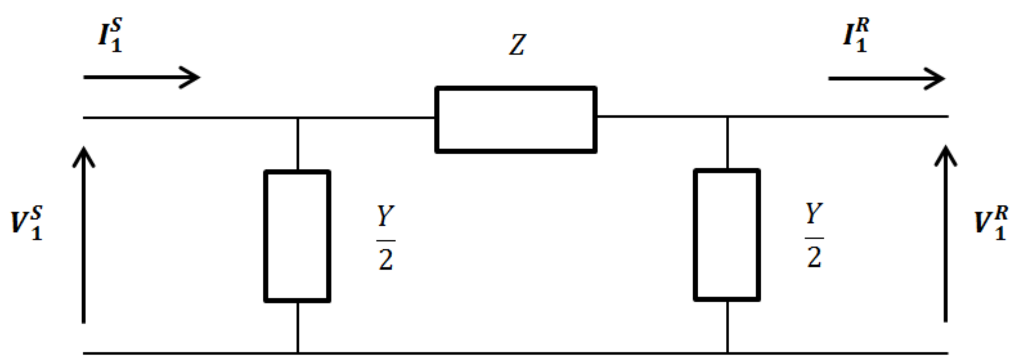

Figure 2.2-1: Nominal Pi Model for Transposed Line

3 Estimation of Line Parameters from Synchronized Measurements

Methods used to estimate line parameters from synchronized measurements can be divided into two categories – transposed line methods and untransposed line methods. Both types of methods utilize the nominal pi transmission line model to develop the estimation problem.

For transposed line methods, the parameters are the positive sequence impedance and susceptance. For untransposed line methods, the parameters are the entries of phase impedance and admittance matrices. The number of parameters for each method changes based on whether the estimation is being performed in the real or complex domain. This difference arises from the fact that in the real domain, the real and imaginary parts (resistance and reactance) of the impedance quantities count as a separate unknown parameter.

In the complex domain, transposed line methods have two unknown parameters and untransposed line methods have 12. In the real domain, transposed line methods have three unknown parameters and untransposed line methods have 18. The unknown parameters for the each case are listed in Table 2.2-1.

	Complex Domain	**Real Domain**
Transposed Line	Z_1 , Y_1	R_1 , X_1 , B_1
Untransposed Line	Z_x , Y_x $(x = a, b, c, ab, ac, or\ bc)$	R_x , X_x , B_x $(x = a, b, c, ab, ac, or\ bc)$

Table 2.2-1: Unknown Parameters of Estimation Methods

3.1 Existing Methods for Transposed Lines

There are numerous methods that estimate the line parameters of transposed lines. All of these methods use the nominal pi model for transposed lines described in Chapter 2.

3.1.1 Single Measurement Method

The single measurement method is proposed in [**8**] and is the simplest of all transposed line methods. As the name states, only one set of PMU measurements is required for its implementation. This method applies KVL and KCL to the nominal pi model for transposed lines. The following equations are produced:

$$V_1^S - V_1^R = Z\left[\frac{1}{2}Y\,(V_1^R) + I_1^R\right] \quad \textbf{(3.1.1)}$$

$$I_1^S - I_1^R = \frac{1}{2}Y[\,V_1^S + V_1^R\,] \quad \textbf{(3.1.2)}$$

Rearranging equation (3.1.2) yields an expression for Y. This expression is given by equation (3.1.3). Substituting equation (3.1.3) into equation (3.1.1) yields an expression for Z. This expression is given by equation (3.1.4).

$$Y = 2\left[\frac{I_1^S - I_1^R}{V_1^S + V_1^R}\right] \quad \textbf{(3.1.3)}$$

$$Z = \frac{(V_1^S)^2 - (V_1^R)^2}{I_1^R\,V_1^S + I_1^S V_1^R} \quad \textbf{(3.1.4)}$$

3.1.2 Double Measurement Method

The double measurement method was developed by the authors of [**9**]. In this method, two sets of unique PMU measurements are used to estimate the ABCD parameters of a transposed line. ABCD (or chain) parameters relate the sending and receiving end quantities of a two-port transmission line model. Each ABCD parameter is a single complex number and is dependent upon the transmission line parameters.

In equations (3.1.5) and (3.1.6), V_1^S and I_1^S are given in terms of V_1^R and I_1^R.

$$V_1^S = V_1^R + \frac{1}{2}\, Y\, Z\, V_1^R + Z\, I_1^R \tag{3.1.5}$$

$$I_1^S = Y\, V_1^R + \frac{1}{4}\, Y^2 Z\, V_1^R + I_1^R + \frac{1}{2}\, Y\, Z\, I_1^R \tag{3.1.6}$$

Substituting the coefficients of V_1^R and I_1^R, we produce the ABCD equations. These equations are given by (3.1.7) and (3.1.8). The matrix format is expressed by (3.1.9).

$$V_1^S = AV_1^R + BI_1^R \tag{3.1.7}$$

$$I_1^S = CV_1^R + DI_1^R \tag{3.1.8}$$

$$\begin{bmatrix} V_1^S \\ I_1^S \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_1^R \\ I_1^R \end{bmatrix} \tag{3.1.9}$$

From these equations, we make the following observations:

- The A parameter relates the sending end voltage to the receiving end voltage
- The B parameter relates the sending end voltage to the receiving end current
- The C parameter relates the sending end current to the receiving end voltage
- The D parameter relates the sending end current to the receiving end current

Comparing equations (3.1.5) and (3.1.6) with equations (3.1.7) and (3.1.8), we see the following equivalencies:

$$A = 1 + \frac{1}{2}\, YZ \tag{3.1.10}$$

$$B = Z \tag{3.1.11}$$

$$C = Y\, \left[1 + \frac{1}{4}\, YZ\right] \tag{3.1.12}$$

$$D = 1 + \frac{1}{2}\, YZ \tag{3.1.13}$$

Each set of PMU measurements used by the double measurements method produce two equations having the form of (3.1.9). The four resulting equations rearranged and represented in matrix form by (3.1.14). In this matrix, a “1” subscript indicates that a measurement belongs to the first measurement set. A “2” subscript indicates that a measurement belongs to the second measurement set.

$$\begin{bmatrix} V_{11}^{R} & I_{11}^{R} & 0 & 0 \\ 0 & 0 & V_{11}^{R} & I_{11}^{R} \\ V_{12}^{R} & I_{12}^{R} & 0 & 0 \\ 0 & 0 & V_{12}^{R} & V_{12}^{R} \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} V_{11}^{S} \\ I_{11}^{S} \\ V_{12}^{S} \\ I_{12}^{S} \end{bmatrix} \tag{3.1.14}$$

At this stage, Cramer’s method is applied to (3.1.14) to solve for the ABCD parameters. Cramer’s Method produces the following equations:

$$A = \frac{V_{12}^{S}\, I_{11}^{R} - V_{11}^{S}\, I_{12}^{R}}{\det} \tag{3.1.15}$$

$$B = \frac{V_{11}^{S}\, V_{12}^{R} - V_{11}^{R} V_{12}^{S}}{\det} \tag{3.1.16}$$

$$C = \frac{I_{12}^{S}\, I_{11}^{R} - I_{11}^{S}\, I_{12}^{R}}{\det} \tag{3.1.17}$$

$$D = \frac{V_{12}^{R}\, I_{11}^{S} - V_{11}^{R}\, I_{12}^{S}}{\det} \tag{3.1.18}$$

where:

$$\det = \mathrm{V}_{12}^{R}\, I_{11}^{R} - V_{11}^{R}\, I_{12}^{R}$$

The expressions for Z and Y based on equations (3.1.10) - (3.1.13) is as follows:

$$Z = B \tag{3.1.19}$$

$$Y = \frac{2(A-1)}{B} \tag{3.1.20}$$

3.1.3 Other Methods

More complex methods for determining the line parameters of a transposed line exist. This section summarizes these methods and provides references where more information can be found. The methods that will be discussed are listed below:

1. Multiple Measurement Method using Linear Least Squares # 1
2. Multiple Measurement Method using Linear Least Squares # 2
3. Multiple Measurement Method using Non-Linear Least Squares
4. Single Measurement Method using Newton-Raphson

Multiple Measurement Method using Linear Least Squares #1

This method is presented in reference [**10**]**.** Here, equation (3.1.5) is rearranged to solve for I_1^R. This result is substituted into equation (3.1.6) and solved for I_1^S. The following equations are produced:

$$I_1^R = \frac{1}{Z}\left[V_1^S - V_1^R\right] - \frac{1}{2} Y\, V_1^R \qquad \textbf{(3.1.21)}$$

$$I_1^S = \frac{1}{Z}\left[V_1^S - V_1^R\right] + \frac{1}{2} Y\, V_1^S \qquad \textbf{(3.1.22)}$$

Equations (3.1.21) and (3.1.22) are complex equations with a total of two complex unknown parameters (the values of Z and Y). These equations can be written in matrix format and arranged in such a manner to set up a least squares problem. When multiple measurements are available, the two unknown parameters can be estimated.

Multiple Measurement Method using Linear Least Squares # 2

This method is presented in reference [**11**]. Observing equations (3.1.7) and (3.1.8), we see that these are two complex equations with a total of four complex unknown parameters (the values of A,B,C, and D). This method expands these complex equations into four real equations with a total of eight real unknown parameters (the real and imaginary parts of A,B,C, and D). The four real equations are written in matrix format and arranged in such a manner to set up a least squares problem. When multiple PMU measurements are available, the eight unknown parameters can be estimated. Equations (3.1.19) and (3.1.20) can then be used to calculate the line parameters once the ABCD parameters are known.

Multiple Measurement Method using Non-Linear Least Squares

This method is presented in reference [**11**]. Observing equations (3.1.1) and (3.1.2), we see that these are two complex equations with a total of two complex unknown parameters (the values of Z and Y). This method expands these complex equations into four real equations with a total of three unknown parameters (the real and imaginary parts of Z and the imaginary part of Y). The four real equations are written in matrix format and arranged in such a manner to set up a non-linear least squares problem. When multiple PMU measurements are available, the three unknown parameters can be estimated. This method requires the construction of a Jacobian matrix and an initial guess for the parameter values.

Single Measurement Method using Newton-Raphson

This method is presented in reference [**12**]. Observing equations (3.1.5) and (3.1.6), we see that these are two complex equations with a total of two unknown parameters (the values of Z and Y). This method expands these complex equations into four real equations with a total of three unknown parameters (the real and imaginary parts of Z and the imaginary part of Y). The four real equations that were created are nonlinear and express the real and imaginary parts of the sending end voltage and current. They can be arranged to produce a set of four objective functions. Using a single measurement, the Newton-Raphson method is applied to the functions to solve for the unknown parameters. The user must form the Jacobian using the objective function partial derivatives and provide an initial guess for the parameter values

3.2 Existing Method for Untransposed Lines

As mentioned in the introduction, most of the transmission lines that compromise the power systems of today are not transposed. This means that the transposed line methods presented in the previous section can only be applied in a limited number of instances. To address this issue, untransposed line methods are required.

During the literature review, only **one** untransposed line method was found. This method is presented by Di Shi in references [**7**] and [**13**].

The existing method uses multiple measurements and ordinary least squares to estimate the unknown transmission line parameters directly. The derivation of the method begins by applying KVL and KCL to the nominal pi-model. KVL and KCL produce the following set of equations:

$$V_{abc}^{S} - V_{abc}^{R} = Z_{abc}\left[\frac{1}{2}Y_{abc}V_{abc}^{R} + I_{abc}^{R}\right] \quad \textbf{(3.2.1)}$$

$$I_{abc}^{S} - I_{abc}^{R} = \frac{1}{2}Y_{abc}[V_{abc}^{S} + V_{abc}^{R}] \quad \textbf{(3.2.2)}$$

When expanded, equations (3.2.1) and (3.2.2) produce six complex equations. The set of equations produced by (3.2.1) are nonlinear. This is because the unknown parameter matrices Z_{abc} and Y_{abc} are multiplied by one another. In order to make the equations linear, we multiply them by the inverse of the phase impedance matrix. The inverse is expressed by equation (3.2.4).

$$y_p = Z_{abc}^{-1} \quad \textbf{(3.2.3)}$$

$$y_p = \begin{bmatrix} y_a & y_{ab} & y_{ac} \\ y_{ab} & y_b & y_{bc} \\ y_{ac} & y_{bc} & y_c \end{bmatrix} \quad \textbf{(3.2.4)}$$

The linearized form of equation (3.2.1) is represented by (3.2.5).

$$y_p[V_{abc}^{S} - V_{abc}^{R}] = \frac{1}{2}Y_{abc}V_{abc}^{R} + I_{abc}^{R} \quad \textbf{(3.2.5)}$$

Expressing equations (3.2.5) and (3.2.2) in matrix format:

$$\begin{bmatrix} y_a & y_{ab} & y_{ac} \\ y_{ab} & y_b & y_{bc} \\ y_{ac} & y_{bc} & y_c \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta V_b \\ \Delta V_c \end{bmatrix} = j\frac{1}{2} \begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \begin{bmatrix} V_a^R \\ V_b^R \\ V_c^R \end{bmatrix} + \begin{bmatrix} I_a^R \\ I_b^R \\ I_c^R \end{bmatrix} \quad \textbf{(3.2.6)}$$

$$\begin{bmatrix} \Delta I_a \\ \Delta I_b \\ \Delta I_c \end{bmatrix} = j\frac{1}{2} \begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \begin{bmatrix} \Sigma V_a \\ \Sigma V_b \\ \Sigma V_c \end{bmatrix} \quad \textbf{(3.2.7)}$$

where:

$$\Delta V_x = V_x^S - V_x^R$$
$$\Delta I_x = I_x^S - I_x^R$$
$$\Sigma V_x = V_x^S + V_x^R$$
$$x = a, b, or\ c$$

Expansion of equations (3.2.6) and (3.2.7) yields the following six complex equations:

$$y_a \Delta V_a + y_{ab} \Delta V_b + y_{ac} \Delta V_c = j\frac{1}{2}\ [B_a V_a^R + B_{ab} V_b^R + B_{ac} V_c^R] + I_a^R \quad \textbf{(3.2.8)}$$

$$y_{ab} \Delta V_a + y_b \Delta V_b + y_{bc} \Delta V_c = j\frac{1}{2}\ [B_{ab} V_a^R + B_b V_b^R + B_{bc} V_c^R] + I_b^R \quad \textbf{(3.2.9)}$$

$$y_{ac} \Delta V_a + y_{bc} \Delta V_b + y_c \Delta V_c = j\frac{1}{2}\ [B_{ac} V_a^R + B_{bc} V_b^R + B_c V_c^R] + I_c^R \quad \textbf{(3.2.10)}$$

$$\Delta I_a = j\frac{1}{2}\ [B_a \Sigma V_a + B_{ab} \Sigma V_b + B_{ac} \Sigma V_c] \quad \textbf{(3.2.11)}$$

$$\Delta I_b = j\frac{1}{2}\ [B_{ab} \Sigma V_a + B_b \Sigma V_b + B_{bc} \Sigma V_c] \quad \textbf{(3.2.12)}$$

$$\Delta I_c = j\frac{1}{2}\ [B_{ac} \Sigma V_a + B_{bc} \Sigma V_b + B_c \Sigma V_c] \quad \textbf{(3.2.13)}$$

Usually, it is preferred to perform least squares estimation with real as opposed to complex equations. For this reason, the author of this method divides equations (3.2.8) - (3.2.13) into real and imaginary parts. The 12 real equations produced by this operation are given by equations (3.2.14) - (3.2.25). $Re[-]$ and $Im[-]$ represent the real and imaginary part of the input argument respectively. It is important to note that each entry of the inverse phase impedance matrix, y_p, is complex. The real and imaginary parts of each entry in the matrix are defined as such:

$$y_x = G_x + jT_x \quad (x = a, b, c, ab, ac, or\ bc)$$

$$Re[I_a^R] = G_aRe[\Delta V_a] - T_aIm[\Delta V_a] + G_{ab}Re[\Delta V_b] - T_{ab}Im[\Delta V_b] + G_{ac}Re[\Delta V_c] - T_{ac}Im[\Delta V_c] + \frac{1}{2}B_aIm[V_a^R] + \frac{1}{2}B_{ab}Im[V_b^R] + \frac{1}{2}B_{ac}Im[V_c^R] \quad \textbf{(3.2.14)}$$

$$Re[I_b^R] = G_{ab}Re[\Delta V_a] - T_{ab}Im[\Delta V_a] + G_bRe[\Delta V_b] - T_bIm[\Delta V_b] + G_{bc}Re[\Delta V_c] - T_{bc}Im[\Delta V_c] + \frac{1}{2}B_{ab}Im[V_a^R] + \frac{1}{2}B_bIm[V_b^R] + \frac{1}{2}B_{bc}Im[V_c^R] \quad \textbf{(3.2.15)}$$

$$Re[I_c^R] = G_{ac}Re[\Delta V_a] - T_{ac}Im[\Delta V_a] + G_{bc}Re[\Delta V_b] - T_{bc}Im[\Delta V_b] + G_cRe[\Delta V_c] - T_cIm[\Delta V_c] + \frac{1}{2}B_{ac}Im[V_a^R] + \frac{1}{2}B_{bc}Im[V_b^R] + \frac{1}{2}B_cIm[V_c^R] \quad \textbf{(3.2.16)}$$

$$Re[\Delta I_a] = -\frac{1}{2}B_aIm[\Sigma V_a] - \frac{1}{2}B_{ab}Im[\Sigma V_b] - \frac{1}{2}B_{ac}Im[\Sigma V_c] \quad \textbf{(3.2.17)}$$

$$Re[\Delta I_b] = -\frac{1}{2}B_{ab}Im[\Sigma V_a] - \frac{1}{2}B_bIm[\Sigma V_b] - \frac{1}{2}B_{bc}Im[\Sigma V_c] \quad \textbf{(3.2.18)}$$

$$Re[\Delta I_c] = -\frac{1}{2}B_{ac}Im[\Sigma V_a] - \frac{1}{2}B_{bc}Im[\Sigma V_b] - \frac{1}{2}B_cIm[\Sigma V_c] \quad \textbf{(3.2.19)}$$

$$Im[I_a^R] = G_aIm[\Delta V_a] + T_aRe[\Delta V_a] + G_{ab}Im[\Delta V_b] + T_{ab}Re[\Delta V_b] + G_{ac}Im[\Delta V_c] + T_{ac}Re[\Delta V_c] - \frac{1}{2}B_aRe[V_a^R] - \frac{1}{2}B_{ab}Re[V_b^R] - \frac{1}{2}B_{ac}Re[V_c^R] \quad \textbf{(3.2.20)}$$

$$Im[I_b^R] = G_{ab}Im[\Delta V_a] + T_{ab}Re[\Delta V_a] + G_bIm[\Delta V_b] + T_bRe[\Delta V_b] + G_{bc}Im[\Delta V_c] + T_{bc}Re[\Delta V_c] - \frac{1}{2}B_{ab}Re[V_a^R] - \frac{1}{2}B_bRe[V_b^R] - \frac{1}{2}B_{bc}Re[V_c^R] \quad \textbf{(3.2.21)}$$

$$Im[I_c^R] = G_{ac}Im[\Delta V_a] + T_{ac}Re[\Delta V_a] + G_{bc}Im[\Delta V_b] + T_{bc}Re[\Delta V_b] + G_cIm[\Delta V_c] + T_cRe[\Delta V_c] - \frac{1}{2}B_{ac}Re[V_a^R] - \frac{1}{2}B_{bc}Re[V_b^R] - \frac{1}{2}B_cRe[V_c^R] \quad \textbf{(3.2.22)}$$

$$Im[\Delta I_a] = \frac{1}{2}B_a Re[\Sigma V_a] + \frac{1}{2}B_{ab} Re[\Sigma V_b] + \frac{1}{2}B_{ac} Re[\Sigma V_c] \quad \textbf{(3.2.23)}$$

$$Im[\Delta I_b] = \frac{1}{2}B_{ab} Re[\Sigma V_a] + \frac{1}{2}B_b Re[\Sigma V_b] + \frac{1}{2}B_{bc} Re[\Sigma V_c] \quad \textbf{(3.2.24)}$$

$$Im[\Delta I_c] = \frac{1}{2}B_{ac} Re[\Sigma V_a] + \frac{1}{2}B_{bc} Re[\Sigma V_b] + \frac{1}{2}B_c Re[\Sigma V_c] \quad \textbf{(3.2.25)}$$

To create an expression for the least squares estimation, we define a measurement vector, unknown parameter vector, and a coefficient matrix. The coefficient matrix relates the measurement and parameter vectors and is constructed using the 12 equations found above.

The measurement vector, $Z,$ is defined as:

$$\begin{aligned} Z = [\ & Re[I_a^R] \quad Re[I_b^R] \quad Re[I_c^R] \quad Re[\Delta I_a] \quad Re[\Delta I_b] \quad Re[\Delta I_c] \quad \cdots \\ & \cdots \quad Im[I_a^R] \quad Im[I_b^R] \quad Im[I_c^R] \quad Im[\Delta I_a] \quad Im[\Delta I_b] \quad Im[\Delta I_c]\ ]^T \end{aligned} \quad \textbf{(3.2.26)}$$

The unknown parameter vector, $\theta,$ is defined as:

$$\theta = [G_a\ T_a\ G_b\ T_b\ G_c\ T_c\ G_{ab}\ T_{ab}\ G_{ac}\ T_{ac}\ G_{bc}\ T_{bc}\ B_a\ B_b\ B_c\ B_{ab}\ B_{ac}\ B_{bc}]^T \quad \textbf{(3.2.27)}$$

The coefficient matrix, H, is defined as:

$$H = [H_1 \quad H_2 \quad H_3] \quad \textbf{(3.2.28)}$$

where:

$$H_1 = \begin{bmatrix} Re[\Delta V_a] & -Im[\Delta V_a] & 0 & 0 & 0 & 0 \\ 0 & 0 & Re[\Delta V_b] & -Im[\Delta V_b] & 0 & 0 \\ 0 & 0 & 0 & 0 & Re[\Delta V_c] & -Im[\Delta V_c] \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ Im[\Delta V_a] & Re[\Delta V_a] & 0 & 0 & 0 & 0 \\ 0 & 0 & Im[\Delta V_b] & Re[\Delta V_b] & 0 & 0 \\ 0 & 0 & 0 & 0 & Im[\Delta V_c] & Re[\Delta V_c] \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} Re[\Delta V_b] & -Im[\Delta V_b] & Re[\Delta V_c] & -Im[\Delta V_c] & 0 & 0 \\ Re[\Delta V_a] & -Im[\Delta V_a] & 0 & 0 & Re[\Delta V_c] & -Im[\Delta V_c] \\ 0 & 0 & Re[\Delta V_a] & -Im[\Delta V_a] & Re[\Delta V_b] & -Im[\Delta V_b] \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ Im[\Delta V_b] & Re[\Delta V_b] & Im[\Delta V_c] & Re[\Delta V_c] & 0 & 0 \\ Im[\Delta V_a] & Re[\Delta V_a] & 0 & 0 & Im[\Delta V_c] & Re[\Delta 0 V_c] \\ 0 & 0 & Im[\Delta V_a] & Re[\Delta V_a] & Im[\Delta V_b] & Re[\Delta V_b] \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H_3 = \frac{1}{2} \begin{bmatrix} Im[V_a^R] & 0 & 0 & Im[V_b^R] & Im[V_c^R] & 0 \\ 0 & Im[V_b^R] & 0 & Im[V_a^R] & 0 & Im[V_c^R] \\ 0 & 0 & Im[V_c^R] & 0 & Im[V_a^R] & Im[V_b^R] \\ -Im[\Sigma V_a] & 0 & 0 & -Im[\Sigma V_b] & -Im[\Sigma V_c] & 0 \\ 0 & -Im[\Sigma V_b] & 0 & -Im[\Sigma V_a] & 0 & -Im[\Sigma V_c] \\ 0 & 0 & -Im[\Sigma V_c] & 0 & -Im[\Sigma V_a] & -Im[\Sigma V_b] \\ -Re[V_a^R] & 0 & 0 & -Re[V_b^R] & -Re[V_c^R] & 0 \\ 0 & -Re[V_b^R] & 0 & -Re[V_a^R] & 0 & -Re[V_c^R] \\ 0 & 0 & -Re[V_c^R] & 0 & -Re[V_a^R] & -Re[V_b^R] \\ Re[\Sigma V_a] & 0 & 0 & Re[\Sigma V_b] & Re[\Sigma V_c] & 0 \\ 0 & Re[\Sigma V_b] & 0 & Re[\Sigma V_a] & 0 & Re[\Sigma V_c] \\ 0 & 0 & Re[\Sigma V_c] & 0 & Re[\Sigma V_a] & Re[\Sigma V_b] \end{bmatrix}$$

The definitions above are arranged according to (3.2.29) to set up the least squares problem.

$$Z = H\theta \tag{3.2.29}$$

To estimate the unknown parameters, 3-phase voltage and current phasor measurements from both ends of the line are required. Measurements taken at one time instant constitute a sample. The 12 real equations discussed above are needed to describe the interdependences of one sample [**13**]. If multiple samples are available, the undetermined set of equations in (3.2.29) becomes an overdetermined set of equations and least squares may be applied [**13**].

The dimensions of $Z, H,$ and θ are dependent on the number of samples available. Table 3.2-1 shows the dimensions of each vector/matrix given that N samples are available.

Vector/Matrix	**Size** [Row x Column]
Z	$12N$ x 1
H	$12N$ x 18
θ	18 x 1

Table 3.2-1: Dimensions of Variables used in Existing Method Estimation Process

The unknown parameter vector can be estimated using the familiar least square result:

$$\theta = (H^T H)^{-1}(H^T Z) \tag{3.2.30}$$

Once the unknown parameter vector is estimated, the inverse phase impedance matrix and the shunt admittance matrix can be constructed. The phase impedance matrix is found by inverting the inverse phase impedance matrix (refer to equation (3.2.3)).

3.3 Proposed Method for Untransposed Lines

A new untransposed line method is proposed in this section. The inspiration for its creation was drawn from two sources – the double measurement method presented in section 3.1.2 and the multiple measurement method using linear least squares #2 presented in section 3.1.3.

The proposed method for determining the line parameters of an untransposed line is similar to the existing method in that it utilizes multiple PMU measurements and the least squares estimation technique. However, the manner in which the least squares equations are formed and the unknown parameters are identified is different. The proposed method begins by expressing the relationship between the sending and receiving end quantities using the nominal pi-model. That is, V_{abc}^{S} and I_{abc}^{S} are given in terms of V_{abc}^{R} and I_{abc}^{R} . The equations described are listed below:

$$V_{abc}^{S} = V_{abc}^{R} + \frac{1}{2} Z_{abc}\, Y_{abc}\, V_{abc}^{R} + Z_{abc}\, I_{abc}^{R} \quad \textbf{(3.3.1)}$$

$$I_{abc}^{S} = Y_{abc} V_{abc}^{R} + \frac{1}{4} Y_{abc}\, Z_{abc}\, Y_{abc}\, V_{abc}^{R} + I_{abc}^{R} + \frac{1}{2} Y_{abc} Z_{abc}\, I_{abc}^{R} \quad \textbf{(3.3.2)}$$

A reader may realize that equations (3.3.1) and (3.3.2) are an extension of the ABCD parameters concept. As stated previously, ABCD parameters relate the sending and receiving end quantities of a two-port transmission line model. For the untransposed case, A,B,C, and D become matrices of complex numbers. The dimension of each of these matrices is 3x3.This means that each matrix contains 9 entries. Across all four matrices, there results a total of 36 parameters. Since the ABCD parameters form a linear set of equations, it is possible to estimate these 36 parameters using linear least squares.

The same number of unknowns would be created weather this method was carried out in the phase or sequence domain. It is the author's preference to work in the phase domain. This would eliminate the middle step of needing to convert the collected PMU measurements from phase to sequence quantities. The ABCD equations represented in the phase domain are given as:

$$V_{abc}^{S} = A\, V_{abc}^{R} + B I_{abc}^{R} \quad \textbf{(3.3.3)}$$

$$I_{abc}^{S} = C\, V_{abc}^{R} + D I_{abc}^{R} \quad \textbf{(3.3.4)}$$

Comparing (3.3.1) and (3.3.2) with (3.3.3) and (3.3.4), we see the following equivalences:

$$A = \frac{1}{2} Z_{abc}\, Y_{abc} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.5}$$

$$B = Z_{abc} \tag{3.3.6}$$

$$C = \frac{1}{4} Y_{abc}\, Z_{abc}\, Y_{abc} + Y_{abc} \tag{3.3.7}$$

$$D = \frac{1}{2} Y_{abc} Z_{abc} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.8}$$

Expressing equations (3.3.5), (3.3.6), (3.3.7), and (3.3.8) in matrix form:

$$\begin{bmatrix} A_1 & A_2 & A_7 \\ A_4 & A_5 & A_8 \\ A_7 & A_6 & A_9 \end{bmatrix} = j\frac{1}{2} \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix} \begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.9}$$

$$\begin{bmatrix} B_1 & B_2 & B_7 \\ B_4 & B_5 & B_8 \\ B_7 & B_6 & B_9 \end{bmatrix} = \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix} \tag{3.3.10}$$

$$\begin{aligned} \begin{bmatrix} C_1 & C_2 & C_7 \\ C_4 & C_5 & C_8 \\ C_7 & C_6 & C_9 \end{bmatrix} = -\tfrac{1}{4} &\begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix} \begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \\ +j &\begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \end{aligned} \tag{3.3.11}$$

$$\begin{bmatrix} D_1 & D_2 & D_7 \\ D_4 & D_5 & D_8 \\ D_7 & D_8 & D_9 \end{bmatrix} = j\frac{1}{2} \begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.12}$$

Equations (3.3.9) - (3.3.12) express the relationship between the ABCD parameters and the transmission line parameters. We see that the B matrix is equivalent to the phase impedance matrix. The D matrix is the transpose of the A matrix and either of these may be used alongside the phase impedance matrix to solve for the shunt admittance matrix. With this knowledge, we can continue to form the least squares problem knowing that estimation of the ABCD parameters will also yield an estimation of the transmission line parameters.

Combining equations (3.2.3) and (3.2.4) into one concise matrix format yields equation (3.3.13):

$$\begin{bmatrix} V_{abc}^S \\ I_{abc}^S \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_{abc}^R \\ I_{abc}^R \end{bmatrix} \tag{3.3.13}$$

Expanding equation (3.3.13) results in equation (3.3.14):

$$\begin{bmatrix} V_a^S \\ V_b^S \\ V_c^S \\ I_a^S \\ I_b^S \\ I_c^S \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 & B_1 & B_2 & B_3 \\ A_4 & A_5 & A_6 & B_4 & B_5 & B_6 \\ A_7 & A_8 & A_9 & B_7 & B_8 & B_9 \\ C_1 & C_2 & C_3 & D_1 & D_2 & D_3 \\ C_4 & C_5 & C_6 & D_4 & D_5 & D_6 \\ C_7 & C_8 & C_9 & D_7 & D_8 & D_9 \end{bmatrix} \begin{bmatrix} V_a^R \\ V_b^R \\ V_c^R \\ I_a^R \\ I_b^R \\ I_c^R \end{bmatrix} \tag{3.3.14}$$

Performing matrix multiplication to (3.3.14) produces the following six complex equations:

$$V_a^S = A_1V_a^R + A_2V_b^R + A_3V_c^R + B_1I_a^R + B_2I_b^R + B_3I_c^R \tag{3.3.15}$$

$$V_b^S = A_4V_a^R + A_5V_b^R + A_6V_c^R + B_4I_a^R + B_5I_b^R + B_6I_c^R \tag{3.3.16}$$

$$V_C^S = A_7V_a^R + A_8V_b^R + A_9V_c^R + B_7I_a^R + B_8I_b^R + B_9I_c^R \tag{3.3.17}$$

$$I_a^S = C_1V_a^R + C_2V_b^R + C_3V_c^R + D_1I_a^R + D_2I_b^R + D_3I_c^R \tag{3.3.18}$$

$$I_b^S = C_4V_a^R + C_5V_b^R + C_6V_c^R + D_4I_a^R + D_5I_b^R + D_6I_c^R \tag{3.3.19}$$

$$I_C^S = C_7V_a^R + C_8V_b^R + C_9V_c^R + D_7I_a^R + D_8I_b^R + D_9I_c^R \tag{3.3.20}$$

Unlike the existing method, the six complex equations listed above will not be converted into 12 real equations. This is because in doing so, an additional 36 unknown parameters would be created -one unknown for the real and complex parts of each entry in the A,B,C, and D matrices. Another disadvantage of dividing the complex equations into real and imaginary parts is that a higher number of samples would be required to estimate the unknowns. At this point, we are now able to form the measurement vector, parameter vector, and coefficient matrix needed for least squares estimation.

The measurement vector, Y, is defined as:

$$Y = \begin{bmatrix} V_{abc}^{S} \\ I_{abc}^{S} \end{bmatrix} = \begin{bmatrix} V_a^S & V_b^S & V_c^S & I_a^S & I_b^S & I_c^S \end{bmatrix}^T \tag{3.3.21}$$

The unknown parameter vector, σ, is defined as:

$$\sigma = \begin{bmatrix} A_N & B_N & C_N & D_N \end{bmatrix}^T \tag{3.3.22}$$

where:

$$A_N = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{bmatrix} \qquad B_N = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \end{bmatrix} \qquad C_N = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{bmatrix} \qquad D_N = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \\ D_8 \\ D_9 \end{bmatrix}$$

The coefficient matrix, X, is defined as:

$$X = \begin{bmatrix} X_V & X_I & 0 & 0 \\ 0 & 0 & X_V & X_I \end{bmatrix} \tag{3.3.23}$$

where:

$$X_V = \begin{bmatrix} V_a^R & V_b^R & V_c^R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & V_a^R & V_b^R & V_c^R & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V_a^R & V_b^R & V_c^R \end{bmatrix}$$

$$X_I = \begin{bmatrix} I_a^R & I_b^R & I_c^R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_a^R & I_b^R & I_c^R & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_a^R & I_b^R & I_c^R \end{bmatrix}$$

In equation (3.3.23), each zero entry of the X matrix is equal to a 3x9 matrix of zeros. The definitions above are arranged according to equation (3.3.24) to set up the least squares problem.

$$Y = X\sigma \tag{3.3.24}$$

Just like the existing method, 3-phase voltage and current phasor measurements from both ends of the line are required to estimate the unknown parameters. Six equations ((3.3.15) - (3.3.20)) describe the interdependences of one sample. If multiple samples are available, the undetermined set of equations in (3.3.24) becomes an overdetermined set of equations and least squares may be applied.

The dimensions of $Y, X,$ and σ are dependent on the number of samples available. Table 3.3-1 shows the dimensions of each vector/matrix given that N samples are available.

Vector/Matrix	**Size** [Row x Column]
Y	6N x 1
X	6N x 36
σ	36 x 1

Table 3.3-1: Dimensions of Variables used in Proposed Method Estimation Process

To solve for the unknown parameter vector, σ, the usual least squares result must be adjusted so that each transpose operation is replaced with a conjugate transpose operation. This is to reflect the fact that complex numbers are present in the coefficient matrix. The solution is therefore given as follows:

$$\sigma = (\bar{X}^{T} X)^{-1}(\bar{X}^{T} Y) \tag{3.3.25}$$

Once the unknown parameter vector is estimated, the B matrix can be constructed to give the phase impedance matrix (refer to equation (3.3.10)). To solve for the shunt admittance matrix, we examine the relationship between the A matrix and the phase impedance parameters. Note that this process can also be carried out using the D matrix.

Rearranging equation (3.3.5), we arrive at an expression for the shunt admittance matrix. This expression is given by equation (3.3.26):

$$Y_{abc} = 2\left(A - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) Z_{abc}^{-1} \qquad \textbf{(3.3.26)}$$

The matrix expanded form of equation (3.3.26) is represented by equation (3.3.27):

$$j\begin{bmatrix} B_a & B_{ab} & B_{ac} \\ B_{ab} & B_b & B_{bc} \\ B_{ac} & B_{bc} & B_c \end{bmatrix} = 2\left(\begin{bmatrix} A_1 & A_2 & A_7 \\ A_4 & A_5 & A_8 \\ A_7 & A_6 & A_9 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \begin{bmatrix} Z_a & Z_{ab} & Z_{ac} \\ Z_{ab} & Z_b & Z_{bc} \\ Z_{ac} & Z_{bc} & Z_c \end{bmatrix}^{-1} \qquad \textbf{(3.3.27)}$$

3.3.1 Validation of Proposed Method

In this section, the proposed method for determining the line parameters of an untransposed line will be validated. The existing method is validated in [**7**], but for comparison with the proposed method, it will undergo the same validation test.

As stated in Sections 3.2 and 3.3, each of the estimation techniques utilizes multiple PMU measurement samples and linear least squares to identify the parameters of a transmission line. The primary difference between the two methods is the minimum number of samples required to estimate the parameters. The minimum number of samples is found my dividing the number of rows of the unknown parameter vector by the number of rows of the measurement vector. If this operation produces a fraction, the result should be rounded up to the nearest whole number.

The existing method requires at least two samples to estimate the transmission line parameters whereas the proposed method requires six. Either of the methods should be able to estimate the line parameter values with relatively low error as long as noiseless samples are used during the estimation process. Chapter 4 will address the performance of each method when noise is present in to the measurements.

In order for each of the methods to work, the samples selected for linear least squares must meet one of the following criteria:

- All measurements must represent the line under a unique loading condition
- All measurement must represent the line during a unique system unbalance
- OR the measurements must be from a combination of unique loading and unbalanced conditions

If the measurements used for parameter estimation are too similar, the coefficient matrix will become rank deficient. If this occurs, an estimation for the parameters cannot be obtained.

Lastly, it is worth nothing that although the methods that will be validated are categorized as “untransposed line methods”, they can be used regardless of whether or not the line is transposed. If these methods are applied to a fully transposed line, 18 parameters will still be estimated. However, the self and mutual terms of each unknown parameter matrix will be equal.

The transmission line that will be used to perform the validation is an overhead, untransposed, three phase line that uses four conductors per phase and has two shielding wires. The geometrical configuration of the line in a Cartesian coordinate system is shown in Figure 3.3-1. The distances associated with the geometrical configuration can be found in Table 3.3-2.

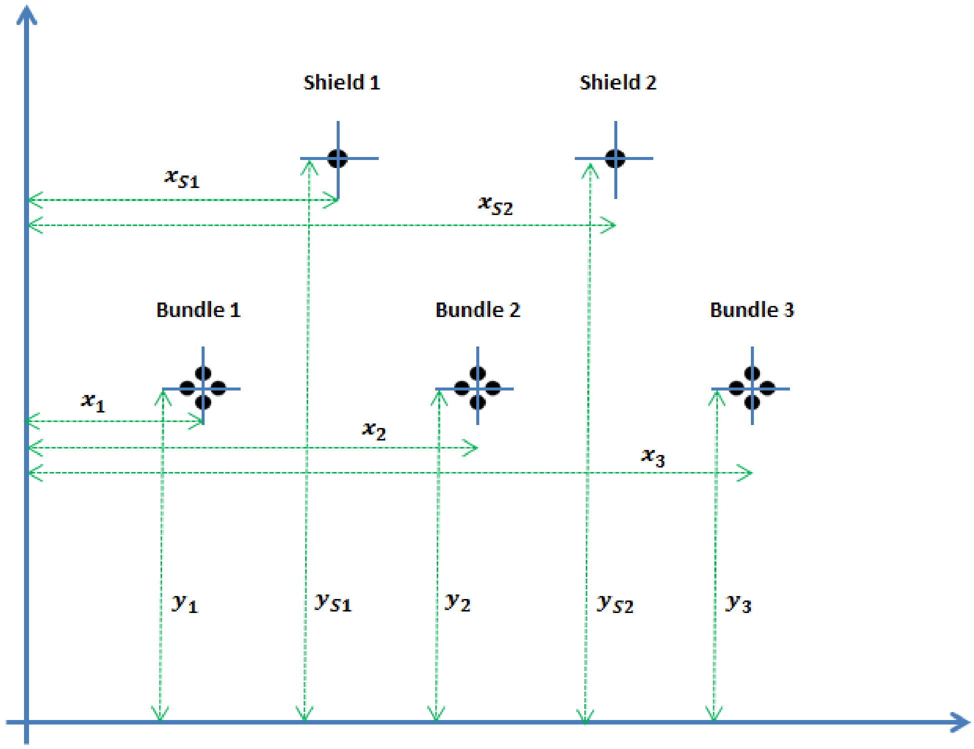

Figure 3.3-1: Geometrical Configuration of Line Used for Method Validation

	x_1	x_2	x_3	x_{S1}	x_{S2}	y_1	y_2	y_3	y_{S1}	y_{S2}
Distance [meters]	-10	0	10	-6	6	20	20	20	27	27

Table 3.3-2: Distances Associated with Geometrical Configuration

The transmission line was modeled using the Alternative Transients Program (ATP). The distributed, nominal pi-model was chosen with the auto-bundling and skin effect features selected. A summary of line properties required for modeling are listed in Table 3.3-3.

The resulting series impedance and shunt admittance matrices per unit length (Ω/ kilometer) are given in Table 3.3-4. The equivalent total series impedance and shunt admittance is found by multiplying each matrix by the length of the line.

Line Property	**Value**
Phase conductor radius	16 cm
Inner radius due to skin effect	0 cm
Phase conductor DC resistance	0.5 Ω/km
Shielding wire radius	4.9 cm
Shielding wire DC resistance	0.8 Ω/km
Ground resistance	0.001 Ω/km
Number of conductor per phase	4
Separation of conductors	20 cm
Length of transmission line	150 km
Operating Frequency	60 Hz

Table 3.3-3: Properties of Line modeled in ATP

Phase Impedance Matrix Per Unit Length (Ω/km)
$Z_{abc} = \begin{bmatrix} 14.117 + j42.531 & 1.671 + j8.592 & 1.319 + j4.356 \\ 1.671 + j8.592 & 14.424 + j42.156 & 1.671 + j8.592 \\ 1.319 + j4.356 & 1.671 + j8.592 & 14.117 + j42.531 \end{bmatrix} \times 10^{-2}$
Phase Admittance Matrix Per Unit Length (Ω/km)
$Y_{abc} = j \begin{bmatrix} 40.367 & -6.767 & -1.997 \\ -6.767 & 42.033 & -6.767 \\ -1.997 & -6.767 & 40.367 \end{bmatrix} \times 10^{-7}$

Table 3.3-4: Phase Impedance and Admittance Matrices Per Unit Length (given in ohms/kilometer)

Since real PMU measurement samples are not available for the line under consideration, ATP will be used to generate them. The samples are obtained by connecting a three phase voltage source to the sending end of the line model and a three phase load to the receiving end. The impedance of the load is varied and three phase voltage/current measurements are obtained from the *.lis simulation report until an adequate number of samples have been acquired. For validation, six samples will be used.

A balanced or unbalanced load can be connected to the line when creating the sample set. Each method should estimate the parameter values correctly regardless of the type of load present. In order to imitate the load on a real transmission line, an unbalanced load will be used for each sample.

MATLAB is used to implement each method and estimate the parameter values. The validation results are contained in Table 3.3-5, Table 3.3-6, and Table 3.3-7 for resistance, reactance, and susceptance respectively. Each table list the true parameter value along with the value estimated by the existing and proposed methods. In addition, the percent error of each parameter value is listed. Percent error is calculated according to equation (3.3.28). To avoid long strings of numbers in each Table, entries are limited to an appropriate number of digits.

$$\text{Percent Error} = \left(\frac{\text{True Value} - \text{Estimated Value}}{\text{True Value}}\right) \times 100 \qquad \textbf{(3.3.28)}$$

Parameter	True Value [ohms]	Existing Method [ohms]	Proposed Method [ohms]	% Error Existing Method	% Error Proposed Method
R_a	21.1765	21.1765	21.1765	0	0
R_b	21.6364	21.6364	21.6364	0	0
R_c	21.1765	21.1765	21.1765	0	0
R_{ab}	2.5065	2.5065	2.5065	0	0
R_{ac}	1.9792	1.9792	1.9792	0	0
R_{bc}	2.5065	2.5065	2.5065	0	0

Table 3.3-5: Validation Results for Resistance Values

Parameter	True Value [ohms]	Existing Method [ohms]	Proposed Method [ohms]	% Error Existing Method	% Error Proposed Method
$\boldsymbol{X_a}$	63.7975	63.7975	63.7975	0	0
$\boldsymbol{X_b}$	63.2339	63.2339	63.2339	0	0
$\boldsymbol{X_c}$	63.7975	63.7975	63.7975	0	0
$\boldsymbol{X_{ab}}$	12.8875	12.8875	12.8875	0	0
$\boldsymbol{X_{ac}}$	6.5341	6.5341	6.5341	0	0
$\boldsymbol{X_{bc}}$	12.8875	12.8875	12.8875	0	0

Table 3.3-6: Validation Results for Reactance Values

Parameter	True Value [mhos]	Existing Method [mhos]	Proposed Method [mhos]	% Error Existing Method	% Error Proposed Method
$\boldsymbol{B_a}$	0.6055 E-03	0.6055 E-03	0.6056 E-03	0	-0.0086
$\boldsymbol{B_b}$	0.6305 E-03	0.6305 E-03	0.6304 E-03	0	0.0166
$\boldsymbol{B_c}$	0.6055 E-03	0.6055 E-03	0.6056 E-03	0	-0.0086
$\boldsymbol{B_{ab}}$	-0.1015 E-03	-0.1015 E-03	-0.1018 E-03	0	-0.2703
$\boldsymbol{B_{ac}}$	-0.0300 E-03	-0.0300 E-03	-0.0299 E-03	0	0.1745
$\boldsymbol{B_{bc}}$	-0.1015 E-03	-0.1015 E-03	-0.1013 E-03	0	0.2421

Table 3.3-7: Validation Results for Susceptance Values

As can be seen from the validation results, each method is able to estimate all 18 parameter values with high accuracy. The results prove that the proposed method is a valid estimation technique that can be used to determine the parameters of an untransposed line.

4 Sensitivity Analysis for Untransposed Line Methods

In this Chapter, a sensitivity analysis is conducted with the untransposed line methods to determine their performance when noise is present in the synchrophasor measurements. Before the analysis is performed, the factors impacting the accuracy of parameter estimation will be discussed. At the end of the analysis, a conclusion section will summarize the observations made.

4.1 Factors Impacting Accuracy of Parameter Estimation

There are various factors that interfere with the ability of each method to accurately estimate line parameters values for an untransposed line with unbalanced load. Some of the more prevailing of these factors include:

- the influence of noise in the synchrophasor measurements
- the influence of mutual coupling between transmission lines
- the influence of the transmission line length

References [**7**] and [**13**] provided a detailed discussion of the role that mutual coupling and transmission line length play into the parameter estimation process. A summary of the conclusions found by these references follows.

Mutual coupling between transmission lines can be compensated for during parameter estimation by subtracting the voltage induced by the coupled line from the voltage measurements. The induced voltage can be calculated using the mutual inductance between the transmission lines of interest and the current measurements of the line inducing the voltage [**7**].

The transmission line length only becomes a factor in parameter estimation when noise is present in the synchrophasor measurements. The percentage error in line parameter values tends to be larger for short transmission lines. As the line length increases, error is still present in the line parameter values, however, its magnitude is greatly reduced.

Since the influences of mutual coupling and transmission line length have already been explored, these factors will not be considered in the sensitivity analysis. The transmission line used in the analysis will have fixed length and will not be coupled with another line. The analysis will focus exclusively on the type and magnitude of noise present in the synchrophasor measurements.

4.2 Types and Sources of Noise

There are two types of noise found in synchrophasor measurements – systematic and random. The presence of systematic and random noise in synchrophasor measurements can cause estimated parameter values to deviate from their true values. Even the smallest quantity of noise can produce large errors.

Systematic (bias) noise appears consistently with the same magnitude and direction in a measurement set. It causes measured values to be offset from their true values by a fixed amount. Systematic noise is often the result of improper use or mis-calibration of a measurement device. It can only be eliminated by finding the source of the noise and correcting it.

Random noise appears with variable magnitude and direction in a measurement set. It typically has a Gaussian normal distribution with the mean being the true value of the measured quantity. The cause of random noise is generally unknown. It can be eliminated by taking the average over a large data set.

Noise in synchrophasor measurements is mostly attributed to inaccuracies of the instrumentation channels and GPS equipment that constitute the overall phasor measurement system. These inaccuracies can be reduced as new technology takes the place of old, but will never completely be eliminated.

An instrumentation channel is defined as the devices in between the original high voltage or current measurement point and the signal that is passed to the A/D converter of a PMU [**14**]. The instrumentation channel is physically spread throughout an electrical substation. The instruments that makes up the channel include potential transformers, current transformers, signal cables, burdens, and attenuators. An illustration of an instrumentation channel is given in Figure 4.2-1.

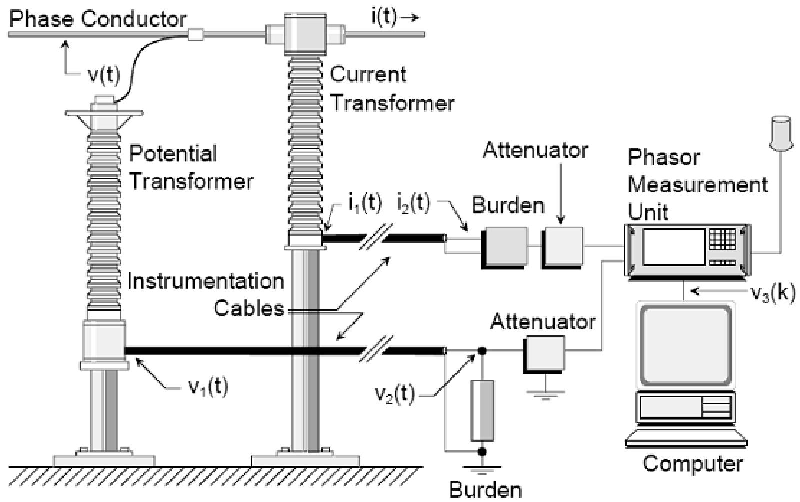

Figure 4.2-1: Illustration of PMU Instrumentation Channel [14**] North American SynchroPhasor Initiative Performance & Standards Task Team, "Synchrophasor Measurement Accuary Characterization," 2007. Used under fair use, 2015**

There are various problems that occur with the individual components of an instrumentation channel. These problems can cause inaccuracies in the synchrophasor measurements. Some examples include mis-calibration of instrument transformers, accumulation of contaminates on cable surfaces, aging of equipment, the presence of nonlinear burdens, and electrical interference from nearby circuitry.

The GPS equipment used to produce synchronized measurements has been known to have high accuracy. The best GPS systems can produce time tagging of measurements with precision better than 1 microsecond. On a system with an operating frequency of 60 Hz, this is equivalent to only 0.02 degrees error in phase [**14**].

The inaccuracy of GPS equipment can be dismissed when all PMUs on a given power system come from the same manufacturer. This is because the error produced by the GPS system is the same for each PMU. GPS inaccuracy becomes more significant in a multi-vendor environment (which is presently the case) [**14**]. PMUs from different vendors have different internal clocks which are subject to run faster or slower than universally coordinated time (UTC). Over time, this issue can result in time skew (drift) errors in the synchronized measurements. More information on the time skew problem as well as methods of correcting it can be found in reference [**15**].

4.3 Setup of Sensitivity Analysis

The sensitivity analysis will determine the performance of each estimation method (existing and proposed) when systematic and random noise is added to the synchrophasor measurements. The transmission line that will be used for the sensitivity analysis is the same line used during validation of the proposed line method in section 3.3.1. The length of the line is 150 km and it is not mutually coupled with other lines.

The systematic noise that will be added to the synchrophasor measurements will have a fixed magnitude and direction. The amount of noise present will be varied from negative to positive one percent of the true measured value. The random noise that will be added to the synchrophasor measurements will follow a Gaussian normal distribution. It will have zero mean and standard deviation that is varied from zero to positive one percent of the true measured value. While it is likely that both types of noise could appear together in real-world synchrophasor measurements, they will not be combined for the purposes of this analysis.

We will assume that for both types of noise, each phase of a measured quantity will be affected equally. For example, phases A,B, and C of the sending end voltage will all be corrupted by the same amount of noise. We will also assume that the real and imaginary parts of a measured quantity are affected equally. For example, the real and imaginary parts of the phase A sending end voltage will be corrupted by the same amount of noise.

Since each estimation method is least squared based, the following variables will need to be considered:

- The number of estimations performed. Depending on the noise present in the measurements, an estimation performed with one set of samples may produce a different result than estimation performed with another set of samples. Typically results from different estimations are averaged to obtain the parameter values that best represent the data set.
- The number of measurement samples used per estimation. A well-known benefit of least squares regression is that increasing the number of samples used to perform an estimation helps to reduce the impact of noise.

For systematic noise, these variables will have little influence over the accuracy of each estimation method. For random noise, these variables will have a large influence. This arises due to the nature and statistical characteristics of each type of noise.

For consistency purposes, the number of estimations and the number of measurement samples used per estimation will be kept the same regardless of the type of noise present. For a given type of noise and percent deviation, each estimation method will perform 500 estimations using 500 unique sets of measurements per estimation. After all estimations have been performed, the parameter values produced by the individual estimations will be averaged.

The sensitivity analysis will be conducted using MATLAB. A script was written that uses the transmission line parameter values generated by ATP to create unique PMU measurements reflecting the line under unbalanced load conditions. After the measurements are created, the user selects the type and magnitude of noise that should be added. The measurements are then passed to the existing and proposed methods where an estimation is obtained for the parameter values. This process is repeated until the desired number of estimations have been reached.

4.4 Sensitivity Analysis

In this section, the performance of each estimation method is evaluated when systematic noise and random noise is present in the synchrophasor measurements. A different simulation will be conducted for each of the five cases:

1. noise is present in the sending end voltage
2. "   " the sending end current
3. "   " in the receiving end voltage
4. "   " in the receiving end current
5. "   " in all measurements

For each case, the average percent error in the estimated parameter values with respect to the true parameter values will be reported. A total of three plots will be produced for each estimation method. The first plots displays the error in resistance values vs. the error in the measurements. The second displays the error in the reactance values vs. the error in the measurements. The third displays the error in the susceptance values vs. the error in the measurements. Since the legends for each of these plots do not change from simulation to simulation, they will be listed in this section to avoid clutter. The legends are given in Table 4.4-1:

Resistance Plots	**Reactance Plots**	**Susceptance Plots**
Ra Rb Rc Rab Rac Rbc	Xa Xb Xc Xab Xac Xbc	Ba Bb Bc Bab Bac Bbc

Table 4.4-1: Plot Legends for Sensitivity Analysis

4.4.1 Systematic Noise Performance

The performance of the estimation methods when systematic noise was added to the measurement samples is presented here. There are a total of five figures that display the simulation results –one for each of the cases described at the start of this section.

A summary of observations made from these figures can be found below. Note that an estimated parameter value is considered “sensitive” to the systematic noise when there is more than ten percent deviation from the true value.

Existing Method Observations

- The self-resistances and self-reactances are sensitive to noise in the sending and receiving end voltage measurements.
- The existing method produces accurate results for all parameter values for all other cases.

Proposed Method Observations

- The susceptance parameters (both self and mutual) are sensitive to noise in the sending and receiving end voltage measurements.
- The proposed method produces accurate results for all parameter values for all other cases.

For both methods, there appears to be a linear relationship between the percent error in measurements and the percent error in the parameter values. The greater the sensitivity to the systematic noise, the greater the magnitude of the slope.

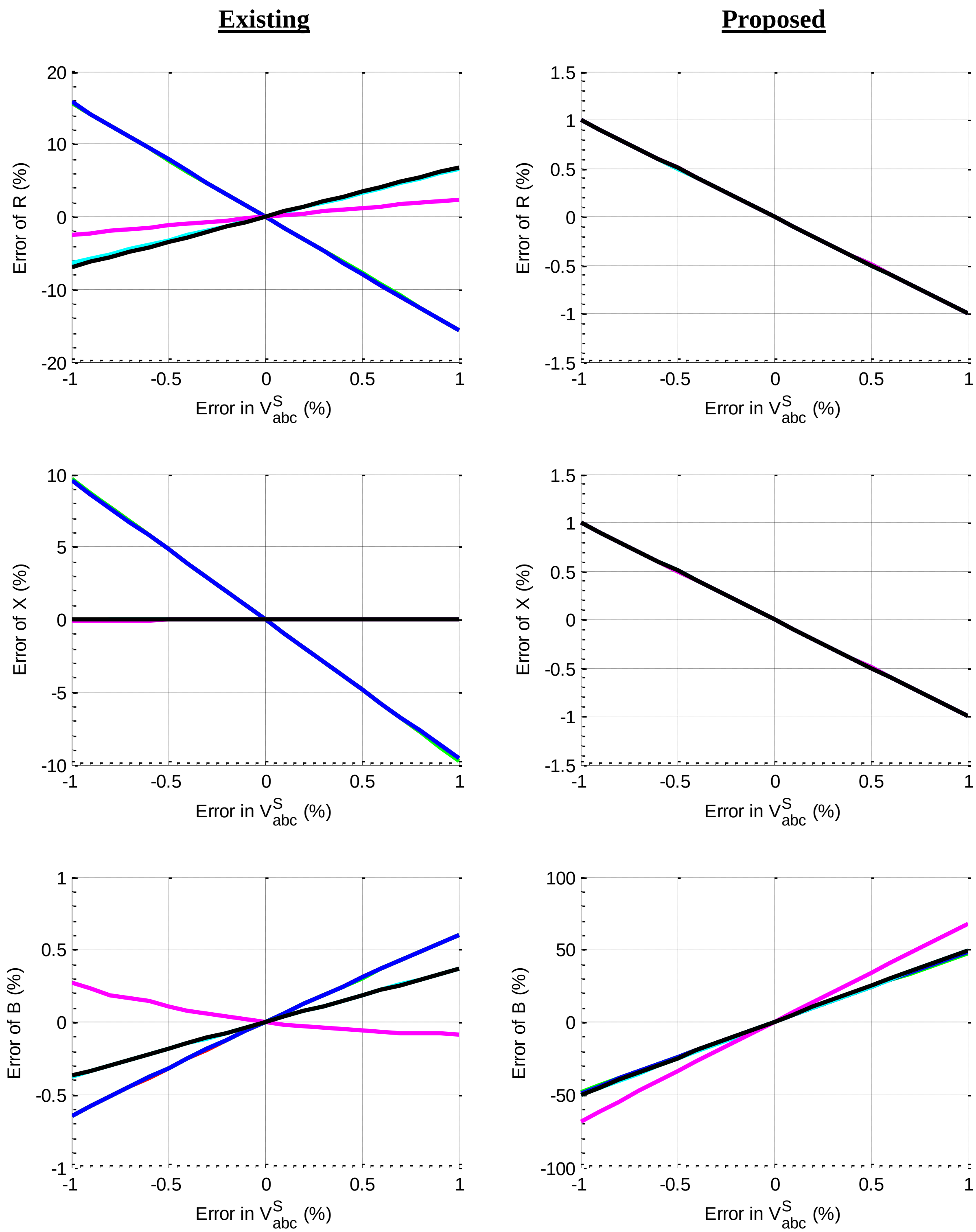

Figure 4.4-1: Performance when Systematic Noise is present in the Sending End Voltage

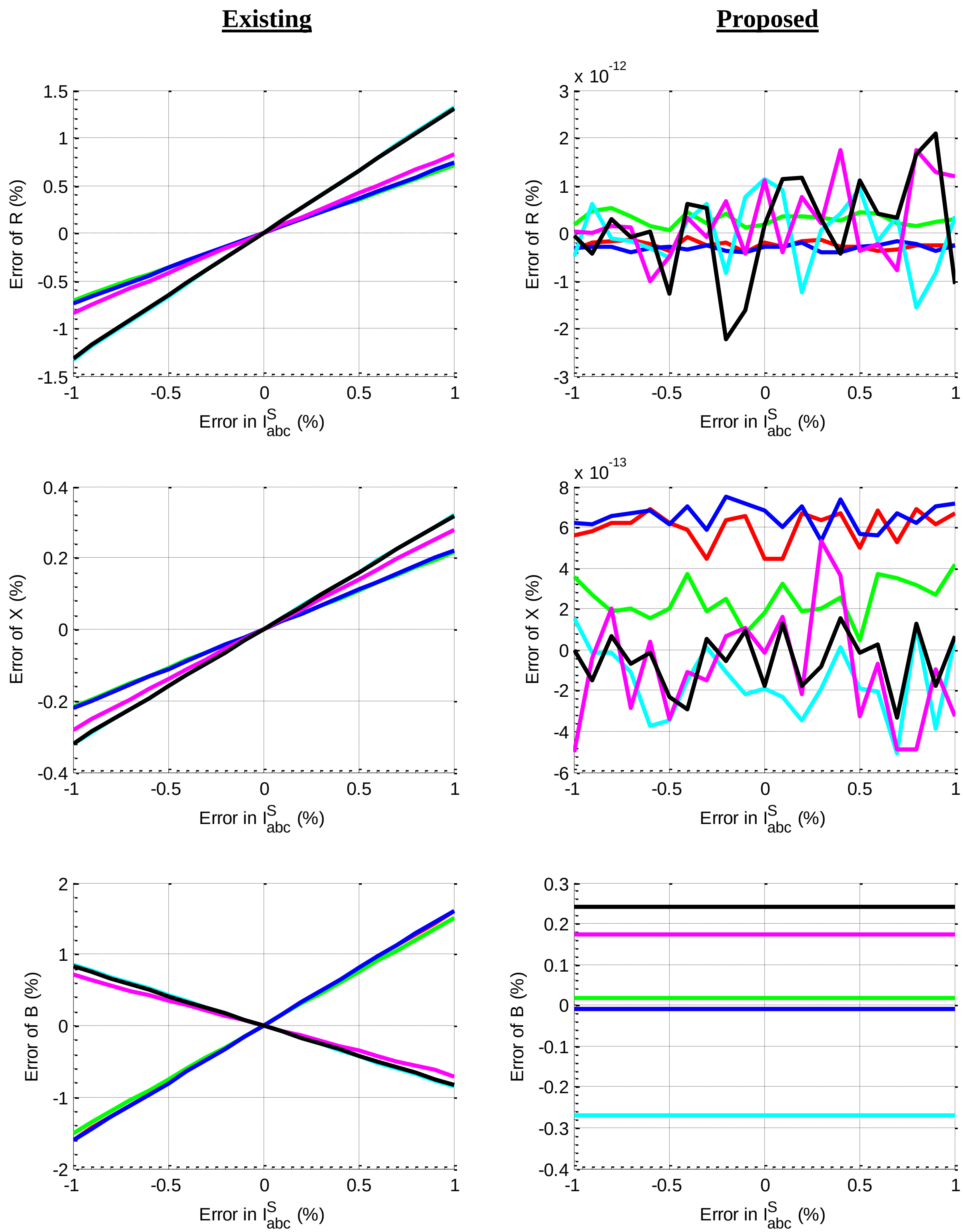

Figure 4.4-2: Performance when Systematic Noise is present in the Sending End Current

Existing **Proposed**

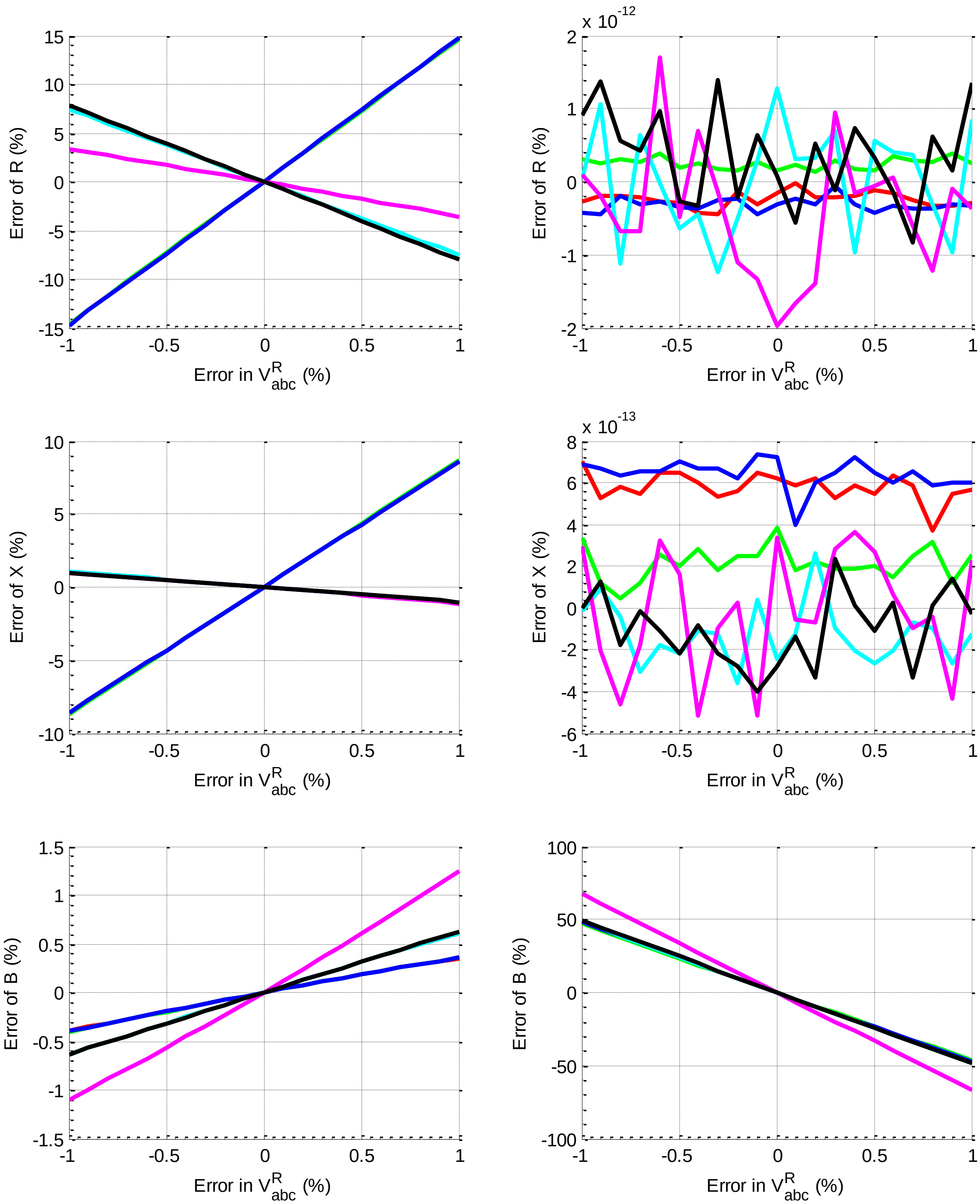

Figure 4.4-3: Performance when Systematic Noise is present in the Receiving End Voltage

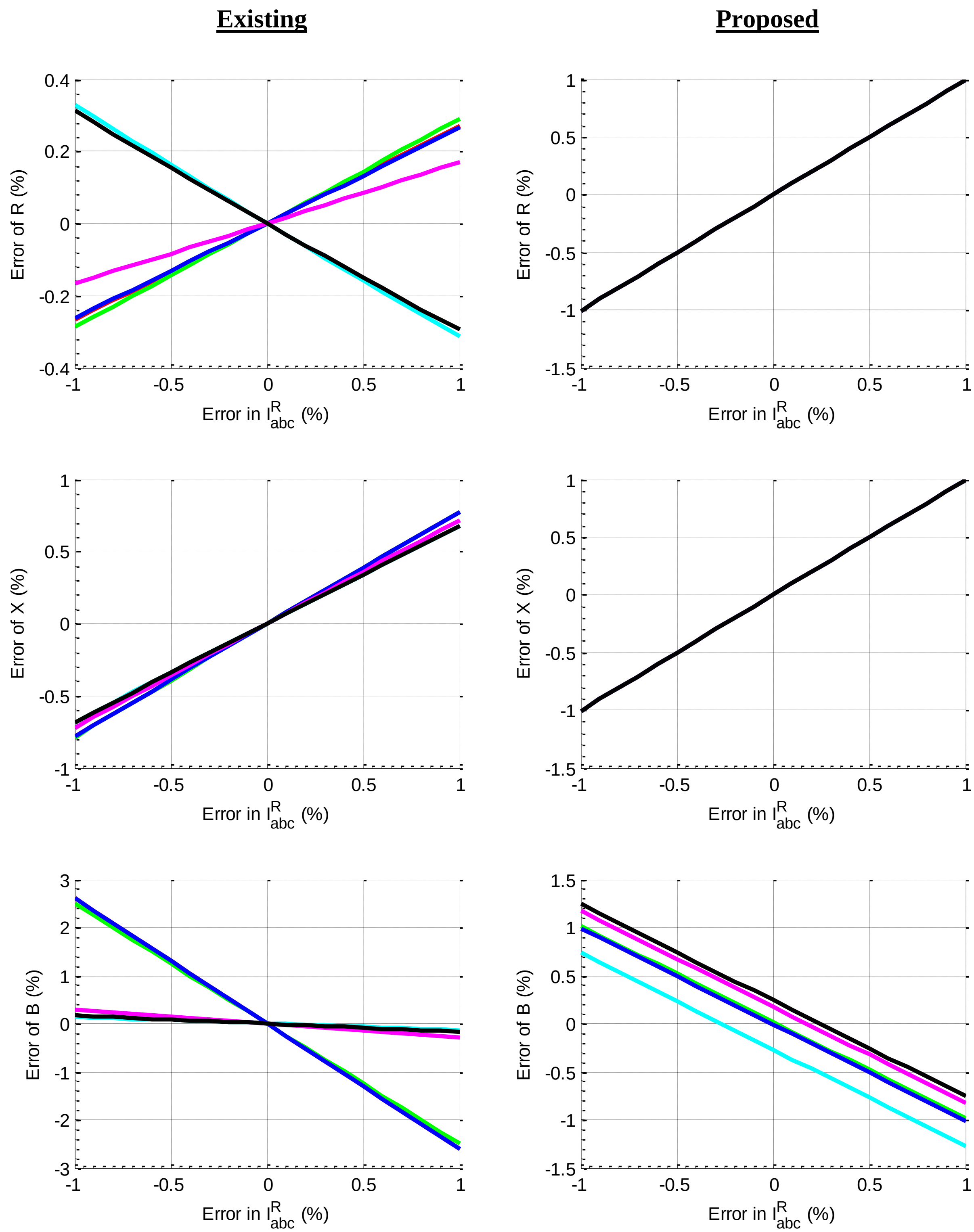

Figure 4.4-4: Performance when Systematic Noise is present in the Receiving End Current

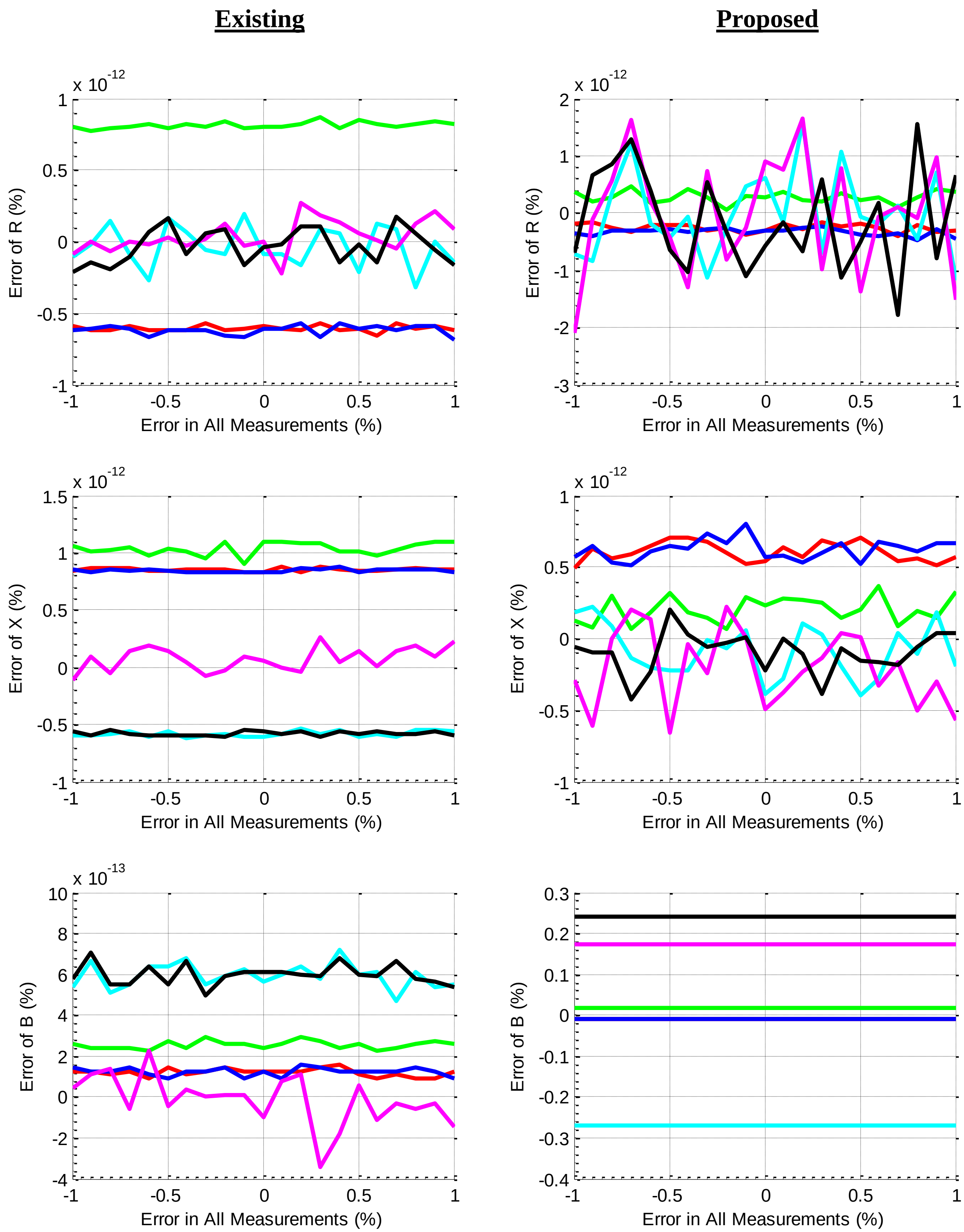

Figure 4.4-5: Performance when Systematic Noise is present in the All Measurements

4.4.2 Random Noise Performance

The performance of the estimation methods when random noise was added to the measurement samples is presented here. There are a total of five figures that display the simulation results – one for each of the cases described at the start of this section.

A summary of observations made from these figures can be found below. Note that an estimated parameter value is considered "sensitive" to the random noise when there is more than ten percent deviation from the true value.

Existing Method Observations

- The mutual-resistances are sensitive to noise in the sending and receiving end voltage measurements. Some mutual-reactances display minor sensitivity.
- The existing method produces accurate results for all parameter values for all other cases.

Proposed Method Observations

- The susceptance parameters (both self and mutual) are sensitive to noise in the sending and receiving end voltage measurements.
- The mutual-susceptance parameters are sensitive when there is noise in all measurements. The self-susceptance parameters display minor sensitivity.
- The proposed method produces accurate results for all parameter values for all other cases.

The relationship between the standard deviation of the noise added to the measurements and the percent error in the parameter values is unclear. For both methods, the magnitude of the percent error tends to increase with increasing standard deviation in noise.

The curves produced by the proposed method are not as smooth as those produced by the existing method. This indicates that there is greater variance amongst the parameter estimations performed by the proposed method.

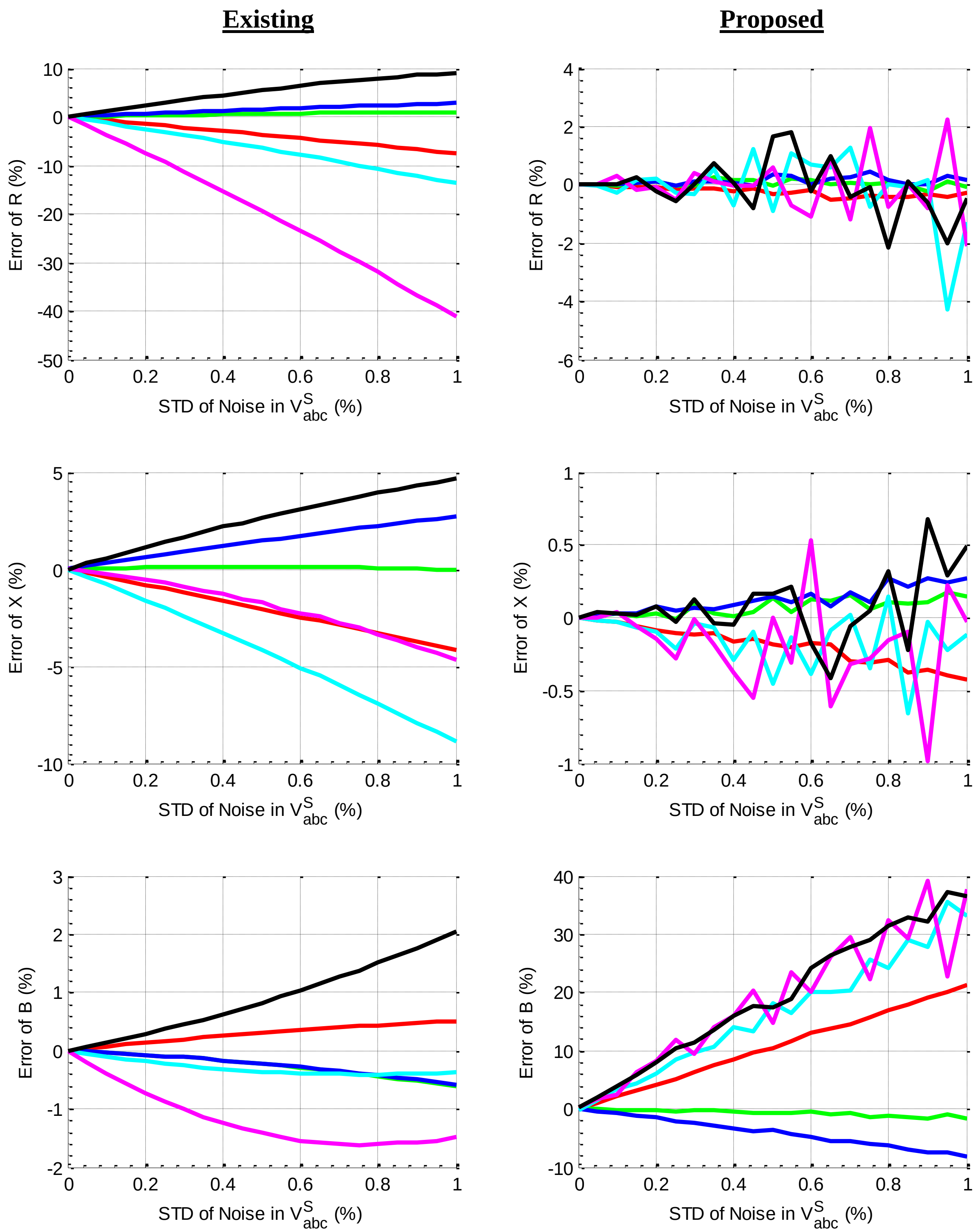

Figure 4.4-6: Performance when Random Noise is present in the Sending End Voltage

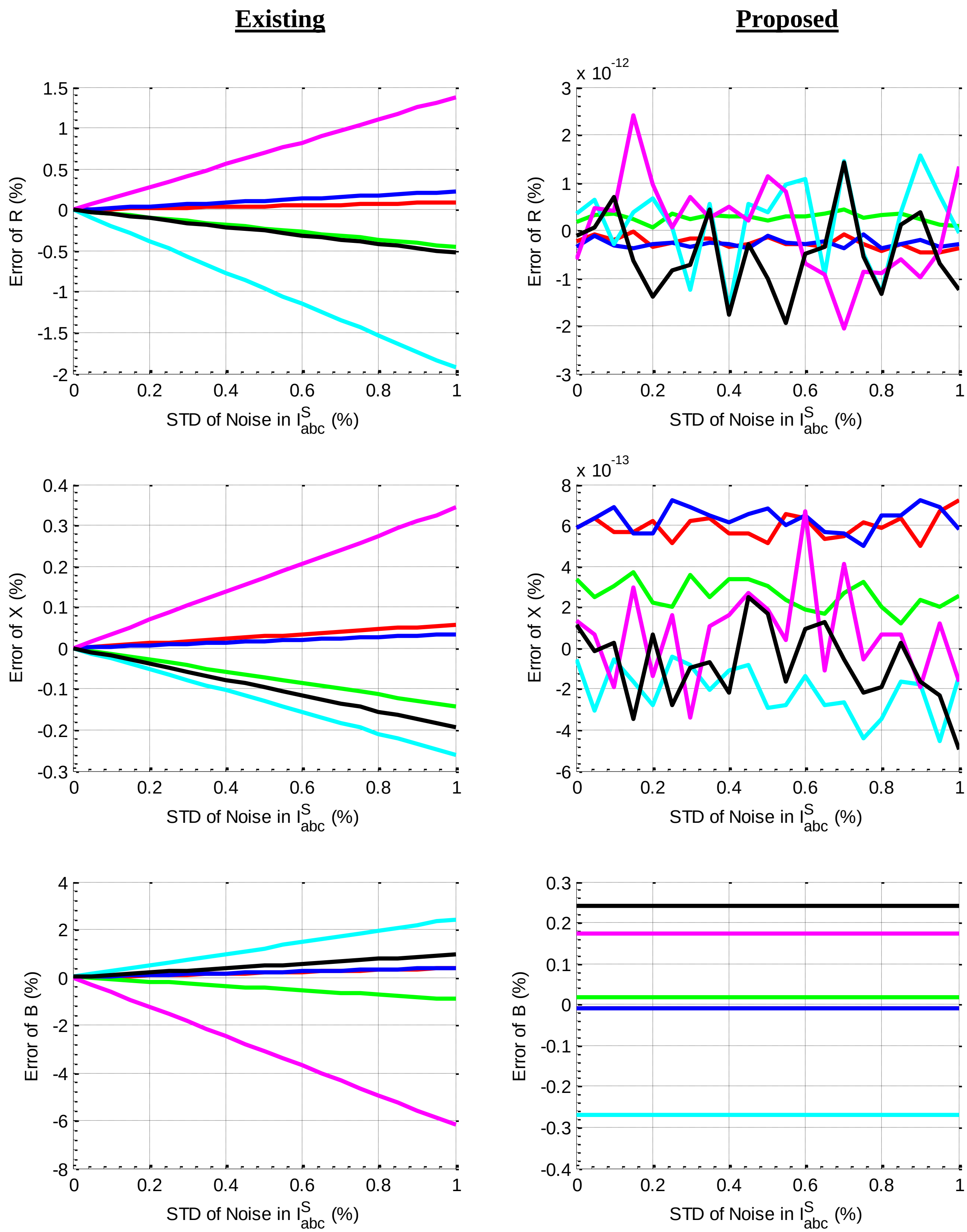

Figure 4.4-7: Performance when Random Noise is present in the Sending End Current

Existing **Proposed**

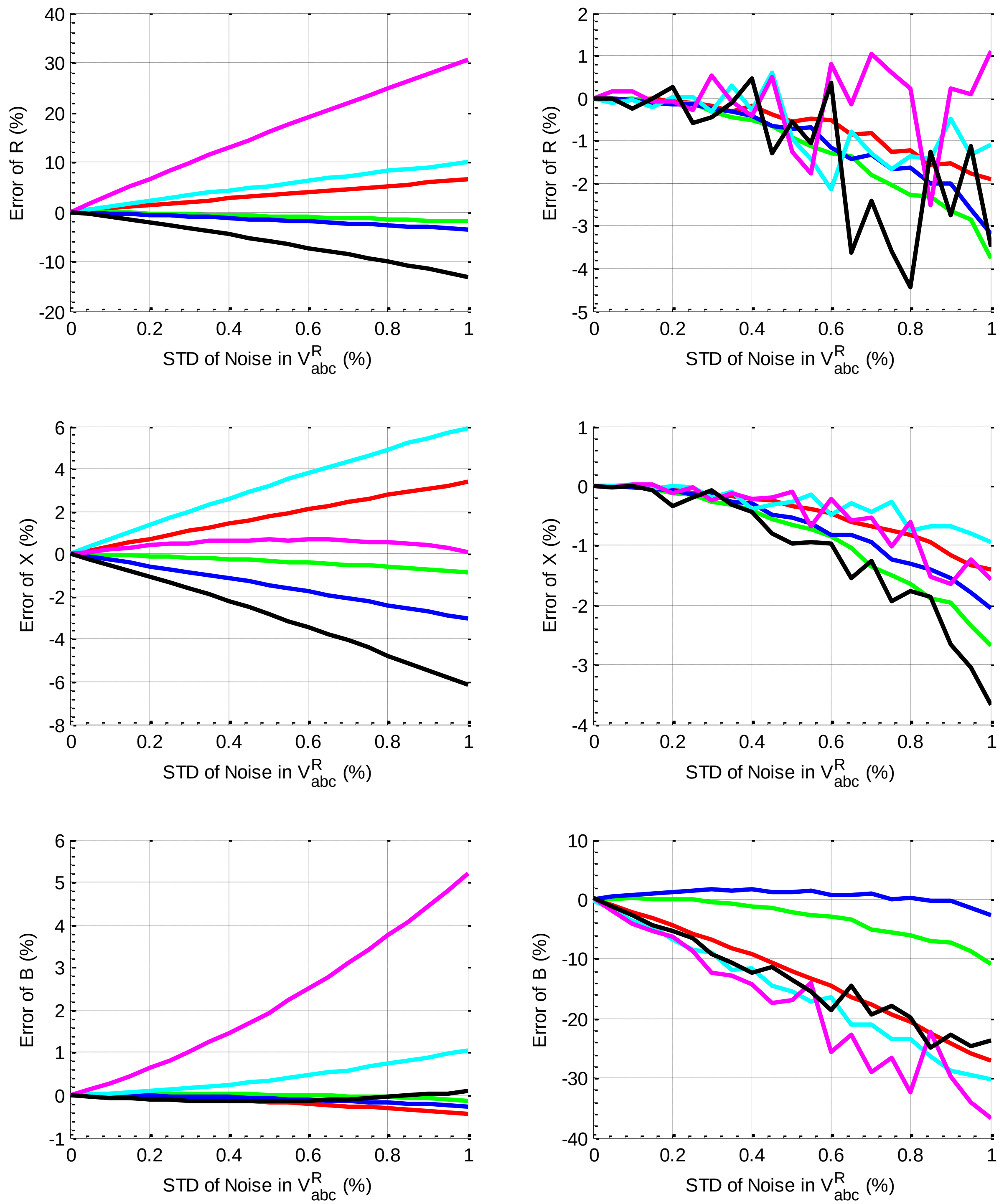

Figure 4.4-8: Performance when Random Noise is present in the Receiving End Voltage

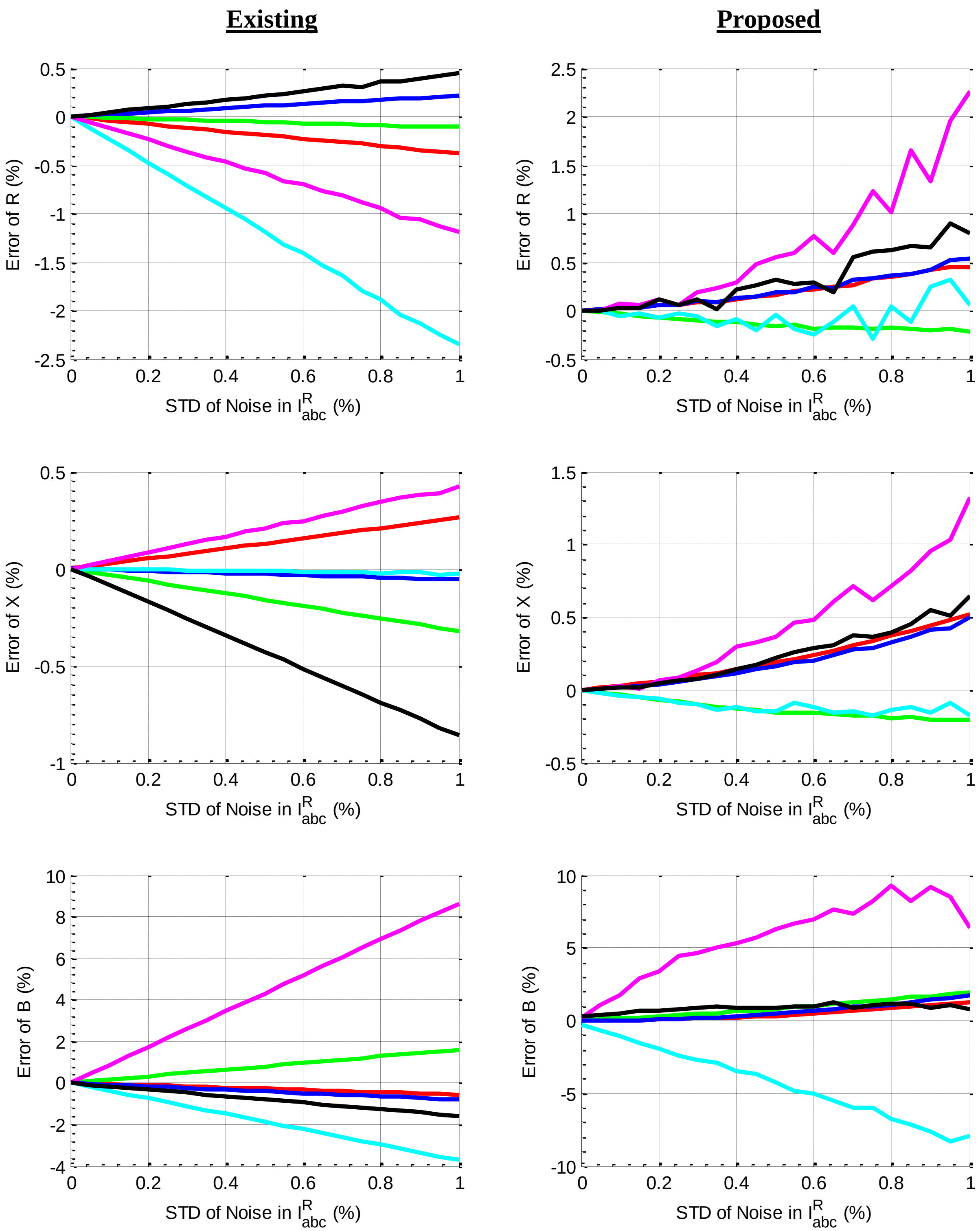

Figure 4.4-9: Performance when Random Noise is present in the Receiving End Current

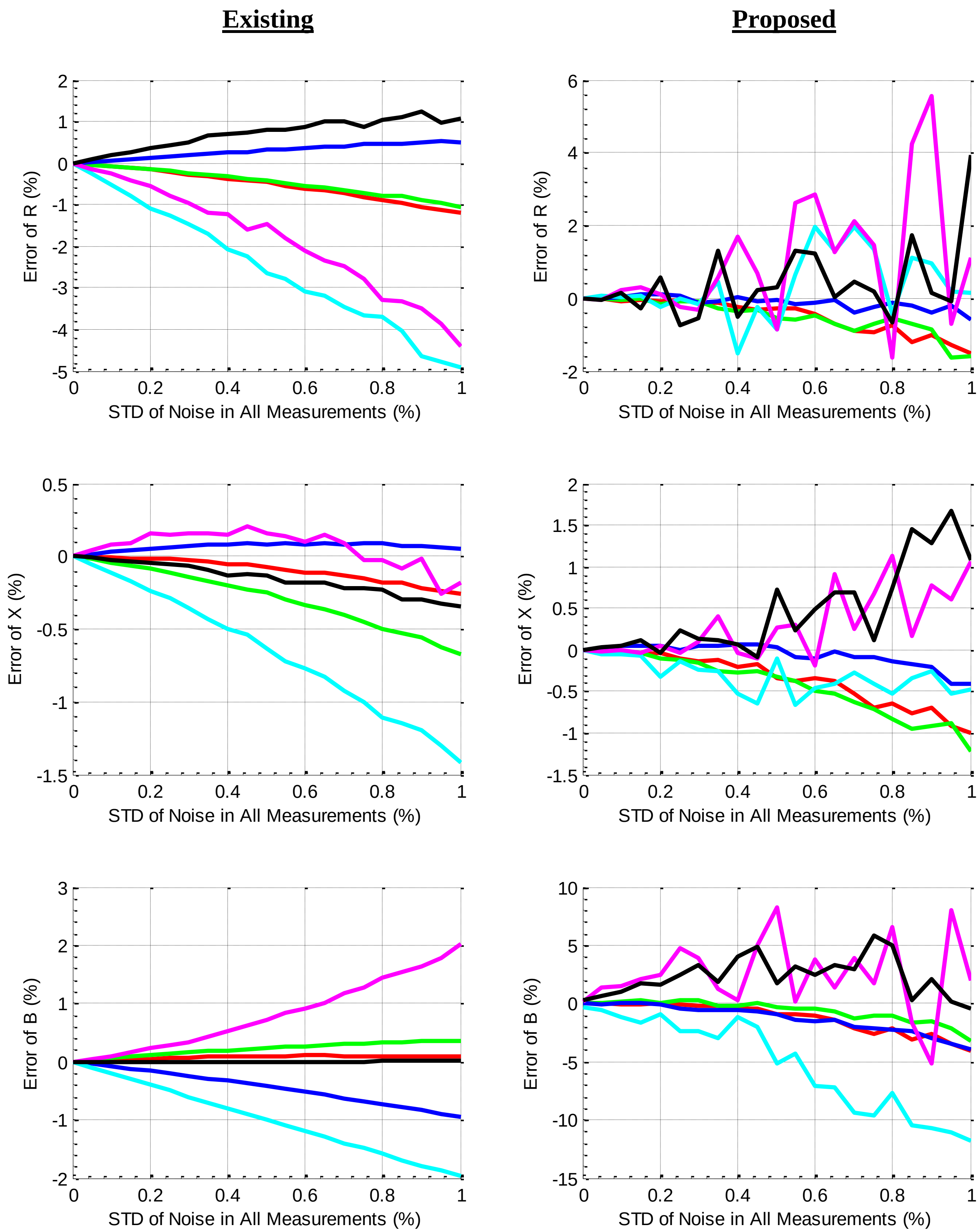

Figure 4.4-10: Performance when Random Noise is present in the All Measurements

4.5 Summary of Results

We have seen through the results of the sensitivity analysis that noisy measurements can have a considerable impact on the accuracy of the untransposed line estimation methods. The figures above show that noise was primarily a factor when it was present in either the sending or receiving end voltage. There was very little deviation in parameter values when it was present in the sending and receiving end currents.

To provide the reader with another way of evaluating the performance of the estimation methods, Table 4.5-2 and Table 4.5-3 were created. These tables are Heatmaps that display the maximum percent error in a parameter value over the simulation range (negative to positive one percent) of error in the measurements (systematic noise) or standard deviation in the measurements (random noise). The key for each Heatmap is located below:

Maximum Percent Error in Parameter Values		
Less than 5%	**Between 5% and 10%**	**Greater than 10%**

Table 4.5-1: Heatmap Key

	Measurement with Systematic Noise									
	Existing Method					**Proposed Method**				
	V^S_{abc}	I^S_{abc}	V^R_{abc}	I^R_{abc}	**ALL**	V^S_{abc}	I^S_{abc}	V^R_{abc}	I^R_{abc}	**ALL**
R_a										
R_b										
R_c										
R_{ab}										
R_{ac}										
R_{bc}										
X_a										
X_b										
X_c										
X_{ab}										
X_{ac}										
X_{bc}										
B_a										
B_b										
B_c										
B_{ab}										
B_{ac}										
B_{bc}										

Table 4.5-2: Systematic Noise Heatmap

	Measurement with Random Noise									
	Existing Method					Proposed Method				
	V^S_{abc}	I^S_{abc}	V^R_{abc}	I^R_{abc}	ALL	V^S_{abc}	I^S_{abc}	V^R_{abc}	I^R_{abc}	ALL
R_a										
R_b										
R_c										
R_{ab}										
R_{ac}										
R_{bc}										
X_a										
X_b										
X_c										
X_{ab}										
X_{ac}										
X_{bc}										
B_a										
B_b										
B_c										
B_{ab}										
B_{ac}										
B_{bc}										

Table 4.5-3: Random Noise Heatmap

After comparing the performance of each estimation method, we can make the following statements:

- The proposed method produces more accurate results for the resistance parameters.
- The proposed method produces more accurate results for the reactance parameters.
- The existing method produces more accurate results for the susceptance parameters.

These statements generally hold true regardless of the type (systematic or random) or the location (sending/receiving end voltage or current) of the noise added to the synchrophasor measurements. There are, of course, a few minor expectations. These statements can be validated by observing the Heatmaps. We see that for the resistance and reactance parameters there are several more yellow and red squares in the existing method quadrants as compared to the proposed method quadrants. For the susceptance parameters there are several more yellow and red squares in the proposed method quadrants as compared to the existing method quadrants.

It is the author's belief that the most accurate results for the line parameters of an untransposed line can be achieved by using a combination of both estimation methods. This is given that the type and location of noise is unbeknown to the user (which is normally the case) and that there is ample measurement samples available for estimation. The new hybrid method would estimate the resistance and reactance parameters using the estimation technique described by the proposed method. It would estimate the susceptance parameters using the estimation techniques described by the existing method.

5 Conclusion and Future Work

5.1 Conclusion

In this document, the author has presented a new method of estimating the transmission line parameters of an untransposed line. In order to validate this method an overhead, untransposed, three phase line was modeled in ATP. Three phase voltage and current measurements were acquired from the sending and receiving ends of the line by simulating different loading conditions. The measurements were applied to the method and the resulting estimated parameters were compared against the ATP parameters values. There was less than one percent error for each of the 18 unknown parameters.

After the proposed untransposed method was validated, a sensitivity analysis was conducted to evaluate the performance of all untransposed methods when noise was present in the measurements. The two types of noise that were considered for the analysis were systematic and random. Systematic noise appeared with the same magnitude and direction in the affected quantities whereas random noise followed a Gaussian normal distribution. To simplify the analysis, each type of noise was considered separately. A different simulation was carried out for when noise was present in the sending end voltage, the sending end current, the receiving end voltage, the receiving end current, and for all measurements. Each phase of the measured quantities received the same amount of noise.

The percent error in the parameter values were plotted against the percent error in the measurements and the standard deviation of noise for systematic and random noise respectively. The analysis found that the estimation methods are primarily sensitive to errors in the sending and receiving end voltages. The proposed untransposed method provided better estimates for the resistance and reactance parameters. The existing untransposed method provided better estimates for the susceptance parameters. For future applications, it was suggested that the best results for an untransposed line parameter estimation can be achieved by using a combination of both methods.

5.2 Future Work

The research performed by this thesis can be extended by implementing other variations of the sensitivity analysis conducted in Chapter 4. The variations of the analysis would further evaluate the performance of the untransposed methods when noise is present in the PMU measurements. Possible variations to the sensitivity analysis are listed below:

- Repeat the analysis for both types of noise but for the case where each phase of a measured quantity is perturbed by a different amount of noise.
- Repeat the analysis for both types of noise but for the case where the real and imaginary parts of a measured quantity are perturbed by a different amount of noise.
- Expand the analysis to include a different simulation for each phase of the sending and receiving end voltage and current quantities.
- Expand the analysis to investigate how combining systematic and random noise might impact the performance of the estimation methods.
- Expand the analysis to find the percent error in both the phase and sequence parameters.

Other future research efforts might focus on the application of real PMU measurements to the estimation methods. In this scenario, the influence of the factors listed in section 4.1 may need to be taken into consideration. In order to test the accuracy of the estimated parameter values, they may be compared with one of the line model verification methods discussed by reference [**16**]. Some of these methods include the performance of protection relays, the performance of fault locating algorithms, and direct measurement of the parameter values using an injection test set.

References

[1] Masoud Farzaneh, Shahab Farokhi, and William Chisholm, *Electrical Design of Overhead Power Transmission Lines*, Molly Wyand, Ed.: McGraw Hill, 2013.

[2] John Grainger and William Stevenson Jr., *Power System Analysis*.: McGraw-Hill, 1994.

[3] Stanley H. Horowitz and Arun G. Phadke, *Power System Relaying*.: John Wiley and Sons Ltd, 2014.

[4] Jonas Traphoner, "Evaluation of Impedance Parameters in Transmission Lines," The University of Texas at Austin, Master's Thesis 2014.

[5] J Duncan Glover, Mulukutla S Sarma, and Thomas Overbye, *Power System Analysis and Design*, 5th ed.: Cengage Learning, 2012.

[6] Martin Grobler and Raj Naidoo, "Determining Transmission Line Parameters from GPS Time-Stamped Data," in *IEEE Industrial Electronics, IECON 2006*, Paris, 2006, pp. 3580-3584.

[7] Di Shi, "Utilizing Synchrophasor Technology to Determine Transmission Line Impedance Parameters," Arizona State University, Master's Thesis 2009.

[8] Xun Tangsheng et al., "Study on Online Calculation Method of Transmission Line Parameters," in *Developments in Power Systems Protection*, Birmingham, UK, 2012, pp. 1-4.

[9] Robert Wilson, Zevenbergen Gary, and Daniel Mah, "Calculation of Transmission Line Parameters From Synchronized Measurements," *Electric Machines and Power Systems,* vol. 27, no. 12, pp. 1269-1278, December 1999.

[10] Tianshu Bi, Jinmeng Chen, Jingtao Wu, and Qixun Yang, "Synchronized Phasor based On-line Parameter Identification of Overhead Transmission Line," in *Electric Utility Deregulation and Restructing and Power Technologies*, Nanjuing, 2008, pp. 1657-1662.

[11] Di Shi, Daniel Tylavsky, Naim Logic, and Kristian Koellner, "Identification of Short Transmission-Line Parameters from Synchrophasor Measurements," in *North American Power Symposium*, Calgary, 2008, pp. 1-8.

[12] C.S. Indulkar and K. Ramalingam, "Estimation of Transmission Line Parameters from Measurements," *Electrical Power and Energy Systems*, pp. 337-342, June 2008.

[13] Di Shi, Daniel Tylavasky, Kristian Koellner, Naim Logic, and David Wheeler, "Transmission Line Parameter Identification using PMU Measurements," *European Transactions on Electrical Power*, no. 21, pp. 1574-1588, November 2011.

[14] North American SynchroPhasor Initiative Performance & Standards Task Team, "Synchrophasor Measurement Accuary Characterization," 2007.

[15] Qing Zhang et al., "The Time Skew Problem in PMU Measurements," in *Power and Energy Society General Meeting*, San Diego, 2012.

[16] Working Group D6, "AC Transmission Line Model Parameter Validation," A Report to the Line Protection Subcommittee of the Power System Relay Committee of the IEEE Power and Energy Society, 2014.

Appendix A: Sensitivity Analysis MATLAB Code

Appendix A.1 – Main Script

```
clear all
close all

%This script estimates the Transmission Line Parameters of an
%untransposed Line using the methods proposed by Di-Shi & Lowe. A
%sensitivity analysis is conducted to evaluate the performance of the
%untransposed methods when noise is present in the PMU measurements

%There are a total of three functions that are called upon in this script.
%The names and purpose of these functions are as follows:
%
% TL_parameters -- contains the distributed transmission line parameter
% values as calculated by ATP. The lumped parameter values are found by
% multiplying by a user selected line length
%
%create_measurements -- this function creates a set of PMU measurements
%using the transmission line parameter values. It also adds noise to the
%measurements based on selections made by the user
%
%analyze -- this function analyzes the results of the parameter
%estimations. It finds the mean, standard deviation, and percent deviation
%for every parameter. This information is passed back to the script for
%plotting purposes

%BEGIN--------------------------------------------------------------------


%enter the number of successive estimations that are performed
%take the mean/std of the result to use for analysis
num_est = 1;

%enter number of measurements to be used per estimation
num_meas = 500;

%enter type of noise present in measurements (1 = bias, 2 = random)
type = 1;

%enter the percent deviation in noise
% PSD = (0:0.001:0.02)';
PSD = 0;

%enter which measurement has the noise
%1 = VS   %2 = IS   %3 = VR   %4 = IR
%5 = ALL
pick = 5;

%enter whether or not data should be recorded ( 1 = yes,  2 = no)
write = 2;
```

```
%initialize matrices that hold parameters value for each estimation
Shi_R = zeros(num_est,6);
Shi_X = zeros(num_est,6);
Shi_B = zeros(num_est,6);

Lowe_R = zeros(num_est,6);
Lowe_X = zeros(num_est,6);
Lowe_B = zeros(num_est,6);

deter = zeros(num_est, 1);

%initialize matrices that hold percent deviation values for plotting
Shi_PDR = zeros(length(PSD),6);
Shi_PDX = zeros(length(PSD),6);
Shi_PDB = zeros(length(PSD),6);

Lowe_PDR = zeros(length(PSD),6);
Lowe_PDX = zeros(length(PSD),6);
Lowe_PDB = zeros(length(PSD),6);

detmn = zeros(length(PSD),1);

%ESTIMATE-----------------------------------------------------------------
for m = 1:length(PSD)

for e = 1:num_est
%obtain PMU measurements
[VabcS, IabcS, VabcR, IabcR] =
create_measurements(num_meas,type,PSD(m,1),pick);

%initalize loop variables
k = 0;  %counter variable
X = zeros(num_meas*6,36);
Y = zeros(num_meas*6,1);
S = zeros(num_meas*12,18);
W = zeros(num_meas*12,1);

for n = 1:num_meas
    %assign receiving end voltages and current values
    Var = VabcR(n,1);
    Vbr = VabcR(n,2);
    Vcr = VabcR(n,3);

    Iar = IabcR(n,1);
    Ibr = IabcR(n,2);
    Icr = IabcR(n,3);

    Vas = VabcS(n,1);
    Vbs = VabcS(n,2);
    Vcs = VabcS(n,3);
```

```
    Ias = IabcS(n,1);
    Ibs = IabcS(n,2);
    Ics = IabcS(n,3);

    %index variables increment to fill coeff. & measurement matrices
    %(Lowe Method)
    index1 = 1 + k*6;
    index2 = 2 + k*6;
    index3 = 3 + k*6;
    index4 = 4 + k*6;
    index5 = 5 + k*6;
    index6 = 6 + k*6;

    %index variables (Di-Shi Method)
    ind1 = 1 + k*12;
    ind2 = 2 + k*12;
    ind3 = 3 + k*12;
    ind4 = 4 + k*12;
    ind5 = 5 + k*12;
    ind6 = 6 + k*12;
    ind7 = 7 + k*12;
    ind8 = 8 + k*12;
    ind9 = 9 + k*12;
    ind10 = 10 + k*12;
    ind11 = 11 + k*12;
    ind12 = 12 + k*12;

    %fill coeff. and measurement matrices (Lowe Method)
    X(index1,:) = [Var Vbr Vcr zeros(1,6) Iar Ibr Icr zeros(1,24)];
    X(index2,:) = [zeros(1,3) Var Vbr Vcr zeros(1,6) Iar Ibr Icr
zeros(1,21)];
    X(index3,:) = [zeros(1,6) Var Vbr Vcr zeros(1,6) Iar Ibr Icr
zeros(1,18)];
    X(index4,:) = [zeros(1,18) Var Vbr Vcr zeros(1,6) Iar Ibr Icr
zeros(1,6)];
    X(index5,:) = [zeros(1,21) Var Vbr Vcr zeros(1,6) Iar Ibr Icr
zeros(1,3)];
    X(index6,:) = [zeros(1,24) Var Vbr Vcr zeros(1,6) Iar Ibr Icr];

    Y(index1,1) = Vas;
    Y(index2,1) = Vbs;
    Y(index3,1) = Vcs;
    Y(index4,1) = Ias;
    Y(index5,1) = Ibs;
    Y(index6,1) = Ics;

    %fill coeff. and measurement matrices (Di-Shi)
S(ind1,:)  = [ real(Vas) - real(Var), imag(Var) - imag(Vas),
0,                      0,                      0,                      0,
real(Vbs) - real(Vbr), imag(Vbr) - imag(Vbs), real(Vcs) - real(Vcr),
imag(Vcr) - imag(Vcs),                      0,                      0,
imag(Var)/2,                            0,                            0,
imag(Vbr)/2,                imag(Vcr)/2,                             0];
```

```
S(ind2,:)  = [                      0,                      0, real(Vbs) -
real(Vbr), imag(Vbr) - imag(Vbs),                    0,
0, real(Vas) - real(Var), imag(Var) - imag(Vas),                      0,
0, real(Vcs) - real(Vcr), imag(Vcr) - imag(Vcs),                            0,
imag(Vbr)/2,                        0,                  imag(Var)/2,
0,                 imag(Vcr)/2];


S(ind3,:)  = [                      0,                      0,
0,                      0, real(Vcs) - real(Vcr), imag(Vcr) - imag(Vcs),
0,                      0, real(Vas) - real(Var), imag(Var) - imag(Vas),
real(Vbs) - real(Vbr), imag(Vbr) - imag(Vbs),                          0,
0,                 imag(Vcr)/2,                          0,
imag(Var)/2,                 imag(Vbr)/2];


S(ind4,:)  = [                      0,                      0,
0,                      0,                      0,                      0,
0,                      0,                      0,                      0,
0,                      0, - imag(Var)/2 - imag(Vas)/2,
0,                            0, - imag(Vbr)/2 - imag(Vbs)/2, - imag(Vcr)/2 -
imag(Vcs)/2,                           0];


S(ind5,:)  = [                      0,                      0,
0,                      0,                      0,                      0,
0,                      0,                      0,                      0,
0,                      0,                           0, - imag(Vbr)/2 -
imag(Vbs)/2,                           0, - imag(Var)/2 - imag(Vas)/2,
0, - imag(Vcr)/2 - imag(Vcs)/2];


S(ind6,:)  = [                      0,                      0,
0,                      0,                      0,                      0,
0,                      0,                      0,                      0,
0,                      0,                           0,
0, - imag(Vcr)/2 - imag(Vcs)/2,                            0, - imag(Var)/2 -
imag(Vas)/2, - imag(Vbr)/2 - imag(Vbs)/2];


S(ind7,:)  = [ imag(Vas) - imag(Var), real(Vas) - real(Var),
0,                      0,                      0,                      0,
imag(Vbs) - imag(Vbr), real(Vbs) - real(Vbr), imag(Vcs) - imag(Vcr),
real(Vcs) - real(Vcr),                   0,                       0,
-real(Var)/2,                       0,                           0,
-real(Vbr)/2,               -real(Vcr)/2,                           0];


S(ind8,:)  = [                      0,                      0, imag(Vbs) -
imag(Vbr), real(Vbs) - real(Vbr),                    0,
0, imag(Vas) - imag(Var), real(Vas) - real(Var),                      0,
0, imag(Vcs) - imag(Vcr), real(Vcs) - real(Vcr),                            0,
-real(Vbr)/2,                       0,                 -real(Var)/2,
0,                -real(Vcr)/2];
```

```
S(ind9,:)  = [                   0,                   0,
0,                   0, imag(Vcs) - imag(Vcr), real(Vcs) - real(Vcr),
0,                   0, imag(Vas) - imag(Var), real(Vas) - real(Var),
imag(Vbs) - imag(Vbr), real(Vbs) - real(Vbr),                         0,
0,               -real(Vcr)/2,                        0,                 -
real(Var)/2,             -real(Vbr)/2];


S(ind10,:) = [                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,   real(Var)/2 + real(Vas)/2,
0,                        0,   real(Vbr)/2 + real(Vbs)/2,   real(Vcr)/2 +
real(Vcs)/2,                         0];


S(ind11,:) = [                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,                        0,   real(Vbr)/2 +
real(Vbs)/2,                         0,   real(Var)/2 + real(Vas)/2,
0,   real(Vcr)/2 + real(Vcs)/2];


S(ind12,:) = [                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,                   0,                   0,
0,                   0,                        0,
0,   real(Vcr)/2 + real(Vcs)/2,                        0,   real(Var)/2 +
real(Vas)/2,   real(Vbr)/2 + real(Vbs)/2];

    W(ind1,1)  = real(Iar);
    W(ind2,1)  = real(Ibr);
    W(ind3,1)  = real(Icr);
    W(ind4,1)  = real(Ias) - real(Iar);
    W(ind5,1)  = real(Ibs) - real(Ibr);
    W(ind6,1)  = real(Ics) - real(Icr);
    W(ind7,1)  = imag(Iar);
    W(ind8,1)  = imag(Ibr);
    W(ind9,1)  = imag(Icr);
    W(ind10,1) = imag(Ias) - imag(Iar);
    W(ind11,1) = imag(Ibs) - imag(Ibr);
    W(ind12,1) = imag(Ics) - imag(Icr);

    %increment counter
    k = k+1;

end
```

```
%Lowe Method-----------------------------------------------------------------
%use matrix inversion to solve for ABCD variables
beta = X\Y;

A1 = beta(1,1);
A2 = beta(2,1);
A3 = beta(3,1);
A4 = beta(4,1);
A5 = beta(5,1);
A6 = beta(6,1);
A7 = beta(7,1);
A8 = beta(8,1);
A9 = beta(9,1);

B1 = beta(10,1);
B2 = beta(11,1);
B3 = beta(12,1);
B4 = beta(13,1);
B5 = beta(14,1);
B6 = beta(15,1);
B7 = beta(16,1);
B8 = beta(17,1);
B9 = beta(18,1);

C1 = beta(19,1);
C2 = beta(20,1);
C3 = beta(21,1);
C4 = beta(22,1);
C5 = beta(23,1);
C6 = beta(24,1);
C7 = beta(25,1);
C8 = beta(26,1);
C9 = beta(27,1);

D1 = beta(28,1);
D2 = beta(29,1);
D3 = beta(30,1);
D4 = beta(31,1);
D5 = beta(32,1);
D6 = beta(33,1);
D7 = beta(34,1);
D8 = beta(35,1);
D9 = beta(36,1);


A = [A1 A2 A3;
     A4 A5 A6;
     A7 A8 A9];

B = [B1 B2 B3;
     B4 B5 B6;
```

```
     B7 B8 B9];

C = [C1 C2 C3;
     C4 C5 C6;
     C7 C8 C9];

D = [D1 D2 D3;
     D4 D5 D6;
     D7 D8 D9];

ABCD = [A B;C D];

%solve for series impedance
Zabc_Lowe = B;
Za  = B1;
Zb  = B5;
Zc  = B9;
Zab = B2;
Zac = B3;
Zbc = B6;


Yabc_Lowe = 2*(A - eye(3))*inv(Zabc_Lowe);

%Di-Shi Method--------------------------------------------------------------
%solve for the inverse series impedance matrix
%solve for the shunt admittance matrix
theta = (S'*S)\(S'*W);

Ga = theta(1,1);
Ta = theta(2,1);
Gb = theta(3,1);
Tb = theta(4,1);
Gc = theta(5,1);
Tc = theta(6,1);

Gab = theta(7,1);
Tab = theta(8,1);
Gac = theta(9,1);
Tac = theta(10,1);
Gbc = theta(11,1);
Tbc = theta(12,1);

Ba = theta(13,1);
Bb = theta(14,1);
Bc = theta(15,1);
Bab = theta(16,1);
Bac = theta(17,1);
Bbc = theta(18,1);

ya = Ga + 1i*Ta;
yb = Gb + 1i*Tb;
yc = Gc + 1i*Tc;
yab = Gab + 1i*Tab;
yac = Gac + 1i*Tac;
```

```
ybc = Gbc + 1i*Tbc;

yp = [ya yab yac;
      yab yb ybc;
      yac ybc yc];

Zabc_Shi = inv(yp);

Yabc_Shi = 1i*[Ba Bab Bac;
            Bab Bb Bbc;
            Bac Bbc Bc];

%gather reuslts from current estimation
Result_Shi_Z = [Zabc_Shi(1,1);Zabc_Shi(2,2);Zabc_Shi(3,3); ...
                Zabc_Shi(1,2);Zabc_Shi(1,3);Zabc_Shi(2,3)];
Result_Shi_Y = [Yabc_Shi(1,1);Yabc_Shi(2,2);Yabc_Shi(3,3); ...
                Yabc_Shi(1,2);Yabc_Shi(1,3);Yabc_Shi(2,3)];

Result_Lowe_Z = [Zabc_Lowe(1,1);Zabc_Lowe(2,2);Zabc_Lowe(3,3); ...
                 Zabc_Lowe(1,2);Zabc_Lowe(1,3);Zabc_Lowe(2,3)];
Result_Lowe_Y = [Yabc_Lowe(1,1);Yabc_Lowe(2,2);Yabc_Lowe(3,3); ...
                 Yabc_Lowe(1,2);Yabc_Lowe(1,3);Yabc_Lowe(2,3)];

%store parameters values from current estimation
Shi_R(e,:) = transpose(real(Result_Shi_Z));
Shi_X(e,:) = transpose(imag(Result_Shi_Z));
Shi_B(e,:) = transpose(imag(Result_Shi_Y));

Lowe_R(e,:) = transpose(real(Result_Lowe_Z));
Lowe_X(e,:) = transpose(imag(Result_Lowe_Z));
Lowe_B(e,:) = transpose(imag(Result_Lowe_Y));

deter(e,1) = det(ABCD);

end

%call function to analyze results
[meanS, meanL, stdS, stdL, pdS, pdL] = analyze(Shi_R, Shi_X, Shi_B, Lowe_R,
Lowe_X, Lowe_B);

%store results percent deviation for given PSD
Shi_PDR(m,:) = pdS(1,:);
Shi_PDX(m,:) = pdS(2,:);
Shi_PDB(m,:) = pdS(3,:);

Lowe_PDR(m,:) = pdL(1,:);
Lowe_PDX(m,:) = pdL(2,:);
Lowe_PDB(m,:) = pdL(3,:);

detmn(m,1) = mean(deter);

disp(m)
```

```
end



%plot-----------------------------------------------------------------------
if type ~= 3

%set line width
lw = 2.25;

if pick == 1
    xtitle = 'Error in V_{abc}^S (%)';
elseif pick == 2
    xtitle = 'Error in I_{abc}^S (%)';
elseif pick == 3
    xtitle = 'Error in V_{abc}^R (%)';
elseif pick == 4
    xtitle = 'Error in I_{abc}^R (%)';
elseif pick == 5
    xtitle = 'Error in All Measurements (%)';
end


%plot percent deivation for each parameter value

figure(1);
subplot(3,2,1)
hold all
grid on
xlabel(xtitle)
ylabel('Error of R (%)')
plot(PSD,Shi_PDR(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Shi_PDR(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Shi_PDR(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Shi_PDR(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Shi_PDR(:,5),'LineWidth',lw,'Color','m')
plot(PSD,Shi_PDR(:,6),'LineWidth',lw,'Color','k')
%legend('Ra','Rb','Rc','Rab','Rac','Rbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')

subplot(3,2,3)
hold all
grid on
xlabel(xtitle)
ylabel('Error of X (%)')
plot(PSD,Shi_PDX(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Shi_PDX(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Shi_PDX(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Shi_PDX(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Shi_PDX(:,5),'LineWidth',lw,'Color','m')
plot(PSD,Shi_PDX(:,6),'LineWidth',lw,'Color','k')
%legend('Xa','Xb','Xc','Xab','Xac','Xbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')
```

```
subplot(3,2,5)
hold all
grid on
xlabel(xtitle)
ylabel('Error of B (%)')
plot(PSD,Shi_PDB(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Shi_PDB(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Shi_PDB(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Shi_PDB(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Shi_PDB(:,5),'LineWidth',lw,'Color','m')
plot(PSD,Shi_PDB(:,6),'LineWidth',lw,'Color','k')
%legend('Ba','Bb','Bc','Bab','Bac','Bbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')

subplot(3,2,2)
hold all
grid on
xlabel(xtitle)
ylabel('Error of R (%)')
plot(PSD,Lowe_PDR(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Lowe_PDR(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Lowe_PDR(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Lowe_PDR(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Lowe_PDR(:,5),'LineWidth',lw,'Color','m')
plot(PSD,Lowe_PDR(:,6),'LineWidth',lw,'Color','k')
%legend('Ra','Rb','Rc','Rab','Rac','Rbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')

subplot(3,2,4)
hold all
grid on
xlabel(xtitle)
ylabel('Error of X (%)')
plot(PSD,Lowe_PDX(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Lowe_PDX(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Lowe_PDX(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Lowe_PDX(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Lowe_PDX(:,5),'LineWidth',lw,'Color','m')
plot(PSD,Lowe_PDX(:,6),'LineWidth',lw,'Color','k')
%legend('Xa','Xb','Xc','Xab','Xac','Xbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')

subplot(3,2,6)
hold all
grid on
xlabel(xtitle)
ylabel('Error of B (%)')
plot(PSD,Lowe_PDB(:,1),'LineWidth',lw,'Color','r')
plot(PSD,Lowe_PDB(:,2),'LineWidth',lw,'Color','g')
plot(PSD,Lowe_PDB(:,3),'LineWidth',lw,'Color','b')
plot(PSD,Lowe_PDB(:,4),'LineWidth',lw,'Color','c')
plot(PSD,Lowe_PDB(:,5),'LineWidth',lw,'Color','m')
```

```
plot(PSD,Lowe_PDB(:,6),'LineWidth',lw,'Color','k')
%legend('Ba','Bb','Bc','Bab','Bac','Bbc')
set(gca,'XMinorTick','on','YMinorTick','on')
%set(legend,'Location','EastOutside')

%write to excel
data = [Shi_PDR Shi_PDX Shi_PDB Lowe_PDR Lowe_PDX Lowe_PDB detmn];

if type == 1
    filename = 'bias_noise.xlsx';
elseif type == 2
    filename = 'random_noise.xlsx';
end

if write == 1
    xlswrite(filename,data,pick)
elseif write == 2
end


%--------------------------------------------------------------------------
else


end
```

Appendix A.2 – Functions

```
function [R, X, B] = TL_parameters()
% TL_parameters contains the distributed transmission line parameter
% values as calculted by ATP. The lumped parameter values are found by
% multiplying by a user selected line length.

%Parameter values for multiple lines can be stored here

%select which transmission line you would like to use
%1 = 230 kV Line    %2 = Di-Shi
line_num = 1;

%enter length of line in kilometers
length  = 150;

if line_num == 1
%230 kV Line--------------------------------------------------------------
%series resistance (ohms/kilometer)
Ra = 1.411765e-01;
Rb = 1.442426e-01;
Rc = 1.411765e-01;
Rab = 1.671007e-02;
Rac = 1.319457e-02;
Rbc = 1.671007e-02;

%series reactance (ohms/kilometer)
Xa = 4.253167e-01;
Xb = 4.215591e-01;
Xc = 4.253167e-01;
Xab = 8.591673e-02;
Xac = 4.356071e-02;
Xbc = 8.591673e-02;

%shunt admittance (mhos/kilometer)
Ba = 4.036680e-06;
Bb = 4.203311e-06;
Bc = 4.036680e-06;
Bab = -6.766561e-07;
Bac = -1.997195e-07;
Bbc = -6.766561e-07;


elseif line_num == 2
%Di Shi--------------------------------------------------------------------
%series resistance
Ra = 1.715100e+00;
Rb = 1.704296e+00;
Rc = 1.693904e+00;
Rab = 8.258347e-01;
Rac = 8.204653e-01;
Rbc = 8.153000e-01;

%series impedance
Xa = 1.220257e+01;
```

```
Xb = 1.221442e+01;
Xc = 1.222631e+01;
Xab = 5.541996e+00;
Xac = 4.791037e+00;
Xbc = 5.553867e+00;

%shunt admittance
Ba = 4.365140e-05;
Bb = 4.217730e-05;
Bc = 4.217730e-05;
Bab = -6.430200e-06;
Bac = -6.430200e-06;
Bbc = -9.697600e-06;

end


%find line total values-----------------------------------------------------
R = length*[Ra;Rb;Rc;Rab;Rac;Rbc];
X = length*[Xa;Xb;Xc;Xab;Xac;Xbc];
B = length*[Ba;Bb;Bc;Bab;Bac;Bbc];

end
```

```
function [VabcS, IabcS, VabcR, IabcR] =
create_measurements(num_meas,type,PSD,pick)
%This function creates a set of PMU measurements using the transmission
%line parameter values. It also adds noise to the measurements based on
%selections made by the user

%TL parameter entry-------------------------------------------------------
[R,X,B] = TL_parameters();

%series resistance
Ra  = R(1,1);
Rb  = R(2,1);
Rc  = R(3,1);
Rab = R(4,1);
Rac = R(5,1);
Rbc = R(6,1);

%series impedance
Xa  = X(1,1);
Xb  = X(2,1);
Xc  = X(3,1);
Xab = X(4,1);
Xac = X(5,1);
Xbc = X(6,1);

%shunt admittance
Ba  = B(1,1);
Bb  = B(2,1);
Bc  = B(3,1);
Bab = B(4,1);
Bac = B(5,1);
Bbc = B(6,1);

%construct phase impedance matrix
Za = Ra + 1i*Xa;
Zb = Rb + 1i*Xb;
Zc = Rc + 1i*Xc;
Zab = Rab + 1i*Xab;
Zac = Rac + 1i*Xac;
Zbc = Rbc + 1i*Xbc;

Zabc = [Za Zab Zac;
        Zab Zb Zbc;
        Zac Zbc Zc];

%construct shunt admittance matrix
Yabc = 1i*[Ba Bab Bac;
           Bab Bb Bbc;
           Bac Bbc Bc];


%create RECEIVING END Measurements in magnitude/angle---------------------
%create impedance & voltage measurements
%enter the bounds for impedance/voltage magnitude & angle
Zlower = 500;   %50
```

```
Zupper = 600;   %60

Vlower = 40E3;  %240E3
Vupper = 20E3;  %220E3

%seperate angle bounds into three phases to maintain positive sequence
Alower =  0;      %0
Aupper =  120;    %120
Blower = -120;    %-120
Bupper =  0;      %0
Clower =  120;    %120
Cupper =  240;    %240
Dlower =  60;     %65
Dupper =  70;     %70

%create vector of impedances and voltages
ZaMag = (Zupper-Zlower)*rand(num_meas,1) + Zlower;
ZbMag = (Zupper-Zlower)*rand(num_meas,1) + Zlower;
ZcMag = (Zupper-Zlower)*rand(num_meas,1) + Zlower;

ZaAng = (Dupper-Dlower)*rand(num_meas,1) + Dlower;
ZbAng = (Dupper-Dlower)*rand(num_meas,1) + Dlower;
ZcAng = (Dupper-Dlower)*rand(num_meas,1) + Dlower;

VarMag = (Vupper-Vlower)*rand(num_meas,1) + Vlower;
VbrMag = (Vupper-Vlower)*rand(num_meas,1) + Vlower;
VcrMag = (Vupper-Vlower)*rand(num_meas,1) + Vlower;

VarAng = (Aupper-Alower)*rand(num_meas,1) + Alower;
VbrAng = (Bupper-Blower)*rand(num_meas,1) + Blower;
VcrAng = (Cupper-Clower)*rand(num_meas,1) + Clower;

%form complex number...
Zaload = ZaMag.*(cosd(ZaAng) + 1i*sind(ZaAng));
Zbload = ZbMag.*(cosd(ZbAng) + 1i*sind(ZbAng));
Zcload = ZcMag.*(cosd(ZcAng) + 1i*sind(ZcAng));

Var = VarMag.*(cosd(VarAng) + 1i*sind(VarAng));
Vbr = VbrMag.*(cosd(VbrAng) + 1i*sind(VbrAng));
Vcr = VcrMag.*(cosd(VcrAng) + 1i*sind(VcrAng));

Zload = [Zaload Zbload Zcload];
Vload = [Var Vbr Vcr];

%compute rec. end currents
Iload = Vload./Zload;


%create SENDING END Measurements in magnitude/angle------------------------
VS = zeros(num_meas,3);
IS = zeros(num_meas,3);

for i = 1:num_meas
```

```
VabcR = transpose(Vload(i,:));
IabcR = transpose(Iload(i,:));

%ABCD equations for pi-model
VabcS = VabcR + 0.5*Zabc*Yabc*VabcR + Zabc*IabcR;
IabcS = Yabc*VabcR + 0.25*Yabc*Zabc*Yabc*VabcR + IabcR + 0.5*Yabc*Zabc*IabcR;

VS(i,:) = transpose(VabcS);
IS(i,:) = transpose(IabcS);

end

VabcR = Vload;
IabcR = Iload;
VabcS = VS;
IabcS = IS;


%add noise to measurements-------------------------------------------------

VSR = real(VabcS);
VRR = real(VabcR);
ISR = real(IabcS);
IRR = real(IabcR);

VSI = imag(VabcS);
VRI = imag(VabcR);
ISI = imag(IabcS);
IRI = imag(IabcR);


%pick decides which measurement will recieve the noise
%1 = VSR  %2 = ISR  %3 = VRR  %4 = IRR
%5 = ALL

if pick == 1
    apply = [PSD;0;0;0;PSD;0;0;0];
elseif pick == 2
    apply = [0;0;PSD;0;0;0;PSD;0];
elseif pick == 3
    apply = [0;PSD;0;0;0;PSD;0;0];
elseif pick == 4
    apply = [0;0;0;PSD;0;0;0;PSD];
elseif pick == 5
    apply = PSD*ones(8,1);
end




if type == 1
%bias noise added to measurement
NVSR = VSR.*apply(1,1) + VSR;
NVRR = VRR.*apply(2,1) + VRR;
```

```
NISR = ISR.*apply(3,1) + ISR;
NIRR = IRR.*apply(4,1) + IRR;

NVSI = VSI.*apply(5,1) + VSI;
NVRI = VRI.*apply(6,1) + VRI;
NISI = ISI.*apply(7,1) + ISI;
NIRI = IRI.*apply(8,1) + IRI;

%combine real & imag parts and form noisy measurement
VabcS = NVSR + 1i*NVSI;
IabcS = NISR + 1i*NISI;
VabcR = NVRR + 1i*NVRI;
IabcR = NIRR + 1i*NIRI;

elseif type == 2
%The standard deviation for each measurement is given by:
%standard deviation = mean * percent
sd_VSR = abs(VSR*apply(1,1));
sd_VRR = abs(VRR*apply(2,1));
sd_ISR = abs(ISR*apply(3,1));
sd_IRR = abs(IRR*apply(4,1));

sd_VSI = abs(VSI*apply(5,1));
sd_VRI = abs(VRI*apply(6,1));
sd_ISI = abs(ISI*apply(7,1));
sd_IRI = abs(IRI*apply(8,1));

%Add normally distributed noise to measurements using randn function
%noise will be added to both real/imag parts equally
J = rand(num_meas,3);
K = rand(num_meas,3);
L = rand(num_meas,3);
M = rand(num_meas,3);

NVSR = sd_VSR.*J + VSR;
NVRR = sd_VRR.*K + VRR;
NISR = sd_ISR.*L + ISR;
NIRR = sd_IRR.*M + IRR;

NVSI = sd_VSI.*J + VSI;
NVRI = sd_VRI.*K + VRI;
NISI = sd_ISI.*L + ISI;
NIRI = sd_IRI.*M + IRI;

%combine real & imag parts and form noisy measurement
VabcS = NVSR + 1i*NVSI;
IabcS = NISR + 1i*NISI;
VabcR = NVRR + 1i*NVRI;
IabcR = NIRR + 1i*NIRI;

elseif type == 3
%combination of bias and random noise
%step one - add bias noise
bupper = 0.01;
blower = -0.01;
```

```
abias = (bupper - blower)*rand(24,1) + blower;

NVSR(:,1) = VSR(:,1).*abias(1,1) + VSR(:,1);
NVRR(:,1) = VRR(:,1).*abias(2,1) + VRR(:,1);
NISR(:,1) = ISR(:,1).*abias(3,1) + ISR(:,1);
NIRR(:,1) = IRR(:,1).*abias(4,1) + IRR(:,1);

NVSI(:,1) = VSI(:,1).*abias(5,1) + VSI(:,1);
NVRI(:,1) = VRI(:,1).*abias(6,1) + VRI(:,1);
NISI(:,1) = ISI(:,1).*abias(7,1) + ISI(:,1);
NIRI(:,1) = IRI(:,1).*abias(8,1) + IRI(:,1);

NVSR(:,2) = VSR(:,2).*abias(9,1) + VSR(:,2);
NVRR(:,2) = VRR(:,2).*abias(10,1) + VRR(:,2);
NISR(:,2) = ISR(:,2).*abias(11,1) + ISR(:,2);
NIRR(:,2) = IRR(:,2).*abias(12,1) + IRR(:,2);

NVSI(:,2) = VSI(:,2).*abias(13,1) + VSI(:,2);
NVRI(:,2) = VRI(:,2).*abias(14,1) + VRI(:,2);
NISI(:,2) = ISI(:,2).*abias(15,1) + ISI(:,2);
NIRI(:,2) = IRI(:,2).*abias(16,1) + IRI(:,2);

NVSR(:,3) = VSR(:,3).*abias(17,1) + VSR(:,3);
NVRR(:,3) = VRR(:,3).*abias(18,1) + VRR(:,3);
NISR(:,3) = ISR(:,3).*abias(19,1) + ISR(:,3);
NIRR(:,3) = IRR(:,3).*abias(20,1) + IRR(:,3);

NVSI(:,3) = VSI(:,3).*abias(21,1) + VSI(:,3);
NVRI(:,3) = VRI(:,3).*abias(22,1) + VRI(:,3);
NISI(:,3) = ISI(:,3).*abias(23,1) + ISI(:,3);
NIRI(:,3) = IRI(:,3).*abias(24,1) + IRI(:,3);


%step two - add random noise
%the mean of the measurements will be the values as computed by ATP
%Therefore, the standard deviation for each measurement is given by:
%standard deviation = mean * percent
sd_VSR = abs(NVSR*apply(1,1));
sd_VRR = abs(NVRR*apply(2,1));
sd_ISR = abs(NISR*apply(3,1));
sd_IRR = abs(NIRR*apply(4,1));

sd_VSI = abs(NVSI*apply(5,1));
sd_VRI = abs(NVRI*apply(6,1));
sd_ISI = abs(NISI*apply(7,1));
sd_IRI = abs(NIRI*apply(8,1));


MVSR = sd_VSR.*rand(num_meas,3) + NVSR;
MVRR = sd_VRR.*rand(num_meas,3) + NVRR;
MISR = sd_ISR.*rand(num_meas,3) + NISR;
MIRR = sd_IRR.*rand(num_meas,3) + NIRR;
```

```
MVSI = sd_VSI.*rand(num_meas,3) + NVSI;
MVRI = sd_VRI.*rand(num_meas,3) + NVRI;
MISI = sd_ISI.*rand(num_meas,3) + NISI;
MIRI = sd_IRI.*rand(num_meas,3) + NIRI;

%combine real & imag parts and form noisy measurement
VabcS = MVSR + 1i*MVSI;
IabcS = MISR + 1i*MISI;
VabcR = MVRR + 1i*MVRI;
IabcR = MIRR + 1i*MIRI;



end
end
```

```
function [meanS, meanL, stdS, stdL, pdS, pdL] = analyze(Shi_R, Shi_X, Shi_B,
Lowe_R, Lowe_X, Lowe_B)
%This function analyzes the results of the parameter
%estimations. It finds the mean, standard deviation, and percent deviation
%for every parameter. This information is passed back to the script for
%plotting purposes

%select domain 1=phase , 2=sequence
domain = 1;

%Enter TL parameters------------------------------------------------------
[R,X,B] = TL_parameters();

%series resistance
Ra  = R(1,1);
Rb  = R(2,1);
Rc  = R(3,1);
Rab = R(4,1);
Rac = R(5,1);
Rbc = R(6,1);

%series impedance
Xa  = X(1,1);
Xb  = X(2,1);
Xc  = X(3,1);
Xab = X(4,1);
Xac = X(5,1);
Xbc = X(6,1);

%shunt admittance
Ba  = B(1,1);
Bb  = B(2,1);
Bc  = B(3,1);
Bab = B(4,1);
Bac = B(5,1);
Bbc = B(6,1);

trueR = [Ra;Rb;Rc;Rab;Rac;Rbc];
trueX = [Xa;Xb;Xc;Xab;Xac;Xbc];
trueB = [Ba;Bb;Bc;Bab;Bac;Bbc];



%transform for phase to sequence domain if selected-----------------------
if domain == 2
a = cosd(120) +1i*sind(120);
A = [1 1 1; 1 a^2 a; 1 a a^2];

%construct phase impedance matrix
Za = Ra + 1i*Xa;
Zb = Rb + 1i*Xb;
Zc = Rc + 1i*Xc;
Zab = Rab + 1i*Xab;
Zac = Rac + 1i*Xac;
Zbc = Rbc + 1i*Xbc;
```

```
Zabc = [Za Zab Zac;
        Zab Zb Zbc;
        Zac Zbc Zc];

%construct shunt admittance matrix
Yabc = 1i*[Ba Bab Bac;
           Bab Bb Bbc;
           Bac Bbc Bc];

%convert TL parameters to sequence domain
Z012 = A\Zabc*A;
Y012 = A\Yabc*A;

R0 = real(Z012(1,1));
R1 = real(Z012(2,2));
R2 = real(Z012(3,3));
R01 = real(Z012(1,2));
R02 = real(Z012(1,3));
R12 = real(Z012(2,3));

X0 = imag(Z012(1,1));
X1 = imag(Z012(2,2));
X2 = imag(Z012(3,3));
X01 = imag(Z012(1,2));
X02 = imag(Z012(1,3));
X12 = imag(Z012(2,3));

B0 = imag(Y012(1,1));
B1 = imag(Y012(2,2));
B2 = imag(Y012(3,3));
B01 = imag(Y012(1,2));
B02 = imag(Y012(1,3));
B12 = imag(Y012(2,3));


trueR = [R0;R1;R2;R01;R02;R12];
trueX = [X0;X1;X2;X01;X02;X12];
trueB = [B0;B1;B2;B01;B02;B12];


%convert data to sequence domain
num_meas = length(Shi_R);

Shi_R012 = zeros(num_meas,6);
Shi_X012 = zeros(num_meas,6);
Shi_B012 = zeros(num_meas,6);

Lowe_R012 = zeros(num_meas,6);
Lowe_X012 = zeros(num_meas,6);
Lowe_B012 = zeros(num_meas,6);

for i = 1:num_meas
Za_Shi = Shi_R(i,1) + 1i*Shi_X(i,1);
```

```
Zb_Shi = Shi_R(i,2) + 1i*Shi_X(i,2);
Zc_Shi = Shi_R(i,3) + 1i*Shi_X(i,3);
Zab_Shi = Shi_R(i,4) + 1i*Shi_X(i,4);
Zac_Shi = Shi_R(i,5) + 1i*Shi_X(i,5);
Zbc_Shi = Shi_R(i,6) + 1i*Shi_X(i,6);

Za_Lowe = Lowe_R(i,1) + 1i*Lowe_X(i,1);
Zb_Lowe = Lowe_R(i,2) + 1i*Lowe_X(i,2);
Zc_Lowe = Lowe_R(i,3) + 1i*Lowe_X(i,3);
Zab_Lowe = Lowe_R(i,4) + 1i*Lowe_X(i,4);
Zac_Lowe = Lowe_R(i,5) + 1i*Lowe_X(i,5);
Zbc_Lowe = Lowe_R(i,6) + 1i*Lowe_X(i,6);

Zabc_Shi = [Za_Shi  Zab_Shi Zac_Shi;
            Zab_Shi Zb_Shi  Zbc_Shi;
            Zac_Shi Zbc_Shi Zc_Shi];

Zabc_Lowe = [Za_Lowe  Zab_Lowe Zac_Lowe;
             Zab_Lowe Zb_Lowe  Zbc_Lowe;
             Zac_Lowe Zbc_Lowe Zc_Lowe];

Yabc_Shi = 1i*[Shi_B(i,1) Shi_B(i,4) Shi_B(i,5)
               Shi_B(i,4) Shi_B(i,2) Shi_B(i,6)
               Shi_B(i,5) Shi_B(i,6) Shi_B(i,3)];

Yabc_Lowe = 1i*[Lowe_B(i,1) Lowe_B(i,4) Lowe_B(i,5)
                Lowe_B(i,4) Lowe_B(i,2) Lowe_B(i,6)
                Lowe_B(i,5) Lowe_B(i,6) Lowe_B(i,3)];

Z012_Shi = A\Zabc_Shi*A;
Y012_Shi = A\Yabc_Shi*A;
Z012_Lowe = A\Zabc_Lowe*A;
Y012_Lowe = A\Yabc_Lowe*A;

R0S = real(Z012_Shi(1,1));
R1S = real(Z012_Shi(2,2));
R2S = real(Z012_Shi(3,3));
R01S = real(Z012_Shi(1,2));
R02S = real(Z012_Shi(1,3));
R12S = real(Z012_Shi(2,3));

X0S = imag(Z012_Shi(1,1));
X1S = imag(Z012_Shi(2,2));
X2S = imag(Z012_Shi(3,3));
X01S = imag(Z012_Shi(1,2));
X02S = imag(Z012_Shi(1,3));
X12S = imag(Z012_Shi(2,3));

B0S = imag(Y012_Shi(1,1));
B1S = imag(Y012_Shi(2,2));
B2S = imag(Y012_Shi(3,3));
B01S = imag(Y012_Shi(1,2));
B02S = imag(Y012_Shi(1,3));
B12S = imag(Y012_Shi(2,3));
```

```
R0L = real(Z012_Lowe(1,1));
R1L = real(Z012_Lowe(2,2));
R2L = real(Z012_Lowe(3,3));
R01L = real(Z012_Lowe(1,2));
R02L = real(Z012_Lowe(1,3));
R12L = real(Z012_Lowe(2,3));

X0L = imag(Z012_Lowe(1,1));
X1L = imag(Z012_Lowe(2,2));
X2L = imag(Z012_Lowe(3,3));
X01L = imag(Z012_Lowe(1,2));
X02L = imag(Z012_Lowe(1,3));
X12L = imag(Z012_Lowe(2,3));

B0L = imag(Y012_Lowe(1,1));
B1L = imag(Y012_Lowe(2,2));
B2L = imag(Y012_Lowe(3,3));
B01L = imag(Y012_Lowe(1,2));
B02L = imag(Y012_Lowe(1,3));
B12L = imag(Y012_Lowe(2,3));


Shi_R012(i,:) = [R0S;R1S;R2S;R01S;R02S;R12S]';
Shi_X012(i,:) = [X0S;X1S;X2S;X01S;X02S;X12S]';
Shi_B012(i,:) = [B0S;B1S;B2S;B01S;B02S;B12S]';
Lowe_R012(i,:) = [R0L;R1L;R2L;R01L;R02L;R12L]';
Lowe_X012(i,:) = [X0L;X1L;X2L;X01L;X02L;X12L]';
Lowe_B012(i,:) = [B0L;B1L;B2L;B01L;B02L;B12L]';

end

Shi_R = Shi_R012;
Shi_X = Shi_X012;
Shi_B = Shi_B012;

Lowe_R = Lowe_R012;
Lowe_X = Lowe_X012;
Lowe_B = Lowe_B012;


end
%--------------------------------------------------------------------------




%find the average of each parameter value---------------------------------
MSR = [mean(Shi_R(:,1)); mean(Shi_R(:,2)); mean(Shi_R(:,3)); ...
       mean(Shi_R(:,4)); mean(Shi_R(:,5)); mean(Shi_R(:,6))];

MSX = [mean(Shi_X(:,1)); mean(Shi_X(:,2)); mean(Shi_X(:,3)); ...
       mean(Shi_X(:,4)); mean(Shi_X(:,5)); mean(Shi_X(:,6))];

MSB = [mean(Shi_B(:,1)); mean(Shi_B(:,2)); mean(Shi_B(:,3)); ...
```

```
        mean(Shi_B(:,4)); mean(Shi_B(:,5)); mean(Shi_B(:,6))];

MLR = [mean(Lowe_R(:,1)); mean(Lowe_R(:,2)); mean(Lowe_R(:,3)); ...
       mean(Lowe_R(:,4)); mean(Lowe_R(:,5)); mean(Lowe_R(:,6))];

MLX = [mean(Lowe_X(:,1)); mean(Lowe_X(:,2)); mean(Lowe_X(:,3)); ...
       mean(Lowe_X(:,4)); mean(Lowe_X(:,5)); mean(Lowe_X(:,6))];

MLB = [mean(Lowe_B(:,1)); mean(Lowe_B(:,2)); mean(Lowe_B(:,3)); ...
       mean(Lowe_B(:,4)); mean(Lowe_B(:,5)); mean(Lowe_B(:,6))];

meanS = [MSR';MSX';MSB'];
meanL = [MLR';MLX';MLB'];


%find the standard deviation of each parameter value-----------------------
SSR = [std(Shi_R(:,1)); std(Shi_R(:,2)); std(Shi_R(:,3)); ...
       std(Shi_R(:,4)); std(Shi_R(:,5)); std(Shi_R(:,6))];

SSX = [std(Shi_X(:,1)); std(Shi_X(:,2)); std(Shi_X(:,3)); ...
       std(Shi_X(:,4)); std(Shi_X(:,5)); std(Shi_X(:,6))];

SSB = [std(Shi_B(:,1)); std(Shi_B(:,2)); std(Shi_B(:,3)); ...
       std(Shi_B(:,4)); std(Shi_B(:,5)); std(Shi_B(:,6))];

SLR = [std(Lowe_R(:,1)); std(Lowe_R(:,2)); std(Lowe_R(:,3)); ...
       std(Lowe_R(:,4)); std(Lowe_R(:,5)); std(Lowe_R(:,6))];

SLX = [std(Lowe_X(:,1)); std(Lowe_X(:,2)); std(Lowe_X(:,3)); ...
       std(Lowe_X(:,4)); std(Lowe_X(:,5)); std(Lowe_X(:,6))];

SLB = [std(Lowe_B(:,1)); std(Lowe_B(:,2)); std(Lowe_B(:,3)); ...
       std(Lowe_B(:,4)); std(Lowe_B(:,5)); std(Lowe_B(:,6))];

stdS = [SSR';SSX';SSB'];
stdL = [SLR';SLX';SLB'];


%find the percent deviation between the mean and true values---------------
PDSR = ((trueR - MSR)./trueR)*100;
PDSX = ((trueX - MSX)./trueX)*100;
PDSB = ((trueB - MSB)./trueB)*100;

PDLR = ((trueR - MLR)./trueR)*100;
PDLX = ((trueX - MLX)./trueX)*100;
PDLB = ((trueB - MLB)./trueB)*100;

pdS = [PDSR';PDSX';PDSB'];
pdL = [PDLR';PDLX';PDLB'];

end
```