

Distributed Gauss-Newton Method for AC State Estimation Using Belief Propagation

Mirsad Cosovic, *Student Member, IEEE*, Dejan Vukobratovic, *Member, IEEE*

Abstract—We present a detailed study on application of factor graphs and the belief propagation (BP) algorithm to the power system state estimation (SE) problem. We start from the BP solution for the linear DC model, for which we provide a detailed convergence analysis. Using insights from the DC model, we use two different approaches to derive the BP algorithm for the non-linear AC model. The first method directly applies BP methodology, however, providing only approximate BP solution for the AC model. In the second approach, we make a key further step by providing the solution in which the BP is applied sequentially over the AC model, akin to what is done by the Gauss-Newton method. The resulting BP-based Gauss-Newton algorithm has the interpretation of a fully distributed Gauss-Newton method with the same accuracy as the centralized SE, preserving a number of advantages of the BP framework. The paper provides extensive numerical study of the proposed algorithms, provides details on their convergence properties, and gives a number of useful insights for their implementation.

Index Terms—State Estimation, Electric Power System, Factor Graphs, Belief Propagation, Distributed Gauss-Newton Method

I. INTRODUCTION

Motivation: Electric power systems consist of generation, transmission and consumption spread over wide geographical areas and operated from the control centers by the system operators. Maintaining normal operation conditions is of the central importance for the power system operators [1, Ch. 1]. Control centers are traditionally operated in centralized and independent fashion. However, increase in the system size and complexity, as well as external socio-economic factors, lead to deregulation of power systems, resulting in decentralized structure with distributed control centers. Cooperation in control and monitoring across distributed control centers is critical for efficient system operation. Consequently, existing centralized algorithms have to be redefined based on a new requirements for distributed operation, scalability and computational efficiency [2].

The system monitoring is an essential part of the control centers, providing control and optimization functionality

whose efficiency relies on accurate state estimation (SE). The core of the SE is the SE algorithm that provides an estimate of the system state (i.e., the set of all complex bus voltages) based on the network topology and available measurements. Besides the SE algorithm, the SE includes several additional routines such as network topology processors, observability analysis and bad data analysis.

The centralized SE assumes that the measurements collected across the system are available at the control center, where the centralized SE algorithm provides the system state estimate. Precisely, the centralized SE algorithm typically uses the Gauss-Newton method to solve the non-linear weighted least-squares (WLS) problem [3], [4]. In contrast, decentralized SE distributes communication and computational effort across multiple control centers to provide the system state estimate. There are two main approaches to distributed SE: i) algorithms which require a global control center to exchange data with local control centers, and ii) algorithms with local control centers only [5]. Distributed SE algorithms target the same state estimate accuracy as achievable using the centralized SE algorithms.

Literature review: The mainstream distributed SE algorithms exploit matrix decomposition techniques applied over the Gauss-Newton method. These algorithms usually achieve the same accuracy as the centralized SE algorithm and work with either global control center [6]–[8] or without it [9]–[12]. Recently, the SE algorithms based on distributed optimization [13], and in particular, the alternating direction method of multipliers became very popular [14]. Examples include algorithms based on convex relaxation [15], [16], and without convex relaxation [17]. In [18], the gossip-based Gauss-Newton algorithm is proposed which provides flexible communication model, but suffers from slight performance degradation compared to the centralized SE. The work in [19] presents a fully distributed SE algorithm for wide-area monitoring which provably converges to the centralized SE. We refer the reader to [20] for a detailed survey of the distributed multi-area SE.

Belief-Propagation Approach: In this paper, we solve the SE problem using probabilistic graphical models [21], a powerful tool for modeling the dependencies among the systems of random variables. We represent the SE problem using a popular class of probabilistic graphical models called factor graphs and solve it using the belief propagation (BP) algorithm. Applying the BP algorithm on probabilistic graphical models without loops, one obtains exact marginal distributions or a mode of the joint distribution of the system of random variables [21], [22]. The BP algorithm can be

This paper was presented in part at the IEEE International Conference on Smart Grid Communications (SmartGridComm), Sydney, Australia, November 2016; in part at the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Edinburgh, United Kingdom, August 2016.

M. Cosovic is with Schneider Electric DMS NS, Novi Sad, Serbia (e-mail: mirsad.cosovic@schneider-electric-dms.com). D. Vukobratovic is with Department of Power, Electronic and Communications Engineering, University of Novi Sad, Novi Sad, Serbia (e-mail: dejanv@uns.ac.rs). Demo source code available online at <https://github.com/mcosovic>.

This paper has received funding from the EU 7th Framework Programme for research, technological development and demonstration under grant agreement no. 607774.

also applied to graphical models with loops (loopy BP) [23], although in that case, the solution may not converge to the correct marginals/modes of the joint distribution. BP is a fully distributed algorithm suitable for accommodation of distributed power sources and time-varying loads. Moreover, placing the SE into the probabilistic graphical modelling framework enables not only efficient inference, but also, a rich collection of tools for learning parameters or structure of the graphical model from observed data [24].

The work in [25] provides the first demonstration of BP applied to the SE problem. Although this work is elaborate in terms of using, e.g., environmental correlation via historical data, it applies BP to a simple linearized DC model. The AC model is recently addressed in [26], where tree-reweighted BP is applied using preprocessed weights obtained by randomly sampling the space of spanning trees. The work in [27] investigates Gaussian BP convergence performance for the DC model. Although the above results provide initial insights on using BP for distributed SE, it is fair to say that a systematic analysis of applying the BP algorithm on the SE problem, and in particular for the AC model, is still missing. This paper intends to fill this gap.

Contributions: In this paper, we present a novel distributed BP-based Gauss-Newton algorithm. In the process of deriving the proposed algorithm, we provide a step-by-step guide for application of BP algorithm to the SE problem, giving this paper strong tutorial flavor. Our methodology is to start with the simplest linear DC model and use insights obtained therein to derive the native BP solution for the non-linear AC model. Unfortunately, as closed-form expressions for certain classes of BP messages cannot be obtained, we present suitable approximations that lead us to propose the AC-BP algorithm as an (approximate) BP solution for the AC model. Then, as a main contribution, we make a key further step where we change the perspective of our BP approach and, instead of applying the BP directly onto the non-linear AC model, we present the solution where the BP is applied sequentially over the AC model, akin to what is done by the Gauss-Newton method. The resulting Gauss-Newton BP (GN-BP) represents a BP counterpart of the Gauss-Newton method achieving the same accuracy, however, preserving a number of advantages brought in by the BP framework. For example, the proposed GN-BP algorithm is of considerably lower computational complexity, can be easily designed to provide asynchronous operation, could be integrated as part of real-time systems [28], and is flexible and easy to distribute and parallelize. Finally, this paper provides a novel and detailed convergence analysis of the DC-BP algorithm and points to extension of this analysis for the proposed GN-BP algorithm. We close this paper by extensive numerical study where we compare the proposed BP-based algorithms with the centralized SE, and provide a number of useful insights for their implementation.

Paper Organization: In Section II, we present basic concepts of factor graphs and BP. Section III reviews the centralized DC and AC SE. Section IV presents closed form BP expressions and detailed convergence analysis for the DC model. In Section V, the AC model is solved via BP-based

Gauss-Newton method. Section VI presents extended numerical convergence and performance results. Concluding remarks are given in Section VII.

II. FACTOR GRAPHS AND BP ALGORITHM

Factor graphs and BP algorithm are widely used tools for probabilistic inference [21], [22]. In the standard setup, the goal of the BP algorithm is to efficiently evaluate the marginals of a system of random variables $\mathbf{x} = (x_1, \dots, x_n)$ described via the joint probability density function $g(\mathbf{x})$. Assuming that the function $g(\mathbf{x})$ can be factorized proportionally (\propto) to a product of local functions:

$$g(\mathbf{x}) \propto \prod_{i=1}^k \psi_i(\mathcal{X}_i), \quad (1)$$

where $\mathcal{X}_i \subseteq \{x_1, \dots, x_n\}$, the marginalization problem can be efficiently solved using BP algorithm.

The first step is forming a factor graph, which is a bipartite graph that describes the structure of the factorization (1). The factor graph structure comprises the set of factor nodes $\mathcal{F} = \{f_1, \dots, f_k\}$, where each factor node f_i represents local function $\psi_i(\mathcal{X}_i)$, and the set of variable nodes $\mathcal{X} = \{x_1, \dots, x_n\}$. The factor node f_i connects to the variable node x_s if and only if $x_s \in \mathcal{X}_i$ [29].

The BP algorithm on factor graphs proceeds by passing two types of messages along the edges of the factor graph: i) a variable node to a factor node, and ii) a factor node to a variable node messages. Both variable and factor nodes in a factor graph process the incoming messages and calculate outgoing messages, where an output message on any edge depends on incoming messages from all other edges. BP messages represent "beliefs" about variable nodes, thus a message that arrives or departs a certain variable node is a function (distribution) of the random variable corresponding to the variable node.

Message from a variable node to a factor node: Consider a part of a factor graph shown in Fig. 1 with a group of factor nodes $\mathcal{F}_s = \{f_i, f_w, \dots, f_W\} \subseteq \mathcal{F}$ that are neighbours of the variable node $x_s \in \mathcal{X}$. The message $\mu_{x_s \rightarrow f_i}(x_s)$ from the

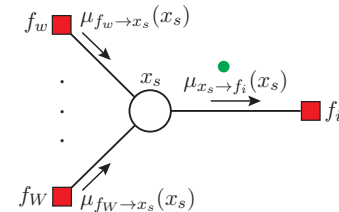


Fig. 1. Message $\mu_{x_s \rightarrow f_i}(x_s)$ from variable node x_s to factor node f_i

variable node x_s to the factor node f_i is equal to the product of all incoming factor node to variable node messages arriving at all the other incident edges:

$$\mu_{x_s \rightarrow f_i}(x_s) = \prod_{f_a \in \mathcal{F}_s \setminus f_i} \mu_{f_a \rightarrow x_s}(x_s), \quad (2)$$

where $\mathcal{F}_s \setminus f_i$ represents the set of factor nodes incident to the variable node x_s , excluding the factor node f_i . Note that each message is a function of the variable x_s .

Message from a factor node to a variable node: Consider a part of a factor graph shown in Fig. 2 that consists of a group of variable nodes $\mathcal{X}_i = \{x_s, x_l, \dots, x_L\} \subseteq \mathcal{X}$ that are neighbours of the factor node $f_i \in \mathcal{F}$. The

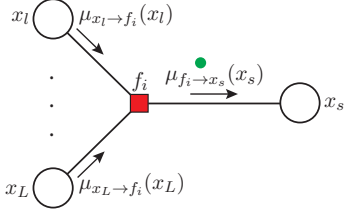


Fig. 2. Message $\mu_{f_i \rightarrow x_s}(x_s)$ from factor node f_i to variable node x_s

message $\mu_{f_i \rightarrow x_s}(x_s)$ from the factor node f_i to the variable node x_s is defined as a product of all incoming variable node to factor node messages arriving at other incident edges, multiplied by the function $\psi_i(\mathcal{X}_i)$ associated to the factor node f_i , and marginalized over all of the variables associated with the incoming messages:

$$\mu_{f_i \rightarrow x_s}(x_s) = \int \cdots \int \psi_i(\mathcal{X}_i) \prod_{x_b \in \mathcal{X}_i \setminus x_s} [\mu_{x_b \rightarrow f_i}(x_b) \cdot dx_b], \quad (3)$$

where $\mathcal{X}_i \setminus x_s$ is the set of variable nodes incident to the factor node f_i , excluding the variable node x_s .

Marginal inference: The marginal of the variable node x_s , illustrated in Fig. 3, is obtained as the product of all incoming messages into the variable node x_s :

$$p(x_s) = \prod_{f_c \in \mathcal{F}_s} \mu_{f_c \rightarrow x_s}(x_s), \quad (4)$$

where \mathcal{F}_s is the set of factor nodes incident to the variable node x_s .

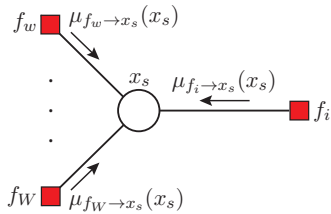


Fig. 3. Marginal inference of the variable node x_s

In the following, we focus on the Gaussian BP (GBP) over the set of real-valued random variables where all the local functions $\psi_i(\mathcal{X}_i)$ and all the inputs to the BP algorithm represent Gaussian distributions. Due to the fact that variable node and factor node processing preserves ‘‘Gaussianity’’ of the messages, the resulting GBP algorithm operates by exchanging the messages that represent Gaussian functions. Therefore, each message exchanged in GBP is completely represented using only two values: the mean and the variance [30].

III. SE IN ELECTRIC POWER SYSTEMS

The SE algorithm estimates the values of the state variables based on the knowledge of network topology and parameters, and measured values obtained from measurement devices spread across the power system. The knowledge of the network topology and parameters is provided by the network topology processor in the form of the bus/branch model, where branches of the grid are usually described using the two-port π -model [1, Ch. 1,2]. The bus/branch model can be represented using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V} = \{1, \dots, N\}$ represents the set of buses, while the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of branches of the power network.

As an input, the SE requires a set of measurements \mathcal{M} of different electrical quantities spread across the power network. Using the bus/branch model and available measurements, the observability analysis defines the measurement model [1, Ch. 4] which can be described as the system of equations [4]:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{u}, \quad (5)$$

where $\mathbf{x} = (x_1, \dots, x_n)$ is the vector of the state variables, $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_k(\mathbf{x}))$ is the vector of measurement functions, $\mathbf{z} = (z_1, \dots, z_k)$ is the vector of measurement values, and $\mathbf{u} = (u_1, \dots, u_k)$ is the vector of uncorrelated measurement errors. The SE problem is commonly an overdetermined system of equations ($k > n$) [31, Sec. 2.1].

Each measurement $M_i \in \mathcal{M}$ is associated with measured value z_i , measurement error u_i , and measurement function $h_i(\mathbf{x})$. Under the assumption that measurement errors u_i follow a zero-mean Gaussian distribution, the probability density function associated with the i -th measurement equals:

$$\mathcal{N}(z_i | \mathbf{x}, v_i) = \frac{1}{\sqrt{2\pi v_i}} \exp \left\{ -\frac{[z_i - h_i(\mathbf{x})]^2}{2v_i} \right\}, \quad (6)$$

where v_i is the measurement variance defined by the measurement error u_i , and the measurement function $h_i(\mathbf{x})$ connects the vector of state variables \mathbf{x} to the value of the i -th measurement.

The SE in electric power systems deals with the problem of determining state variables \mathbf{x} according to the noisy observed data \mathbf{z} and a prior knowledge:

$$p(\mathbf{x} | \mathbf{z}) = \frac{p(\mathbf{z} | \mathbf{x}) p(\mathbf{x})}{p(\mathbf{z})}. \quad (7)$$

Assuming that the prior probability distribution $p(\mathbf{x})$ is uniform, and given that $p(\mathbf{z})$ does not depend on \mathbf{x} , the maximum a posteriori (MAP) solution of (7) reduces to the maximum likelihood solution, as given below [32]:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} p(\mathbf{x} | \mathbf{z}) = \arg \max_{\mathbf{x}} p(\mathbf{z} | \mathbf{x}) = \arg \max_{\mathbf{x}} \mathcal{L}(\mathbf{z} | \mathbf{x}). \quad (8)$$

One can find the solution (8) via maximization of the likelihood function $\mathcal{L}(\mathbf{z} | \mathbf{x})$, which is defined via likelihoods of k independent measurements:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \mathcal{L}(\mathbf{z} | \mathbf{x}) = \arg \max_{\mathbf{x}} \prod_{i=1}^k \mathcal{N}(z_i | \mathbf{x}, v_i). \quad (9)$$

It can be shown that the solution of the MAP problem can be obtained by solving the following optimization problem, known as the WLS problem [33, Sec. 9.3]:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \sum_{i=1}^k \frac{[z_i - h_i(\mathbf{x})]^2}{v_i}. \quad (10)$$

The state estimate $\hat{\mathbf{x}}$ representing the solution of the optimization problem (10) is known as the WLS estimator, and is equivalent to the MAP estimator (9).

A. AC State Estimation

The AC SE model is defined using the measurement functions $\mathbf{h}(\mathbf{x})$ that precisely follow the physical laws that connect the measured variables and the state variables. As a result, the system (5) in general represents the system of non-linear equations.

In a usual scenario, the AC SE model takes bus voltage magnitudes and bus voltage angles, transformer magnitudes of turns ratio and transformer angles of turns ratio as state variables \mathbf{x} . Without loss of generality, in the rest of the paper, for the AC SE we observe bus voltage angles $\boldsymbol{\theta} = (\theta_1, \dots, \theta_N)$ and bus voltage magnitudes $\mathbf{V} = (V_1, \dots, V_N)$ as state variables $\mathbf{x} \equiv (\boldsymbol{\theta}, \mathbf{V})$. Consequently, the number of state variables is $n = 2N$.

The typical set of measurements \mathcal{M} in electric power systems includes: active and reactive power flow and current magnitude, $\{M_{P_{ij}}, M_{Q_{ij}}, M_{I_{ij}}\}$, $(i, j) \in \mathcal{E}$, respectively; active and reactive power injection, $\{M_{P_i}, M_{Q_i}\}$, $i \in \mathcal{V}$; and bus voltage angle and magnitude, $\{M_{\theta_i}, M_{V_i}\}$, $i \in \mathcal{V}$. For completeness, we provide the set of non-linear measurement functions $\mathbf{h}(\mathbf{x})$ of the AC SE model given in Appendix A, equations (62) - (65).

Based on the available set of measurements, the WLS estimator $\hat{\mathbf{x}} \equiv (\hat{\boldsymbol{\theta}}, \hat{\mathbf{V}})$, i.e., the solution of the WLS problem (10), can be found using the Gauss-Newton method:

$$\left[\mathbf{J}(\mathbf{x}^{(\nu)})^T \mathbf{W} \mathbf{J}(\mathbf{x}^{(\nu)}) \right] \Delta \mathbf{x}^{(\nu)} = \mathbf{J}(\mathbf{x}^{(\nu)})^T \mathbf{W} \mathbf{r}(\mathbf{x}^{(\nu)}) \quad (11a)$$

$$\mathbf{x}^{(\nu+1)} = \mathbf{x}^{(\nu)} + \Delta \mathbf{x}^{(\nu)}, \quad (11b)$$

where $\nu = \{0, 1, 2, \dots\}$ is the iteration index, $\Delta \mathbf{x}^{(\nu)} \in \mathbb{R}^n$ is the vector of increments of the state variables, $\mathbf{J}(\mathbf{x}^{(\nu)}) \in \mathbb{R}^{k \times n}$ is the Jacobian matrix of measurement functions $\mathbf{h}(\mathbf{x}^{(\nu)})$ at $\mathbf{x} = \mathbf{x}^{(\nu)}$ (see Appendix A, for details), $\mathbf{W} \in \mathbb{R}^{k \times k}$ is a diagonal matrix containing inverses of measurement variances, and $\mathbf{r}(\mathbf{x}^{(\nu)}) = \mathbf{z} - \mathbf{h}(\mathbf{x}^{(\nu)})$ is the vector of residuals [3, Ch. 10].

B. DC State Estimation

The DC model is obtained by linearisation of the AC model. In the typical operating conditions, the difference of bus voltage angles between adjacent buses $(i, j) \in \mathcal{E}$ is very small $\theta_i - \theta_j \approx 0$, which implies $\cos \theta_{ij} \approx 1$ and $\sin \theta_{ij} \approx \theta_{ij}$. Further, all bus voltage magnitudes are $V_i \approx 1$, $i \in \mathcal{V}$, and all shunt elements and branch resistances can be neglected. This implies that the DC model ignores the reactive powers and transmission losses and takes into account only the active powers. Therefore, the DC SE takes

only bus voltage angles $\mathbf{x} \equiv \boldsymbol{\theta}$ as state variables. Consequently, the number of state variables is $n = N$.

The set of DC model measurements \mathcal{M} involves only active power flow $\{M_{P_{ij}}\}$, $(i, j) \in \mathcal{E}$, active power injection $\{M_{P_i}\}$, $i \in \mathcal{V}$, and bus voltage angle $\{M_{\theta_i}\}$, $i \in \mathcal{V}$, measurements and the model is dealing with linear measurement functions $\mathbf{h}(\mathbf{x})$ given Appendix A.

The DC state estimate $\hat{\mathbf{x}} \equiv \hat{\boldsymbol{\theta}}$, which is a solution to the WLS problem (10), is obtained through the non-iterative procedure by solving the system of linear equations:

$$(\mathbf{H}^T \mathbf{W} \mathbf{H}) \hat{\mathbf{x}} = \mathbf{H}^T \mathbf{W} \mathbf{z}, \quad (12)$$

where $\mathbf{H} \in \mathbb{R}^{k \times N}$ is the Jacobian matrix of measurement functions given in Appendix A, equations (66) - (68).

IV. BP-BASED ALGORITHM FOR THE DC SE

In the SE problem defined above, each measurement function $h_i(\mathbf{x})$ depends on a limited (typically small) subset of state variables \mathbf{x} . Hence, the likelihood function $\mathcal{L}(\mathbf{z}|\mathbf{x})$ can be factorized into factors (9) affecting small subsets of state variables. This motivates solving the SE problem scalably and efficiently using probabilistic graphical models. The solution involves defining the factor graph corresponding to (9), and deriving expressions for BP messages exchanged over the factor graph, as detailed next.

For completeness of exposition, we first present the solution of the DC SE problem using the BP algorithm; we refer to the corresponding method as the DC-BP. Furthermore, and as a novel contribution, we provide an in-depth convergence analysis of the DC-BP algorithm, and present methods to improve its convergence. The material in this section sets the stage for the main contribution of this paper - the BP-based Gauss-Newton method for AC model presented in the following section.

A. The Factor Graph Construction

For the DC model, the set of variable nodes is defined by the state variables $\boldsymbol{\theta}$, i.e., $\mathcal{X} = \{\theta_1, \dots, \theta_N\}$. The set of factor nodes $\mathcal{F} = \{f_1, \dots, f_k\}$ is defined by the set of measurements \mathcal{M} , with measurement functions given in Appendix A, equations (66) - (68). Measurements define likelihood functions $\mathcal{N}(z_i|\mathbf{x}, v_i)$ that are in turn equal to local functions $\psi_i(\mathcal{X}_i)$ associated to factor nodes. More precisely, a factor node f_i connects to a variable node $x_s \in \mathcal{X}$ if and only if the state variable x_s is an argument of the corresponding measurement function $h_i(\mathbf{x})$.

Example 1 (Constructing factor graph). *In this toy example, using a simple 3-bus model presented in Fig. 4a, we demonstrate the conversion from a bus/branch model with a given measurement configuration into the corresponding factor graph for the DC model. The variable nodes represent state variables, i.e., $\mathcal{X} = \{\theta_1, \theta_2, \theta_3\}$. Factor nodes are defined by corresponding measurements, where in our example, measurements $M_{P_{12}}$ and M_{P_3} are mapped into factor nodes $\mathcal{F} = \{f_{P_{12}}, f_{P_3}\}$. \triangle*

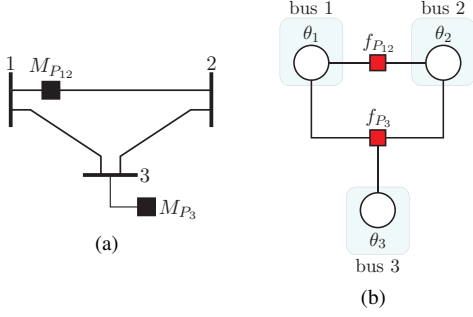


Fig. 4. Transformation of the bus/branch model and measurement configuration (subfigure a) into the corresponding factor graph for the DC model (subfigure b).

B. Derivation of BP Messages

Message from a variable node to a factor node: Let us assume that the incoming messages $\mu_{f_w \rightarrow x_s}(x_s)$, \dots , $\mu_{f_W \rightarrow x_s}(x_s)$ into the variable node x_s are Gaussian and represented by their mean-variance pairs $(z_{f_w \rightarrow x_s}, v_{f_w \rightarrow x_s})$, \dots , $(z_{f_W \rightarrow x_s}, v_{f_W \rightarrow x_s})$. According to (2), it can be shown that the message $\mu_{x_s \rightarrow f_i}(x_s)$ from the variable node x_s to the factor node f_i is proportional (\propto) to:

$$\mu_{x_s \rightarrow f_i}(x_s) \propto \mathcal{N}(z_{x_s \rightarrow f_i} | x_s, v_{x_s \rightarrow f_i}), \quad (13)$$

with mean $z_{x_s \rightarrow f_i}$ and variance $v_{x_s \rightarrow f_i}$ obtained as:

$$z_{x_s \rightarrow f_i} = \left(\sum_{f_a \in \mathcal{F}_s \setminus f_i} \frac{z_{f_a \rightarrow x_s}}{v_{f_a \rightarrow x_s}} \right) v_{x_s \rightarrow f_i} \quad (14a)$$

$$\frac{1}{v_{x_s \rightarrow f_i}} = \sum_{f_a \in \mathcal{F}_s \setminus f_i} \frac{1}{v_{f_a \rightarrow x_s}}. \quad (14b)$$

After the variable node x_s receives the messages from all of the neighbouring factor nodes from the set $\mathcal{F}_s \setminus f_i$, it evaluates the message $\mu_{x_s \rightarrow f_i}(x_s)$ according to (14) and sends it to the factor node f_i .

Message from a factor node to a variable node: Due to linearity of measurement functions $h_i(\cdot)$, closed form expressions for these messages is easy to obtain and follow a Gaussian form:

$$\mu_{f_i \rightarrow x_s}(x_s) \propto \mathcal{N}(z_{f_i \rightarrow x_s} | x_s, v_{f_i \rightarrow x_s}). \quad (15)$$

Let us assume that the messages into factor nodes are Gaussian, denoted by:

$$\begin{aligned} \mu_{x_l \rightarrow f_i}(x_l) &\propto \mathcal{N}(z_{x_l \rightarrow f_i} | x_l, v_{x_l \rightarrow f_i}) \\ &\vdots \\ \mu_{x_L \rightarrow f_i}(x_L) &\propto \mathcal{N}(z_{x_L \rightarrow f_i} | x_L, v_{x_L \rightarrow f_i}). \end{aligned} \quad (16)$$

The Gaussian function associated with the factor node f_i is given by (Section III, equation (6)):

$$\begin{aligned} \mathcal{N}(z_i | x_s, x_l, \dots, x_L, v_i) \\ \propto \exp \left\{ -\frac{[z_i - h_i(x_s, x_l, \dots, x_L)]^2}{2v_i} \right\}. \end{aligned} \quad (17)$$

The DC model contains only linear measurement functions which we represent in a general form as:

$$h_i(x_s, x_l, \dots, x_L) = C_{x_s} x_s + \sum_{x_b \in \mathcal{X}_i \setminus x_s} C_{x_b} x_b, \quad (18)$$

where $\mathcal{X}_i \setminus x_s$ is the set of variable nodes incident to the factor node f_i , excluding the variable node x_s . From the expression (3), and using (16)-(18), it can be shown that the message $\mu_{f_i \rightarrow x_s}(x_s)$ from the factor node f_i to the variable node x_s is represented by the Gaussian function (15), with mean $z_{f_i \rightarrow x_s}$ and variance $v_{f_i \rightarrow x_s}$ obtained as:

$$z_{f_i \rightarrow x_s} = \frac{1}{C_{x_s}} \left(z_i - \sum_{x_b \in \mathcal{X}_i \setminus x_s} C_{x_b} z_{x_b \rightarrow f_i} \right) \quad (19a)$$

$$v_{f_i \rightarrow x_s} = \frac{1}{C_{x_s}^2} \left(v_i + \sum_{x_b \in \mathcal{X}_i \setminus x_s} C_{x_b}^2 v_{x_b \rightarrow f_i} \right). \quad (19b)$$

To summarize, after the factor node f_i receives the messages from all of the neighbouring variable nodes from the set $\mathcal{X}_i \setminus x_s$, it evaluates the message $\mu_{f_i \rightarrow x_s}(x_s)$ according to (19a) and (19b), and sends it to the variable node x_s .

Marginal inference: According to (4), it can be shown that the marginal of the state variable x_s is represented by:

$$p(x_s) \propto \mathcal{N}(\hat{x}_s | x_s, v_{x_s}), \quad (20)$$

with the mean value \hat{x}_s and variance v_{x_s} :

$$\hat{x}_s = \left(\sum_{f_c \in \mathcal{F}_s} \frac{z_{f_c \rightarrow x_s}}{v_{f_c \rightarrow x_s}} \right) v_{x_s} \quad (21a)$$

$$\frac{1}{v_{x_s}} = \sum_{f_c \in \mathcal{F}_s} \frac{1}{v_{f_c \rightarrow x_s}}. \quad (21b)$$

Finally, the mean-value \hat{x}_s is adopted as the estimated value of the state variable x_s .

C. Iterative DC-BP Algorithm

The SE scenario is in general an instance of loopy BP since the corresponding factor graph usually contains cycles. Loopy BP is an iterative algorithm, with an iteration index $\tau = \{0, 1, 2, \dots\}$, and requires a message-passing schedule. The scheduling where messages from variable to factor nodes, and messages from factor nodes to variable nodes, are updated in parallel in respective half-iterations, is known as synchronous scheduling. Synchronous scheduling updates all messages in a given iteration using the output of the previous iteration as an input [34].

To present the algorithm precisely, we need to introduce different types of factor nodes. The *indirect factor nodes* $\mathcal{F}_{\text{ind}} \subset \mathcal{F}$ correspond to measurements that measure state variables indirectly. In the DC scenario, this includes active power flow and power injection measurements. The *direct factor nodes* $\mathcal{F}_{\text{dir}} \subset \mathcal{F}$ correspond to the measurements that measure state variables directly. For our choice of state variables for the DC scenario, an example includes measurements of angles.

Besides direct and indirect factor nodes, we define three additional types of singly-connected factor nodes. The *slack factor node* corresponds to the slack or reference bus where the voltage angle has a given value. The *initialization factor node* is required to initialize the algorithm at certain variable nodes. Finally, the *virtual factor node* is a singly-connected factor node used if the variable node is not directly measured. Both initialization and virtual factor nodes take the value of "flat start" with variance $\sigma_{x_i}^2 \rightarrow \infty$ or a priori given mean value and variance of state variables. We refer to direct factor nodes and three additional types of singly-connected factor nodes¹ as local factor nodes $\mathcal{F}_{\text{loc}} \subset \mathcal{F}$.

Algorithm 1 The DC-BP

```

1: procedure INITIALIZATION  $\tau = 0$ 
2:   for each  $f_s \in \mathcal{F}_{\text{loc}}$  do
3:     send  $\mu_{f_s \rightarrow x_s}^{(0)}$  to incident  $x_s \in \mathcal{X}$ 
4:   end for
5:   for each  $x_s \in \mathcal{X}$  do
6:     send  $\mu_{x_s \rightarrow f_i}^{(0)} = \mu_{f_s \rightarrow x_s}^{(0)}$  to incident  $f_i \in \mathcal{F}_{\text{ind}}$ 
7:   end for
8: end procedure
9: procedure ITERATION LOOP  $\tau = 1, 2, \dots$ 
10:  while stopping criterion is not met do
11:    for each  $f_i \in \mathcal{F}_{\text{ind}}$  do
12:      Compute  $\mu_{f_i \rightarrow x_s}^{(\tau)}$  using (19a)*, (19b)*
13:    end for
14:    for each  $x_s \in \mathcal{X}$  do
15:      Compute  $\mu_{x_s \rightarrow f_i}^{(\tau)}$  using (14)
16:    end for
17:  end while
18: end procedure
19: procedure OUTPUT
20:  for each  $x_s \in \mathcal{X}$  do
21:    Compute  $\hat{x}_s, v_{x_s}$  using (21)
22:  end for
23: end procedure

```

*Incoming messages are obtained in previous iteration $\tau - 1$

Example 2 (Different types of factor nodes). *In this example, we consider the bus/branch model with four measurements illustrated in Fig. 5a that we use to describe different types of factor nodes. The corresponding factor graph is given in Fig. 5b, with indirect factor nodes (red squares), direct factor nodes (orange squares), the initialization factor node (green square) and the virtual factor node (blue square).* \triangle

The DC-BP algorithm is presented in Algorithm 1, and it is an instance of a loopy GBP applied over a linear model defined by linear measurement functions $\mathbf{h}(\mathbf{x})$. It is well known that, if loopy GBP applied over a linear model converges, it will converge to a fixed point representing a solution of an equivalent WLS problem (12) [35]. Unlike means, the variances of GBP messages need not converge to

¹We note that each variable node in the initialization step of the algorithm has the corresponding local factor node attached. Local factor nodes only send, but do not receive, the message to/from incident variable nodes. Direct and virtual factor nodes repeatedly transmit the same message to the corresponding variable node throughout BP iterations.

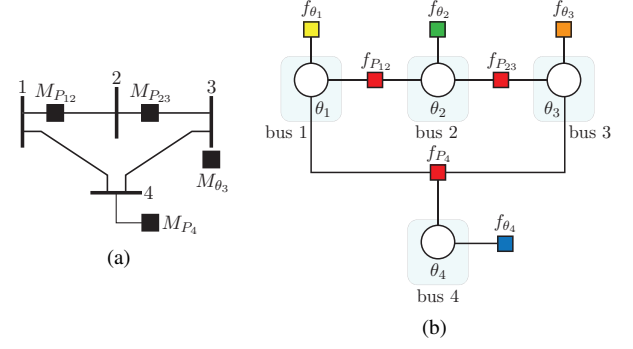


Fig. 5. Transformation of the bus/branch model and measurement configuration (subfigure a) into the corresponding factor graph with different types of factor nodes for the DC model (subfigure b)

correct values. In the following, it is sufficient to investigate the conditions for the DC-BP convergence as in case of convergence, the obtained mean values are the WLS solution.

D. Convergence of DC-BP Algorithm

In this part, we present convergence analysis of DC-BP algorithm with synchronous scheduling. In the following, it will be useful to consider a subgraph of the factor graph that contains the set of variable nodes $\mathcal{X} = \{x_1, \dots, x_N\}$, the set of indirect factor nodes $\mathcal{F}_{\text{ind}} = \{f_1, \dots, f_m\} \subset \mathcal{F}$, and a set of edges $\mathcal{B} \subseteq \mathcal{X} \times \mathcal{F}_{\text{ind}}$ connecting them. The number of edges in this subgraph is $b = |\mathcal{B}|$. Within the subgraph, we will consider a factor node $f_i \in \mathcal{F}_{\text{ind}}$ connected to its neighboring set of variable nodes $\mathcal{X}_i = \{x_q, \dots, x_Q\} \subset \mathcal{X}$ by a set of edges $\mathcal{B}_i = \{b_i^q, \dots, b_i^Q\} \subset \mathcal{B}$, where $d_i = |\mathcal{X}_i|$ is the degree of f_i . Next, we provide results on convergence of both variances and means of DC-BP messages, respectively.

Convergence of the Variances: From equations (14b) and (19b), we note that the evolution of the variances is independent of mean values of messages and measurements. Let $\mathbf{v}_s \in \mathbb{R}^b$ denote a vector of variance values of messages from indirect factor nodes \mathcal{F}_{ind} to variable nodes \mathcal{X} . Note that this vector can be decomposed as:

$$\mathbf{v}_s^{(\tau)} = [\mathbf{v}_{s,1}^{(\tau)}, \dots, \mathbf{v}_{s,m}^{(\tau)}]^T, \quad (22)$$

where the i -th element $\mathbf{v}_{s,i} \in \mathbb{R}^{d_i}$ is equal to:

$$\mathbf{v}_{s,i}^{(\tau)} = [v_{f_i \rightarrow x_q}^{(\tau)}, \dots, v_{f_i \rightarrow x_Q}^{(\tau)}]. \quad (23)$$

Substituting (14b) in (19b), the evolution of variances \mathbf{v}_s is equivalent to the following iterative equation:

$$\mathbf{v}_s^{(\tau)} = f(\mathbf{v}_s^{(\tau-1)}). \quad (24)$$

More precisely, using simple matrix algebra, one can obtain the evolution of the variances \mathbf{v}_s in the following matrix form:

$$\mathbf{v}_s^{(\tau)} = \left[(\tilde{\mathbf{C}}^{-1} \mathbf{\Pi} \tilde{\mathbf{C}}) \cdot (\mathfrak{D}(\mathbf{A}))^{-1} + \mathbf{\Sigma}_a \tilde{\mathbf{C}}^{-1} \right] \mathbf{i} \quad (25)$$

where

$$\tilde{\mathbf{C}} = \mathbf{C} \mathbf{C}^T \quad (26a)$$

$$\mathbf{A} = \mathbf{\Gamma} \mathbf{\Sigma}_s^{-1} \mathbf{\Gamma}^T + \mathbf{L}. \quad (26b)$$

Note that in (25), the dependance on $\mathbf{v}_s^{(\tau-1)}$ is hidden in matrix \mathbf{A} , or more precisely, in matrix Σ_s . Next, we briefly describe both the matrices and matrix-operators involved in (25).

The operator $\mathfrak{D}(\mathbf{A}) \equiv \text{diag}(A_{11}, \dots, A_{bb})$, where A_{ii} is the i -th diagonal entry of the matrix \mathbf{A} . The unit vector \mathbf{i} is of dimension b and is equal to $\mathbf{i} = [1, \dots, 1]^T$. The diagonal matrix Σ_s is obtained as $\Sigma_s = \text{diag}(\mathbf{v}_s^{(\tau-1)}) \in \mathbb{R}^{b \times b}$.

The matrix $\mathbf{C} = \text{diag}(\mathbf{C}_1, \dots, \mathbf{C}_m) \in \mathbb{R}^{b \times b}$ contains diagonal entries of the Jacobian non-zero elements, where i -th element $\mathbf{C}_i = [C_{x_q}, \dots, C_{x_Q}] \in \mathbb{R}^{d_i}$. The matrix $\Sigma_a = \text{diag}(\Sigma_{a,1}, \dots, \Sigma_{a,m}) \in \mathbb{R}^{b \times b}$ contains indirect factor node variances, with the i -th entry $\Sigma_{a,i} = [v_i, \dots, v_i] \in \mathbb{R}^{d_i}$.

The matrix $\mathbf{L} = \text{diag}(\mathbf{L}_1, \dots, \mathbf{L}_m) \in \mathbb{R}^{b \times b}$ contains inverse variances from singly-connected factor nodes to a variable node, if such nodes exist, where the i -th element $\mathbf{L}_i = [l_{x_q}, \dots, l_{x_Q}] \in \mathbb{R}^{d_i}$. For example, l_{x_q} equals:

$$l_{x_q} = \begin{cases} \frac{1}{v_{f_{d,q} \rightarrow x_q}}, & \text{if } x_q \text{ is incident to } f_{d,q} \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

The matrix $\mathbf{\Pi} = \text{diag}(\mathbf{\Pi}_1, \dots, \mathbf{\Pi}_m) \in \mathbb{F}_2^{b \times b}$, $\mathbb{F}_2 = \{0, 1\}$, is a block-diagonal matrix in which the i -th element is a block matrix $\mathbf{\Pi}_i = \mathbf{1}_i - \mathbf{I}_i \in \mathbb{F}_2^{d_i \times d_i}$, where the matrix $\mathbf{1}_i$ is $d_i \times d_i$ block matrix of ones, and \mathbf{I}_i is $d_i \times d_i$ identity matrix. The matrix $\mathbf{\Gamma} \in \mathbb{F}_2^{b \times b}$ is of the following block structure:

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{0}_{1,1} & \mathbf{\Gamma}_{1,2} & \dots & \mathbf{\Gamma}_{1,m} \\ \mathbf{\Gamma}_{2,1} & \mathbf{0}_{2,2} & \dots & \mathbf{\Gamma}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Gamma}_{m,1} & \mathbf{\Gamma}_{m,2} & \dots & \mathbf{0}_{m,m} \end{pmatrix}, \quad (28)$$

where $\mathbf{0}_{i,i}$ is a block matrix $d_i \times d_i$ of zeros, and $\mathbf{\Gamma}_{i,j} \in \mathbb{F}_2^{d_i \times d_j}$ with the (i, j) -th entry:

$$\mathbf{\Gamma}_{i,j}(i, j) = \begin{cases} 1, & \text{if both } b_i^q \text{ and } b_j^q \text{ are incident to } x_q \\ 0, & \text{otherwise.} \end{cases} \quad (29)$$

Note that the following holds: $\mathbf{\Gamma}_{j,i} = \mathbf{\Gamma}_{i,j}^T$.

Theorem 1. *The variances \mathbf{v}_s from indirect factor nodes to variable nodes always converge to a unique fixed point $\lim_{\tau \rightarrow \infty} \mathbf{v}_s^{(\tau)} = \hat{\mathbf{v}}_s$ for any initial point $\mathbf{v}_s^{(\tau=0)} > 0$.*

Proof. The theorem can be proved by showing that $f(\mathbf{v}_s)$ satisfies the conditions of the so-called standard function [36], following similar steps as in the proof of Lemma 1 in [37]. \square

Convergence of the Means: Equations (14a) and (19a) show that the evolution of the mean values depends on the variance values. Due to Theorem 1, it is possible to simplify evaluation of mean values \mathbf{z}_s from indirect factor nodes \mathcal{F}_{ind} to variable nodes \mathcal{X} by using the fixed-point values of $\hat{\mathbf{v}}_s$. The evolution of means \mathbf{z}_s becomes a set of linear equations:

$$\mathbf{z}_s^{(\tau)} = \tilde{\mathbf{z}} - \mathbf{\Omega} \mathbf{z}_s^{(\tau-1)}, \quad (30)$$

where

$$\tilde{\mathbf{z}} = \mathbf{C}^{-1} \mathbf{z}_a - \mathbf{D} \cdot (\mathfrak{D}(\hat{\mathbf{A}}))^{-1} \cdot \mathbf{L} \mathbf{z}_b \quad (31a)$$

$$\mathbf{\Omega} = \mathbf{D} \cdot (\mathfrak{D}(\hat{\mathbf{A}}))^{-1} \cdot \mathbf{\Gamma} \hat{\Sigma}_s^{-1} \quad (31b)$$

$$\hat{\mathbf{A}} = \mathbf{\Gamma} \hat{\Sigma}_s^{-1} \mathbf{\Gamma}^T + \mathbf{L} \quad (31c)$$

$$\mathbf{D} = \mathbf{C}^{-1} \mathbf{\Pi} \mathbf{C}. \quad (31d)$$

Note that the vector of means $\mathbf{z}_s \in \mathbb{R}^b$ can be decomposed as:

$$\mathbf{z}_s^{(\tau)} = [\mathbf{z}_{s,1}^{(\tau)}, \dots, \mathbf{z}_{s,m}^{(\tau)}]^T, \quad (32)$$

where the i -th element $\mathbf{z}_{s,i} \in \mathbb{R}^{d_i}$ is equal to:

$$\mathbf{z}_{s,i}^{(\tau)} = [z_{f_i \rightarrow x_k}^{(\tau)}, \dots, z_{f_i \rightarrow x_K}^{(\tau)}]. \quad (33)$$

The vector $\mathbf{z}_a = [\mathbf{z}_{a,1}, \dots, \mathbf{z}_{a,m}]^T \in \mathbb{R}^b$ contains means of indirect factor nodes, where $\mathbf{z}_{a,i} = [z_i, \dots, z_i] \in \mathbb{R}^{d_i}$. The diagonal matrix $\hat{\Sigma}_s \in \mathbb{R}^{b \times b}$ is obtained as $\hat{\Sigma}_s = \lim_{\tau \rightarrow \infty} \Sigma_s^{(\tau)}$. The vector $\mathbf{z}_b = [\mathbf{z}_{b,1}, \dots, \mathbf{z}_{b,m}] \in \mathbb{R}^b$ contains means from direct and virtual factor nodes to a variable node, if such nodes exist, where the i -th element $\mathbf{z}_{b,i} = [z_{x_k}, \dots, z_{x_K}] \in \mathbb{R}^{d_i}$. For example, the element z_{x_k} of $\mathbf{z}_{b,i}$ is equal to:

$$z_{x_k} = \begin{cases} z_{f_{d,k} \rightarrow x_k}, & \text{if } x_k \text{ is incident to } f_{d,k} \\ 0, & \text{otherwise.} \end{cases} \quad (34)$$

Theorem 2. *The means \mathbf{z}_s from indirect factor nodes to variable nodes converge to a unique fixed point $\lim_{\tau \rightarrow \infty} \mathbf{z}_s^{(\tau)} = \hat{\mathbf{z}}_s$:*

$$\hat{\mathbf{z}}_s = (\mathbf{I} + \mathbf{\Omega})^{-1} \tilde{\mathbf{z}}, \quad (35)$$

for any initial point $\mathbf{z}_s^{(\tau=0)}$ if and only if the spectral radius $\rho(\mathbf{\Omega}) < 1$.

Proof. The proof steps follow the proof of Theorem 5.2, [36]. \square

To summarize, the convergence of the DC-BP algorithm depends on the spectral radius of the matrix:

$$\mathbf{\Omega} = (\mathbf{C}^{-1} \mathbf{\Pi} \mathbf{C}) \cdot [\mathfrak{D}(\mathbf{\Gamma} \hat{\Sigma}_s^{-1} \mathbf{\Gamma}^T + \mathbf{L})]^{-1} \cdot (\mathbf{\Gamma} \hat{\Sigma}_s^{-1}). \quad (36)$$

If the spectral radius $\rho(\mathbf{\Omega}) < 1$, the DC-BP algorithm will converge and the resulting vector of mean values will be equal to the solution of the MAP estimator.

E. Convergence of DC-BP with Randomized Damping

In this section, we propose an improved DC-BP algorithm that applies synchronous scheduling with randomized damping. Several previous works reported that damping the BP messages improves the convergence of BP [37], [38]. Here, we propose a different randomized damping approach, where each mean value message from indirect factor node to a variable node is damped independently with probability p , otherwise, the message is calculated as in the standard DC-BP algorithm. The damped message is evaluated as a linear combination of the message from the previous and the current iteration, with weights α_1 and $1 - \alpha_1$, respectively. In Section VI, we demonstrate that the DC-BP with randomized damping dramatically improves convergence as compared to the standard DC-BP.

In the proposed damping, the equation (30) is redefined as:

$$\mathbf{z}_d^{(\tau)} = \mathbf{z}_q^{(\tau)} + \alpha_1 \mathbf{z}_r^{(\tau-1)} + \alpha_2 \mathbf{z}_r^{(\tau)}, \quad (37)$$

where $0 < \alpha_1 < 1$ is the weighting coefficient, and $\alpha_2 = 1 - \alpha_1$. In the above expression, $\mathbf{z}_q^{(\tau)}$ and $\mathbf{z}_r^{(\tau)}$ are obtained as:

$$\mathbf{z}_q^{(\tau)} = \mathbf{Q}\tilde{\mathbf{z}} - \mathbf{Q}\Omega\mathbf{z}_s^{(\tau-1)} \quad (38a)$$

$$\mathbf{z}_r^{(\tau)} = \mathbf{R}\tilde{\mathbf{z}} - \mathbf{R}\Omega\mathbf{z}_s^{(\tau-1)}, \quad (38b)$$

where diagonal matrices $\mathbf{Q} \in \mathbb{F}_2^{b \times b}$ and $\mathbf{R} \in \mathbb{F}_2^{b \times b}$ are defined as $\mathbf{Q} = \text{diag}(1 - q_1, \dots, 1 - q_b)$, $q_i \sim \text{Ber}(p)$, and $\mathbf{R} = \text{diag}(q_1, \dots, q_b)$, respectively, where $\text{Ber}(p) \in \{0, 1\}$ is a Bernoulli random variable with probability p independently sampled for each mean value message.

Substituting (38a) and (38b) in (37), we obtain:

$$\mathbf{z}_d^{(\tau)} = (\mathbf{Q} + \alpha_2\mathbf{R})\tilde{\mathbf{z}} - (\mathbf{Q}\Omega + \alpha_2\mathbf{R}\Omega - \alpha_1\mathbf{R})\mathbf{z}_s^{(\tau-1)}. \quad (39)$$

Note that $\mathbf{z}_r^{(\tau-1)} = \mathbf{R}\mathbf{z}_s^{(\tau-1)}$. In a more compact form, equation (39) can be written as follows:

$$\mathbf{z}_d^{(\tau)} = \bar{\mathbf{z}} - \bar{\Omega}\mathbf{z}_s^{(\tau-1)}, \quad (40)$$

where

$$\bar{\mathbf{z}} = (\mathbf{Q} + \alpha_2\mathbf{R})\tilde{\mathbf{z}} \quad (41a)$$

$$\bar{\Omega} = \mathbf{Q}\Omega + \alpha_2\mathbf{R}\Omega - \alpha_1\mathbf{R}. \quad (41b)$$

Theorem 3. *The means \mathbf{z}_d from indirect factor nodes to variable nodes converge to a unique fixed point $\hat{\mathbf{z}}_d = \lim_{\tau \rightarrow \infty} \mathbf{z}_d^{(\tau)}$ for any initial point $\mathbf{z}_d^{(\tau=0)}$ if and only if the spectral radius $\rho(\bar{\Omega}) < 1$. For the resulting fixed point, it holds that $\hat{\mathbf{z}}_d = \hat{\mathbf{z}}_s$.*

Proof. To prove theorem it is sufficient to show that equation (40) converges to the fixed point defined in (35). We can write:

$$\mathbf{z}_r^{(\tau-1)} = \mathbf{R}\tilde{\mathbf{z}} - \mathbf{R}\Omega\mathbf{z}_s^{(\tau-2)}. \quad (42)$$

Substituting (38a), (38b) and (42) in (37):

$$\begin{aligned} \mathbf{z}_d^{(\tau)} &= (\mathbf{Q} + \alpha_2\mathbf{R} + \alpha_1\mathbf{R})\tilde{\mathbf{z}} \\ &\quad - (\mathbf{Q}\Omega + \alpha_2\mathbf{R}\Omega)\mathbf{z}_s^{(\tau-1)} - \alpha_1\mathbf{R}\Omega\mathbf{z}_s^{(\tau-2)}. \end{aligned} \quad (43)$$

The fixed point $\hat{\mathbf{z}}_d = \lim_{\tau \rightarrow \infty} \mathbf{z}_d^{(\tau)}$ is equal to:

$$\begin{aligned} \hat{\mathbf{z}}_d &= (\mathbf{I} + \mathbf{Q}\Omega + \alpha_2\mathbf{R}\Omega + \alpha_1\mathbf{R}\Omega)^{-1} \\ &\quad \cdot (\mathbf{Q} + \alpha_2\mathbf{R} + \alpha_1\mathbf{R})\tilde{\mathbf{z}}. \end{aligned} \quad (44)$$

From definitions of \mathbf{Q} , \mathbf{R} and α_2 , we have $\mathbf{Q}\Omega + \alpha_2\mathbf{R}\Omega + \alpha_1\mathbf{R}\Omega = \Omega$ and $\mathbf{Q} + \alpha_2\mathbf{R} + \alpha_1\mathbf{R} = \mathbf{I}$, thus (44) becomes:

$$\hat{\mathbf{z}}_d = (\mathbf{I} + \Omega)^{-1}\tilde{\mathbf{z}}. \quad (45)$$

This concludes the proof. \square

V. BP-BASED DISTRIBUTED GAUSS-NEWTON METHOD FOR AC SE

The rest of the paper focuses on the more challenging AC model, where our goal is to derive the BP-based AC SE algorithm. However, due to non-linearity of measurement functions, the closed-form expressions for certain classes of BP messages cannot be obtained. Using approximations, we

derive the AC-BP algorithm as an (approximate) BP solution for the AC SE problem. Due to space constraints, we refer the reader to Appendix C, for a detailed derivation of the AC-BP algorithm. Unfortunately, due to approximations, the AC-BP algorithm does not match the performance of the centralized AC SE based on Gauss-Newton method (as we show in Section VI).

As the main contribution of this paper, we adopt different methodology to derive efficient BP-based AC SE method. Namely, instead of using the AC-BP method, we present the solution where BP is applied sequentially over the AC model, akin to what is done by the Gauss-Newton method. The resulting GN-BP algorithm represents a BP counterpart of the Gauss-Newton method, achieving exactly the same accuracy as the Gauss Newton method.

A. Gauss-Newton Method as a Sequential MAP Problem

Consider the Gauss-Newton method (11) where, at each iteration step ν , the algorithm returns a new estimate of \mathbf{x} denoted as $\mathbf{x}^{(\nu)}$. Note that, after a given iteration, an estimate $\mathbf{x}^{(\nu)}$ is a vector of known (constant) values. If the Jacobian matrix $\mathbf{J}(\mathbf{x}^{(\nu)})$ has a full column rank, the equation (11a) represents the linear WLS solution of the minimization problem [39]:

$$\min_{\Delta\mathbf{x}^{(\nu)}} \|\mathbf{W}^{1/2}[\mathbf{r}(\mathbf{x}^{(\nu)}) - \mathbf{J}(\mathbf{x}^{(\nu)})\Delta\mathbf{x}^{(\nu)}]\|_2^2. \quad (46)$$

Hence, at each iteration ν , the Gauss-Newton method produces WLS solution of the following system of linear equations:

$$\mathbf{r}(\mathbf{x}^{(\nu)}) = \mathbf{g}(\Delta\mathbf{x}^{(\nu)}) + \mathbf{u}, \quad (47)$$

where $\mathbf{g}(\Delta\mathbf{x}^{(\nu)}) = \mathbf{J}(\mathbf{x}^{(\nu)})\Delta\mathbf{x}^{(\nu)}$ comprises linear functions, while \mathbf{u} is the vector of measurement errors. The equation (11a) is the weighted normal equation for the minimization problem defined in (46), or alternatively, equation (11a) is a WLS solution of (47). Consequently, the probability density function associated with the i -th measurement (i.e., the i -th residual component r_i) at any iteration step ν :

$$\begin{aligned} \mathcal{N}(r_i(\mathbf{x}^{(\nu)})|\Delta\mathbf{x}^{(\nu)}, v_i) \\ = \frac{1}{\sqrt{2\pi}v_i} \exp \left\{ -\frac{[r_i(\mathbf{x}^{(\nu)}) - g_i(\Delta\mathbf{x}^{(\nu)})]^2}{2v_i} \right\}. \end{aligned} \quad (48)$$

The MAP solution of (9) can be redefined as an iterative optimization problem where, instead of solving (11), we solve the MAP (sub)problem:

$$\begin{aligned} \Delta\hat{\mathbf{x}}^{(\nu)} &= \arg \max_{\Delta\mathbf{x}^{(\nu)}} \mathcal{L}(\mathbf{r}(\mathbf{x}^{(\nu)})|\Delta\mathbf{x}^{(\nu)}) \\ &= \arg \max_{\Delta\mathbf{x}^{(\nu)}} \prod_{i=1}^k \mathcal{N}(r_i(\mathbf{x}^{(\nu)})|\Delta\mathbf{x}^{(\nu)}, v_i) \end{aligned} \quad (49a)$$

$$\mathbf{x}^{(\nu+1)} = \mathbf{x}^{(\nu)} + \Delta\hat{\mathbf{x}}^{(\nu)}. \quad (49b)$$

Next, we show that the solution of the above MAP sub-problem (49a) over increment variables $\Delta\mathbf{x}^{(\nu)}$ can be efficiently obtained using the BP algorithm applied over the underlying factor graph.

Note that, in general, the resulting BP algorithm represents an instance of a loopy GBP over a linear model defined by linear functions $\mathbf{g}(\Delta \mathbf{x}^{(\nu)})$. Thus if it converges, it provides a solution equal to the linear WLS solution $\Delta \mathbf{x}^{(\nu)}$ of (11a). The BP solution of $\Delta \hat{\mathbf{x}}^{(\nu)}$ in each iteration ν (outer iteration loop) is obtained via iterative BP algorithm (inner iteration loops). Every inner BP iteration $\tau = \{0, 1, 2, \dots, \eta(\nu)\}$ outputs $\Delta \hat{\mathbf{x}}^{(\nu, \tau)}$, where $\eta(\nu)$ is the number of inner BP iterations within the outer iteration ν .

B. The Factor Graph Construction

From the factorization of the likelihood expression (49a), one easily obtains the factor graph corresponding to the GN-BP method as follows. The increments $\Delta \mathbf{x}$ of state variables \mathbf{x} determine the set of variable nodes $\mathcal{X} = \{(\Delta \theta_1, \Delta V_1), \dots, (\Delta \theta_N, \Delta V_N)\}$ and each likelihood function $\mathcal{N}(r_i(\mathbf{x}^\nu) | \Delta \mathbf{x}^\nu, v_i)$ represents the local function $\psi(\mathcal{X}_i)$. Since the residual equals $r_i(\mathbf{x}^\nu) = z_i - h_i(\mathbf{x}^\nu)$, in general, the set of factor nodes $\mathcal{F} = \{f_1, \dots, f_k\}$ is defined by the set of measurements \mathcal{M} . The factor node f_i connects to the variable node $\Delta x_s \in \{\Delta \theta_s, \Delta V_s\}$ iff the increment of the state variable Δx_s is an argument of the corresponding function $g_i(\Delta \mathbf{x})$, i.e., if the state variable $x_s \in \{\theta_s, V_s\}$ is an argument of the measurement function $h_i(\mathbf{x})$.

C. Derivation of BP Messages

Message from a variable node to a factor node: It is easy to show that the message $\mu_{\Delta x_s \rightarrow f_i}(\Delta x_s)$ is proportional to:

$$\mu_{\Delta x_s \rightarrow f_i}(\Delta x_s) \propto \mathcal{N}(r_{\Delta x_s \rightarrow f_i} | \Delta x_s, v_{\Delta x_s \rightarrow f_i}), \quad (50)$$

with mean $r_{\Delta x_s \rightarrow f_i}$ and variance $v_{\Delta x_s \rightarrow f_i}$ obtained as:

$$r_{\Delta x_s \rightarrow f_i} = \left(\sum_{f_a \in \mathcal{F}_s \setminus f_i} \frac{r_{f_a \rightarrow \Delta x_s}}{v_{f_a \rightarrow \Delta x_s}} \right) v_{\Delta x_s \rightarrow f_i} \quad (51a)$$

$$\frac{1}{v_{\Delta x_s \rightarrow f_i}} = \sum_{f_a \in \mathcal{F}_s \setminus f_i} \frac{1}{v_{f_a \rightarrow \Delta x_s}}. \quad (51b)$$

Message from a factor node to a variable node: Similarly, the message $\mu_{f_i \rightarrow \Delta x_s}(\Delta x_s)$ is proportional to:

$$\mu_{f_i \rightarrow \Delta x_s}(\Delta x_s) \propto \mathcal{N}(r_{f_i \rightarrow \Delta x_s} | \Delta x_s, v_{f_i \rightarrow \Delta x_s}), \quad (52)$$

with mean $r_{f_i \rightarrow \Delta x_s}$ and variance $v_{f_i \rightarrow \Delta x_s}$ obtained as:

$$r_{f_i \rightarrow \Delta x_s} = \frac{1}{C_{\Delta x_s}} \left(r_i - \sum_{\Delta x_b \in \mathcal{X}_i \setminus \Delta x_s} C_{\Delta x_b} \cdot r_{\Delta x_b \rightarrow f_i} \right) \quad (53a)$$

$$v_{f_i \rightarrow \Delta x_s} = \frac{1}{C_{\Delta x_s}^2} \left(v_i + \sum_{\Delta x_b \in \mathcal{X}_i \setminus \Delta x_s} C_{\Delta x_b}^2 \cdot v_{\Delta x_b \rightarrow f_i} \right). \quad (53b)$$

The coefficients $C_{\Delta x_p}$, $\Delta x_p \in \mathcal{X}_i$, are Jacobian elements of the measurement function associated with the factor node f_i (see Appendix A and D for details):

$$C_{\Delta x_p} = \frac{\partial h(x_s, x_l, \dots, x_L)}{\partial x_p}. \quad (54)$$

Marginal inference: It can be shown that the marginal of the state variable Δx_s , according to (4), is represented by the Gaussian function:

$$p(\Delta x_s) \propto \mathcal{N}(\Delta \hat{x}_s | \Delta x_s, v_{\Delta x_s}), \quad (55)$$

with mean $\Delta \hat{x}_s$ which represents the estimated value of the state variable increment Δx_s and variance $v_{\Delta x_s}$:

$$\Delta \hat{x}_s = \left(\sum_{f_c \in \mathcal{F}_s} \frac{r_{f_c \rightarrow \Delta x_s}}{v_{f_c \rightarrow \Delta x_s}} \right) v_{\Delta x_s} \quad (56a)$$

$$\frac{1}{v_{\Delta x_s}} = \sum_{f_c \in \mathcal{F}_s} \frac{1}{v_{f_c \rightarrow \Delta x_s}}. \quad (56b)$$

To summarize, the MAP sub-problem defined in (49a) can be efficiently solved using (51), (53) and (56).

D. Iterative GN-BP Algorithm

To introduce the GN-BP algorithm, we use the same classification of factor nodes and the same message-passing schedule as in Section IV-C for the DC-BP algorithm. The slack factor node is specified as a node where the voltage angle has a given value, therefore, the residual of the corresponding state variable is equal to zero, and its variance tends to zero. Residuals of initialization and virtual factor nodes approach zero, while their variances tend to infinity. It is important to note that local factor nodes send messages represented by a triplet: mean (of the residual), variance and the state variable value (see Appendix D, for details).

The GN-BP algorithm is presented in Algorithm 2. After the initialization, the outer loop starts by computing residuals for direct and indirect factor nodes, as well as the Jacobian elements, and passes them to the inner iteration loop. The inner iteration loop represents the main algorithm routine which includes BP-based message inference described in the previous subsection. The output of the inner iteration loop is the estimate of the state variable increments. Finally, the outer loop updates the set of state variables. The outer loop iterations are repeated until the stopping criteria is met.

E. Convergence of the GN-BP Algorithm

The GN-BP in the inner iteration loop solves the system of linear equations (47). By directly applying the results from Section IV-D of the (linear) DC-BP convergence analysis, the convergence of the GN-BP with synchronous scheduling in each outer iteration loop ν depends on the spectral radius of the matrix:

$$\mathbf{\Omega}(\mathbf{x}^{(\nu)}) = [\mathbf{C}(\mathbf{x}^{(\nu)})^{-1} \mathbf{\Pi} \mathbf{C}(\mathbf{x}^{(\nu)})] \cdot [\mathbf{\mathcal{D}}(\mathbf{\Gamma} \hat{\mathbf{\Sigma}}^{-1} \mathbf{\Gamma}^T + \mathbf{L})]^{-1} \cdot (\mathbf{\Gamma} \hat{\mathbf{\Sigma}}^{-1}), \quad (57)$$

where $\mathbf{C}(\mathbf{x}^{(\nu)})$ contains diagonal entries of the Jacobian $\mathbf{J}(\mathbf{x}^{(\nu)})$ non-zero elements in every outer iteration ν . According to Theorem 2, the GN-BP with synchronous scheduling converges to a unique fixed point if $\rho(\mathbf{\Omega}(\mathbf{x}^{(\nu)})) < 1$, $\nu = \{0, 1, 2, \dots\}$. We note that this

Algorithm 2 The GN-BP

```

1: procedure INITIALIZATION  $\nu = 0$ 
2:   for Each  $x_s \in \mathcal{X}$  do
3:     Initialize value of  $x_s^{(0)}$ , typically as a "flat start"
4:   end for
5: end procedure
6: procedure OUTER ITERATION LOOP  $\nu = 0, 1, 2, \dots; \tau = 0$ 
7:   while stopping criterion for the outer loop is not met do
8:     for Each  $f_s \in \mathcal{F}_{dir}$  do
9:       compute residual  $r_s^{(\nu)} = z_s - x_s^{(\nu)}$ 
10:    end for
11:    for Each  $f_s \in \mathcal{F}_{loc}$  do
12:      send  $\mu_{f_s \rightarrow \Delta x_s}^{(\nu)}, x_s^{(\nu)}$  to incident  $\Delta x_s \in \mathcal{X}$ 
13:    end for
14:    for Each  $\Delta x_s \in \mathcal{X}$  do
15:      send  $\mu_{\Delta x_s \rightarrow f_i}^{(\nu)(\tau=0)} = \mu_{f_s \rightarrow \Delta x_s}^{(\nu)}, x_s^{(\nu)}$  to incident  $f_i \in \mathcal{F}_{id}$ 
16:    end for
17:    for Each  $f_i \in \mathcal{F}_{ind}$  do
18:      compute residual  $r_i^{(\nu)} = z_i - h_i(\mathbf{x}^{(\nu)})$ 
19:      compute Jacobian elements  $C_{i,\Delta x_p}^{(\nu)}; \Delta x_p \in \mathcal{X}_i$ 
20:    end for
21:    procedure INNER ITERATION LOOP  $\tau = 1, 2, \dots$ 
22:      while stopping criterion for the inner loop is not met do
23:        for Each  $f_i \in \mathcal{F}_{ind}$  do
24:          Compute  $\mu_{f_i \rightarrow \Delta x_s}^{(\tau)}$  using (53)*
25:        end for
26:        for Each  $\Delta x_s \in \mathcal{X}$  do
27:          Compute  $\mu_{\Delta x_s \rightarrow f_i}^{(\tau)}$  using (51)
28:        end for
29:      end while
30:    end procedure
31:    for Each  $\Delta x_s \in \mathcal{X}$  do
32:      compute  $\hat{\Delta x}_s^{(\nu)}$  using (56)
33:      update  $x_s^{(\nu+1)} = x_s^{(\nu)} + \hat{\Delta x}_s^{(\nu)}$ 
34:    end for
35:  end while
36: end procedure

```

*Incoming messages are obtained in previous iteration $\tau - 1$

convergence condition is required to hold in every outer iteration ν .

Similarly, the convergence of the GN-BP with randomized damping in every outer iteration loop ν is governed by the spectral radius of the matrix:

$$\bar{\Omega}(\mathbf{x}^{(\nu)}) = \mathbf{Q}\Omega(\mathbf{x}^{(\nu)}) + \alpha_2 \mathbf{R}\Omega(\mathbf{x}^{(\nu)}) - \alpha_1 \mathbf{R}. \quad (58)$$

According to Theorem 3, in every outer iteration ν , the algorithm will converge to a unique fixed point if $\rho(\bar{\Omega}(\mathbf{x}^{(\nu)})) < 1$, $\nu = \{0, 1, 2, \dots\}$, and the resulting fixed point is equal to the fixed point obtained by the GN-BP with synchronous scheduling.

VI. NUMERICAL RESULTS

In this section, using numerical simulations, we analyze the convergence and the accuracy of the BP-based SE methods.

For convergence evaluation, we focus on the DC-BP algorithm whose convergence is analyzed in Sections IV-D and IV-E. We verify these results via numerical simulations.

Although similar results can be obtained for the GN-BP algorithm (see Section V-E), for brevity, we present the convergence results only for the DC-BP case.

For evaluation of accuracy, we focus on the proposed BP-based AC SE methods and compare them to the Gauss-Newton method (the centralized SE algorithm). More precisely, we first consider the IEEE 14 bus test case and provide a detailed performance comparison between the AC-BP and the GN-BP algorithms. Then, due to favorable performance, we provide an in-depth evaluation of the GN-BP algorithm using the IEEE 14 bus test case, providing the results for IEEE 30 and IEEE 118 bus test cases in Appendix F.

A. Simulation Setup

In the simulated model, we start with a given IEEE bus test case and apply the power flow analysis to generate the exact solution. In case of the DC model, we apply the DC power flow analysis to calculate voltage angles and active powers, while for the AC model, we apply the AC power flow analysis to generate currents, voltages and powers across the network. Further, we corrupt the exact solution by the additive white Gaussian noise of variance v and we observe the set of measurements.

In the DC model, the measurements contain active power flows and power injections, and bus voltage angles, while in the AC model, they contain active and reactive power flows, active and reactive power injections, bus voltage magnitudes and bus voltage angles. The set of measurements is selected in such a way that the system is observable. More precisely, for each scenario, we generate 1000 random measurement configurations with the number of measurements equal either to double or triple the size of the number of state variables (i.e., we consider the redundancy to be equal 2 or 3).

To evaluate the performance of the BP-based SE algorithms, we convert each of the above randomly generated IEEE bus test cases with a given measurement configuration into the corresponding factor graph and we run the DC-BP, AC-BP or GN-BP algorithms over the factor graph. Note that, in order to initialize the AC-BP, the GN-BP, and the Gauss-Newton method, we use the "flat start" assumption² ($V_i = 1$, $\theta_i = 0$, $i = 1, \dots, N$).

B. Convergence Analysis of DC-BP Algorithm

As detailed in Sec. IV-D and IV-E, the DC-BP with synchronous scheduling with or without randomized damping will converge if $\rho(\Omega) < 1$ and $\rho(\bar{\Omega}) < 1$, respectively. This condition is verified in our simulations, thus we present the convergence performance by comparing spectral radii of matrices Ω and $\bar{\Omega}$.

Figure 6 shows spectral radii for different redundancies for IEEE 14 and IEEE 118 bus test case. For each scenario, the randomized damping case behaves superior in terms of the

²Usually, IEEE bus test cases include in-phase transformers with zero series resistances. Consequently, the "flat start" will cause undefined messages due to the corresponding Jacobian elements being equal to zero. This problem can be alleviated by initializing bus voltage angles with a small random perturbation at transformers' buses.

spectral radius. As an interesting and somewhat extreme case, for the IEEE 118 bus test case, the DC-BP algorithm with synchronous scheduling could not converge at all, while with randomized damping³, we recorded convergence with probability above 0.9. As expected, the algorithm with randomized damping performs better for larger redundancy.

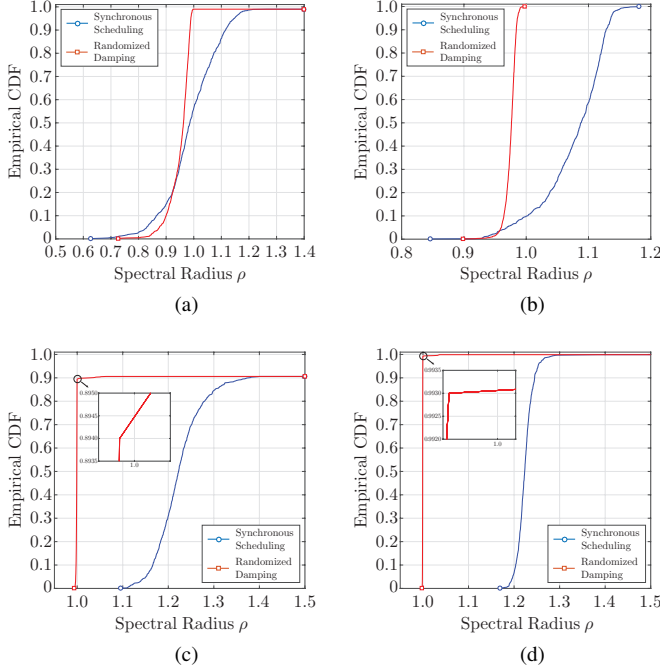


Fig. 6. The spectral radius of matrices Ω for synchronous scheduling and $\tilde{\Omega}$ for randomized damping for redundancy equal 2 for IEEE 14 (subfigure a) and IEEE 118 (subfigure c) bus test case and for redundancy equal 3 for IEEE 14 (subfigure b) and IEEE 118 (subfigure d) bus test case.

In scenarios when the standard DC-BP algorithm did not converge (i.e., $\rho(\Omega) > 1$), the algorithm with randomized damping had smaller spectral radius, i.e., $\rho(\tilde{\Omega}) < \rho(\Omega)$, and moreover, the spectral radius was usually smaller than one ($\rho(\tilde{\Omega}) < 1$). On the other hand, when DC-BP converged ($\rho(\Omega) < 1$), then in most cases the randomized damping method also converged, albeit with slightly higher spectral radius ($\rho(\Omega) < \rho(\tilde{\Omega}) < 1$). These results demonstrate the benefit of randomized damping for improved DC-BP stability and convergence⁴.

C. Performance Evaluation of AC-BP and GN-BP

In the following, compare the performance of the AC-BP and GN-BP algorithms via extensive numerical study (the AC-BP is presented in detail in Appendix B and C). We use the weighted residual sum of squares (WRSS) as a metric:

$$\text{WRSS} = \sum_{i=1}^k \frac{[z_i - h_i(\mathbf{x})]^2}{\sigma_i^2}. \quad (59)$$

³Note that randomized damping parameters, which are optimized via numerical experiments in Sec. VI-D, are set to $p = 0.6$ and $\alpha_1 = 0.5$.

⁴Another trick for improving the convergence behavior is to connect virtual factor nodes to all variable nodes that do not already have direct factor nodes.

Note that WRSS is the value of the objective function of the optimization problem (10) we are solving, thus it is suitable metric for the SE accuracy. Finally, we normalize the obtained WRSS by WRSS_{WLS} of the centralized SE obtained using the Gauss-Newton method after 12 iterations (which we adopt as a normalization constant). This way, we compare the accuracy of BP-based algorithms to the one of the centralized SE.

Besides WRSS, we use the mean absolute difference (MAD) between the state variables in two consecutive iterations:

$$\text{MAD} = \frac{1}{n} \sum_{i=1}^n |\Delta x_i|. \quad (60)$$

The MAD value represents average component-wise shift of the state estimate over the iterations, thus it may be used to quantify the rate of convergence.

The IEEE 14 bus test case with fixed measurement configuration containing 61 measurement devices, as shown in Fig. 7, is used to compare the performance of the BP-based SE algorithms. For each value of noise variance $v = \{v_1, v_2, v_3, v_4\} = \{10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}\}$ p.u., using Monte Carlo approach, we generate 1000 random sets of measurement values and feed them to the proposed BP-based SE algorithms.

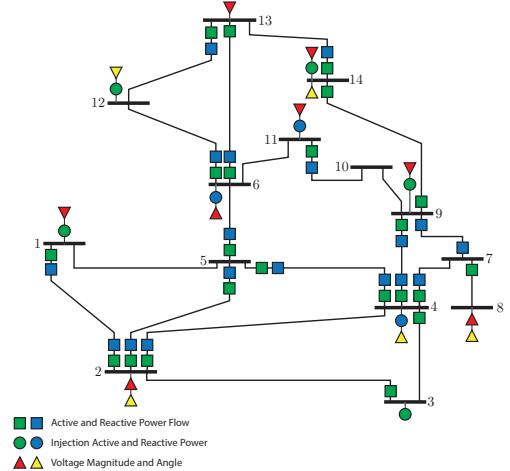


Fig. 7. The IEEE 14 bus test case with given measurement configuration.

For both algorithms, we apply randomized damping due to improved convergence, and initially, we set the randomized damping parameters to $p = 0.6$ and $\alpha_1 = 0.5$. The GN-BP is configured so that, for every outer iteration ν , the number of inner iterations grows exponentially as $\eta(\nu) = \nu^e$, where e is the inner iteration exponent which we set to $e = 4$ [40]⁵.

In order to present average performance results, we use box plots in which we: i) set the minimum and the maximum value to 0% and 90% of the obtained simulation results, ii) the box is defined by the first and the third quartile, iii) the middle line is the median value, and iv) the results above the maximum value are represented as outliers.

Fig. 8 shows the weighted residual sum of squares of the AC-BP $\text{WRSS}_{\text{BP}}^\tau$ over the iterations τ , normalized by

⁵Note that we perform detailed performance analysis as a function of damping parameters in [40], Appendix E, and as a function of the number of inner iterations in Section VI-D.

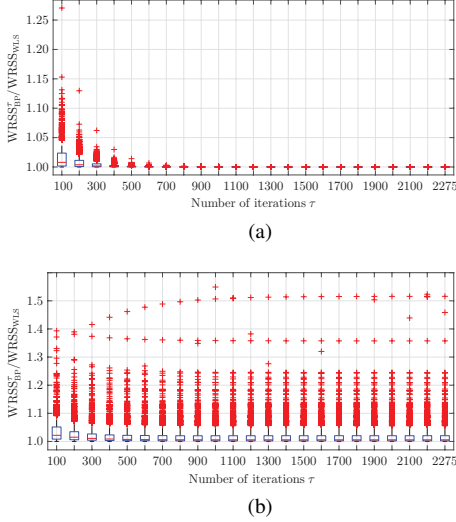


Fig. 8. The AC-BP normalized WRSS (i.e., $WRSS_{BP}^{\tau}/WRSS_{WLS}$) for the low noise level v_1 (subfigure a) and the high noise level v_4 (subfigure b).

$WRSS_{WLS}$ (i.e., $WRSS_{BP}^{\tau}/WRSS_{WLS}$). We observe that the AC-BP converges for both the low and the high noise level, however, for the high noise level, the solution of the AC-BP algorithm does not correspond to the solution of the centralized SE. This is expected, since as the noise variance increases, the accuracy of Gaussian approximation of the BP messages is decreasing, which affects the accuracy of the AC-BP solution.

Fig. 9 shows $WRSS_{BP}^{\nu}/WRSS_{WLS}$ of the GN-BP over the outer iterations ν . As shown, the GN-BP algorithm converges to the solution of the centralized SE for both the low and the high noise level.

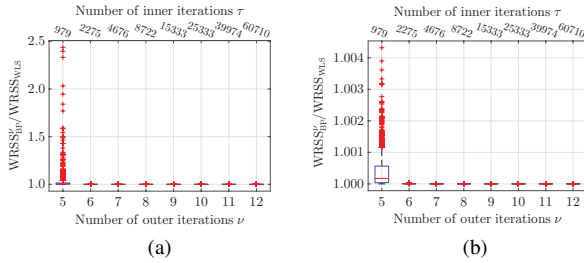


Fig. 9. The GN-BP normalized WRSS (i.e., $WRSS_{BP}^{\nu}/WRSS_{WLS}$) for the low noise level v_1 (subfigure a) and the high noise level v_4 (subfigure b).

To compare BP-based algorithms, we consider $WRSS_{BP}$ of the BP-based algorithms after the same total number of iterations. More precisely, the AC-BP algorithm is terminated after $\tau = 2275$ iterations, while the GN-BP is terminated after $\nu = 5$ outer iterations, which corresponds exactly to the total of 2275 inner iterations. Fig. 10 demonstrates that, for the low noise level, the AC-BP converges faster than the GN-BP. However, as the noise level increases, the GN-BP converges faster than the AC-BP.

To summarize, the GN-BP is able to reach exactly the same solution as the centralized SE, while the AC-BP generally fails to achieve it for higher noise levels. In

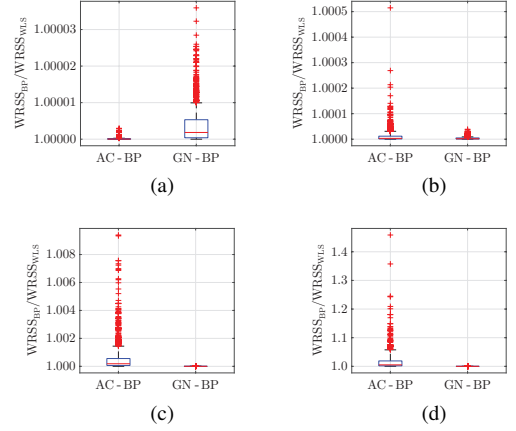


Fig. 10. The normalized WRSS for the AC-BP and GN-BP after 2275 iteration for variances v_1 (subfigure a), v_2 (subfigure b), v_3 (subfigure c) and v_4 (subfigure d).

addition, in terms of implementation, the AC-BP suffers two drawbacks as compared to the GN-BP: i) the AC-BP messages have considerably more complex form, and ii) the AC-BP requires prior knowledge (e.g., historical data). Consequently, in the remaining part of this section, we focus on the performance of the GN-BP.

D. Design of Inner Iteration Schemes

In order to reduce the number of non-converging simulations due to an insufficient number of inner iterations, in this section we test the GN-BP convergence performance with different inner iteration schemes. The damping parameters, investigated in detail in Appendix E, are set to $p = 0.4$ and $\alpha_1 = 0.3$. As a reference point, the MAD values of the centralized SE for variances v_1 and v_4 are shown in Fig. 11.

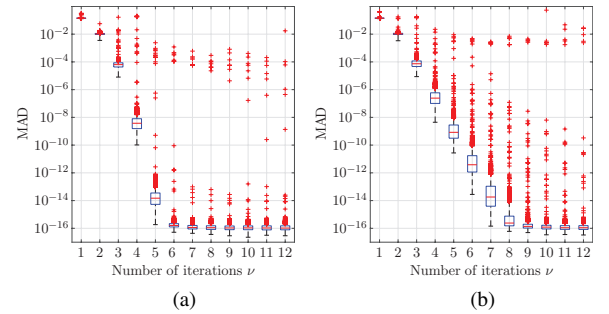


Fig. 11. The MAD values of the centralized SE for the low noise level v_1 (subfigure a) and high noise level v_4 (subfigure b).

For comparison, the MAD values of the GN-BP with the exponential inner iteration scheme ($\eta(\nu) = \nu^e$, $e = 4$) is shown in Fig. 12. Note that the GN-BP exhibits comparable convergence performance to the centralized SE algorithm. Note also that it is difficult to directly compare the two, due to a large difference in computational loads of a single (outer) iteration. For example, the complexity of a single iteration remains constant but significant (due to matrix

inversion) over iterations for the centralized SE algorithm, while it gradually increases for the GN-BP starting from an extremely low complexity at initial outer iterations.

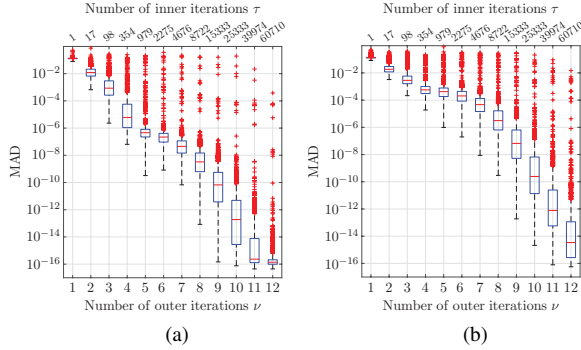


Fig. 12. The MAD values of the GN-BP with the exponential inner iteration scheme for the low noise level v_1 (subfigure a) and high noise level v_4 (subfigure b).

In the following, we define an alternative accuracy-based inner iteration scheme. Namely, instead of prescribing the number of inner iterations in advance, we let the number of inner iterations evolve until a certain accuracy-based criterion is met. More precisely, the algorithm in the inner iteration loop is running until the following convergence criterion is reached:

$$|\mathbf{r}_{f \rightarrow \Delta x}^{(\nu, \tau)} - \mathbf{r}_{f \rightarrow \Delta x}^{(\nu, \tau-1)}| < \epsilon(\nu), \quad (61)$$

where $\mathbf{r}_{f \rightarrow \Delta x}$ represents the vector of mean-value messages from factor nodes to variable nodes, and $\epsilon(\nu)$ is the threshold at iteration ν . Note that, due to the fact that the BP is a distributed algorithm, imposing this condition in real systems would require additional communication overhead.

Fig. 13 shows MAD values and the total number of inner iterations for the threshold $\epsilon(\nu) = [10^{-3}, 10^{-6}, \dots, 10^{-15}, 10^{-16}, \dots, 10^{-16}]$, where the number of outer iterations is $\nu = 1, \dots, 12$. Comparing the results with the centralized SE (Fig. 11), we note that the GN-BP has almost identical convergence rate. We note that the proposed accuracy-based inner iteration scheme dramatically reduces the number of non-converging simulations.

To summarize, we have demonstrated that by using different inner iteration schemes, one can influence and control the trade-off between the GN-BP convergence rate and the total number of inner iterations. For additional results on scalability and complexity of GN-BP, we refer the reader to Appendix F.

VII. CONCLUSIONS

In this paper, we presented an in-depth study of the application of the BP algorithm to the SE problem in power systems. We provided detailed derivation, convergence and performance analysis of BP-based SE algorithms for both DC and AC model. The main contribution of our study is the GN-BP algorithm, which is shown to represent a BP-based implementation of the iterative Gauss-Newton method. In a thorough numerical study, we have shown how

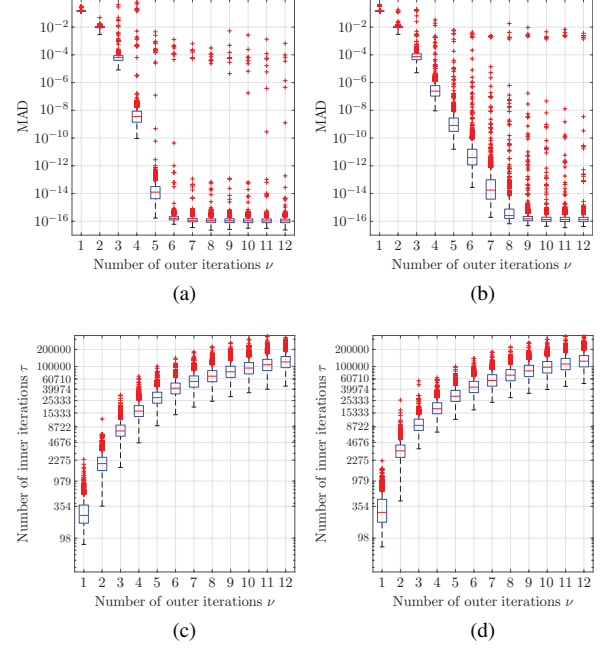


Fig. 13. The MAD values and the number of inner iterations for the low noise level v_1 (subfigure a and subfigure c) and the high noise level v_4 (subfigure b and subfigure d) of the GN-BP with threshold $\epsilon(\nu) = [10^{-3}, 10^{-6}, \dots, 10^{-15}, 10^{-16}, \dots, 10^{-16}]$.

to fine-tune the GN-BP implementation and balance between the accuracy and the convergence rate. In our future work, we will investigate extensions to asynchronous BP in dynamic and real-time state estimation, including suitable models for communication constraints.

APPENDIX A

MEASUREMENT FUNCTIONS AND JACOBIAN ELEMENTS

In the following, we present measurement functions and corresponding Jacobian elements for both, the AC and DC SE model. Note that, for simplicity, we assume that the model does not contain phase-shifting transformers.

AC SE Model

The **active** and **reactive power flow** at the branch $(i, j) \in \mathcal{E}$ that connects buses i and j :

$$h_{P_{ij}}(\cdot) = V_i^2(g_{ij} + g_{si}) - V_i V_j(g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (62a)$$

$$h_{Q_{ij}}(\cdot) = -V_i^2(b_{ij} + b_{si}) - V_i V_j(g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}), \quad (62b)$$

where V_i and V_j are bus voltage magnitudes, while $\theta_{ij} = \theta_i - \theta_j$ is the bus voltage angle difference between bus voltage angles at buses i and j . The parameters in above equations include the conductance g_{ij} and susceptance b_{ij} of the branch, as well as the conductance g_{si} and susceptance b_{si} of the branch shunt element connected at the bus i .

The Jacobian expressions corresponding to $h_{P_{ij}}(\cdot)$ are:

$$\begin{aligned}\frac{\partial h_{P_{ij}}(\cdot)}{\partial \theta_i} &= V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \\ \frac{\partial h_{P_{ij}}(\cdot)}{\partial \theta_j} &= -V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \\ \frac{\partial h_{P_{ij}}(\cdot)}{\partial V_i} &= -V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) + 2V_i (g_{ij} + g_{si}) \\ \frac{\partial h_{P_{ij}}(\cdot)}{\partial V_j} &= -V_i (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}).\end{aligned}$$

The Jacobian expressions corresponding to $h_{Q_{ij}}(\cdot)$ are as follows:

$$\begin{aligned}\frac{\partial h_{Q_{ij}}(\cdot)}{\partial \theta_i} &= -V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \\ \frac{\partial h_{Q_{ij}}(\cdot)}{\partial \theta_j} &= V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \\ \frac{\partial h_{Q_{ij}}(\cdot)}{\partial V_i} &= -V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) - 2V_i (b_{ij} + b_{si}) \\ \frac{\partial h_{Q_{ij}}(\cdot)}{\partial V_j} &= -V_i (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}).\end{aligned}$$

The **current magnitude** at the branch $(i, j) \in \mathcal{E}$ that connects buses i and j :

$$h_{I_{ij}}(\cdot) = [A_c V_i^2 + B_c V_j^2 - 2V_i V_j (C_c \cos \theta_{ij} - D_c \sin \theta_{ij})]^{1/2}. \quad (63)$$

$$\begin{aligned}A_c &= (g_{ij} + g_{si})^2 + (b_{ij} + b_{si})^2; & B_c &= g_{ij}^2 + b_{ij}^2 \\ C_c &= g_{ij} (g_{ij} + g_{si}) + b_{ij} (b_{ij} + b_{si}); & D_c &= g_{ij} b_{si} - b_{ij} g_{si}.\end{aligned}$$

The Jacobian expressions corresponding to current magnitude measurement function $h_{I_{ij}}(\cdot)$ are:

$$\begin{aligned}\frac{\partial h_{I_{ij}}(\cdot)}{\partial \theta_i} &= \frac{V_i V_j (D_c \cos \theta_{ij} + C_c \sin \theta_{ij})}{I_{ij}} \\ \frac{\partial h_{I_{ij}}(\cdot)}{\partial \theta_j} &= -\frac{V_i V_j (D_c \cos \theta_{ij} + C_c \sin \theta_{ij})}{I_{ij}} \\ \frac{\partial h_{I_{ij}}(\cdot)}{\partial V_i} &= \frac{V_j (D_c \sin \theta_{ij} - C_c \cos \theta_{ij}) + A_c V_i}{I_{ij}} \\ \frac{\partial h_{I_{ij}}(\cdot)}{\partial V_j} &= \frac{V_i (D_c \sin \theta_{ij} - C_c \cos \theta_{ij}) + B_c V_j}{I_{ij}}.\end{aligned}$$

The **active and reactive power injection** into the bus $i \in \mathcal{V}$:

$$h_{P_i}(\cdot) = V_i \sum_{j \in \mathcal{H}_i} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (64a)$$

$$h_{Q_i}(\cdot) = V_i \sum_{j \in \mathcal{H}_i} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}), \quad (64b)$$

where \mathcal{H}_i is the set of buses adjacent to the bus i , including the bus i . The parameters G_{ij} and B_{ij} are conductance and susceptance of the complex bus matrix [1, Sec. 2.3]. The

Jacobian expressions corresponding to $h_{P_i}(\cdot)$ are:

$$\begin{aligned}\frac{\partial h_{P_i}(\cdot)}{\partial \theta_i} &= V_i \sum_{j \in \mathcal{H}_i \setminus i} V_j (-G_{ij} \sin \theta_{ij} + B_{ij} \cos \theta_{ij}) \\ \frac{\partial h_{P_i}(\cdot)}{\partial \theta_j} &= V_i V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \\ \frac{\partial h_{P_i}(\cdot)}{\partial V_i} &= \sum_{j \in \mathcal{H}_i \setminus i} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) + 2V_i G_{ii} \\ \frac{\partial h_{P_i}(\cdot)}{\partial V_j} &= V_i (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}),\end{aligned}$$

where $\mathcal{H}_i \setminus i$ is the set of buses adjacent to the bus i . The Jacobian expressions corresponding to $h_{Q_i}(\cdot)$ are:

$$\begin{aligned}\frac{\partial h_{Q_i}(\cdot)}{\partial \theta_i} &= V_i \sum_{j \in \mathcal{H}_i \setminus i} V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ \frac{\partial h_{Q_i}(\cdot)}{\partial \theta_j} &= V_i V_j (-G_{ij} \cos \theta_{ij} - B_{ij} \sin \theta_{ij}) \\ \frac{\partial h_{Q_i}(\cdot)}{\partial V_i} &= \sum_{j \in \mathcal{H}_i \setminus i} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) - 2V_i B_{ii} \\ \frac{\partial h_{Q_i}(\cdot)}{\partial V_j} &= V_i (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}).\end{aligned}$$

The **voltage magnitude and angle** on the bus $i \in \mathcal{V}$:

$$h_{V_i}(\cdot) = V_i; \quad h_{\theta_i}(\cdot) = \theta_i. \quad (65)$$

The Jacobian expressions corresponding to measurements of voltage magnitude and voltage angle are as follows:

$$\begin{aligned}\frac{\partial h_{V_i}(\cdot)}{\partial V_i} &= 1; & \frac{\partial h_{V_i}(\cdot)}{\partial V_j} &= 0 \\ \frac{\partial h_{\theta_i}(\cdot)}{\partial \theta_i} &= 1; & \frac{\partial h_{\theta_i}(\cdot)}{\partial \theta_j} &= 0.\end{aligned}$$

DC SE Model

The **active power flow** at the branch $(i, j) \in \mathcal{E}$ that connects buses i and j :

$$h_{P_{ij}}(\cdot) = -b_{ij} (\theta_i - \theta_j). \quad (66)$$

The Jacobian expressions corresponding to $h_{P_{ij}}(\cdot)$ are as follows:

$$\frac{\partial h_{P_{ij}}(\cdot)}{\partial \theta_i} = -b_{ij}; \quad \frac{\partial h_{P_{ij}}(\cdot)}{\partial \theta_j} = b_{ij}.$$

The **active power injection** into the bus $i \in \mathcal{V}$ is:

$$h_{P_i}(\cdot) = - \sum_{j \in \mathcal{H}_i \setminus i} b_{ij} (\theta_i - \theta_j), \quad (67)$$

where $\mathcal{H}_i \setminus i$ is the set of buses adjacent to the bus i . The Jacobian expressions corresponding to $h_{P_i}(\cdot)$ are as follows:

$$\frac{\partial h_{P_i}(\cdot)}{\partial \theta_i} = - \sum_{j \in \mathcal{H}_i \setminus i} b_{ij}; \quad \frac{\partial h_{P_i}(\cdot)}{\partial \theta_j} = \sum_{j \in \mathcal{H}_i \setminus i} b_{ij}.$$

The **voltage angle** on the bus $i \in \mathcal{V}$:

$$h_{\theta_i}(\cdot) = \theta_i. \quad (68)$$

The Jacobian expressions corresponding to $h_{\theta_i}(\cdot)$ are as follows:

$$\frac{\partial h_{\theta_i}(\cdot)}{\partial \theta_i} = 1; \quad \frac{\partial h_{\theta_i}(\cdot)}{\partial \theta_j} = 0.$$

APPENDIX B

BP-BASED ALGORITHM FOR THE AC SE

The AC model is characterized by the set of state variables $\mathbf{x} \equiv (\boldsymbol{\theta}, \mathbf{V})$, while measurement functions are defined in Appendix A, equations (62)-(65).

The factor graph construction: According to (9), in the AC scenario, the set of state variables $\boldsymbol{\theta}$ and \mathbf{V} determines the set of variable nodes $\mathcal{X} = \{(\theta_1, V_1), \dots, (\theta_N, V_N)\}$, while the set of factor nodes $\mathcal{F} = \{f_1, \dots, f_k\}$ is defined by the set of measurements \mathcal{M} . A factor node f_i connects to a variable node $x_s \in \mathcal{X}$ if and only if the state variable x_s is an argument of the corresponding measurement function $h_i(\mathbf{x})$.

Example 3 (Constructing factor graph). *In this example, using a simple 3-bus model presented in Fig. 14a, we demonstrate the conversion from a bus/branch model with a given measurement configuration into the corresponding factor graph for the AC model. The variable nodes represent*

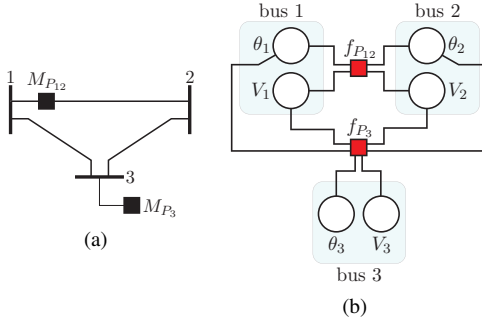


Fig. 14. Transformation of the bus/branch model and measurement configuration (subfigure a) into the corresponding factor graph for the DC model (subfigure b) and the AC model (subfigure c)

state variables $\mathcal{X} = \{\theta_1, V_1, \theta_2, V_2, \theta_3, V_3\}$. Factor nodes are defined by corresponding measurements, where in our example, measurements $M_{P_{12}}$ and M_{P_3} are mapped into factor nodes $\mathcal{F} = \{f_{P_{12}}, f_{P_3}\}$. \triangle

Message from a variable node to a factor node: Due to the fact that variable node output messages do not depend on measurement functions (see equation (2)), relations (13) and (14) hold for the AC-BP.

Message from a factor node to a variable node: Due to non-linear measurement functions $h_i(\cdot)$, the integral in (3) for the AC-BP cannot be evaluated in closed form. Consequently, the message from a factor node to a variable node will not be Gaussian. In the following, as an approximation, we assume that for the AC-BP, the message $\mu_{f_i \rightarrow x_s}(x_s)$ also has the Gaussian form (15). According to DC-BP we provide arguments that lead us to approximations used to derive messages for the AC-BP.

Mean value evaluation: The expression for the mean of the DC-BP $z_{f_i \rightarrow x_s}$ is exact and equals (19a). Although the expression (19a) is obtained by directly evaluating (3) for the linear DC model, we note that it has a useful interpretation via conditional expectation. For that purpose, let us define a vector $\mathbf{x}_b = \mathcal{X}_i \setminus x_s$, and let $\mathbf{z}_{\mathbf{x}_b \rightarrow f_i}$ denote a vector of mean values of messages from variable nodes $\mathcal{X}_i \setminus x_s$ to the factor node f_i . Then, the conditional expectation $\mathbb{E}[h_i(x_s, \mathbf{x}_b) | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}]$ can be calculated as:

$$\mathbb{E}[h_i(x_s, \mathbf{x}_b) | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] = C_{x_s} \mathbb{E}[x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + \sum_{x_b \in \mathcal{X}_i \setminus x_s} C_{x_b} z_{x_b \rightarrow f_i} = z_i. \quad (70)$$

From the BP perspective, the conditional expected value $\mathbb{E}[x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}]$ represents the mean $z_{f_i \rightarrow x_s}$. Hence, it is possible to define the conditional expectation of non-linear measurement function $h_i(\cdot)$:

$$\mathbb{E}[h_i(x_s, \mathbf{x}_b) | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] = z_i. \quad (71)$$

Due different forms of non-linear measurement functions $h_i(\cdot)$, see equations (62)-(64), the equation (71) will produce different forms of conditional expectation $\mathbb{E}[x_s | \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] \equiv z_{f_i \rightarrow x_s}$:

$$a \mathbb{E}[x_s | \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b = 0 \quad (72a)$$

$$a \mathbb{E}[x_s^2 | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b \mathbb{E}[x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + c = 0 \quad (72b)$$

$$a \mathbb{E}[\sin^2 x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b \mathbb{E}[\sin x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + c = 0, \quad (72c)$$

where a , b and c are coefficients derived from non-linear measurement functions (see Appendix C for details).

Due to quadratic form of (72b) and (72c), we may obtain two possible values for the mean value $z_{f_i \rightarrow x_s}$. Thus in order to unambiguously define $z_{f_i \rightarrow x_s}$, we assume that certain a priori knowledge of state variables, denoted as $\tilde{\mathbf{x}} \equiv (\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{V}})$, is available (e.g., historical data). Given the prior data, we evaluate the mean value as:

$$z_{f_i \rightarrow x_s} = \begin{cases} z_{f_i \rightarrow x_s}^{(1)}, & \text{if } \Delta > 0 \text{ and } d_1 < d_2 \\ z_{f_i \rightarrow x_s}^{(2)}, & \text{if } \Delta > 0 \text{ and } d_1 > d_2 \\ \tilde{x}_s, & \text{if } \Delta < 0 \end{cases} \quad (73)$$

where Δ is the discriminant of the quadratic polynomial, and $d_1 = |z_{f_i \rightarrow x_s}^{(1)} - \tilde{x}_s|$, $d_2 = |z_{f_i \rightarrow x_s}^{(2)} - \tilde{x}_s|$, (see Appendix C for details).

The variance evaluation: The expression for the variance of the DC-BP $v_{f_i \rightarrow x_s}$ is equal (19b). Let us provide another interpretation of the variance $v_{f_i \rightarrow x_s}$. For this purpose, we observe the factor graph presented in Fig. 15. Consider the set of messages $\mu_{x_b \rightarrow f_i} = \mathcal{N}(z_{x_b \rightarrow f_i} | x_b, v_{x_b \rightarrow f_i})$ arriving to the factor node f_i from any variable node neighbour $x_b \in \mathcal{X}_i$. Informally, we note that this message carries a “belief” about itself that the variable node x_b sends to the factor node f_i , representing collective evidence the rest of the factor graph provides about the variable node x_b . Let us represent this belief by an equivalent factor node attached to each variable node. Thus for a set of variable nodes \mathcal{X}_i , we introduce a set of factor nodes $\mathcal{F}_{eq} = \{f_s, f_l, \dots, f_L\}$,

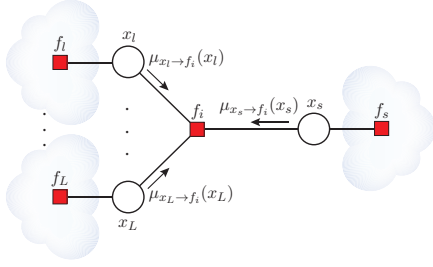


Fig. 15. Factor graph which provides interpretation of the variance $\sigma^2_{f_i \rightarrow x_s}$.

where for each $x_b \in \mathcal{X}_i$, the corresponding factor node $f_b \in \mathcal{F}_{eq}$ is singly-connected to x_b and by $\mathcal{N}(z_{x_b \rightarrow f_i} | x_b, v_{x_b \rightarrow f_i})$. Note that, from the perspective of SE, this factor node can be observed as a measurement defined by the value $z_{x_b \rightarrow f_i}$, variance $v_{x_b \rightarrow f_i}$, and measurement function $h_b(x_b) = x_b$.

Let us now solve the system illustrated in Fig. 15 using the WLS method. It is easy to show that the corresponding Jacobian matrix \mathbf{H} and weighted matrix \mathbf{W} have the following form:

$$\mathbf{H} = \begin{pmatrix} C_{x_s} & C_{x_l} & \dots & C_{x_L} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad (74)$$

$$\mathbf{W} = \text{diag}(1/v_i, 1/v_{x_s \rightarrow f_i}, 1/v_{x_l \rightarrow f_i}, \dots, 1/v_{x_L \rightarrow f_i}). \quad (75)$$

A variance-covariance matrix of WLS method is defined as:

$$\mathbb{V}(\mathbf{x}_i) = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} = \begin{pmatrix} \text{var}(x_s) & \text{cov}(x_s, x_l) & \dots & \text{cov}(x_s, x_L) \\ \text{cov}(x_l, x_s) & \text{var}(x_l) & \dots & \text{cov}(x_l, x_L) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(x_L, x_s) & \text{cov}(x_L, x_l) & \dots & \text{var}(x_L) \end{pmatrix}. \quad (76)$$

According to (76), and using (74) and (75), the variance $\text{var}(x_s)$ is:

$$\frac{1}{\text{var}(x_s)} = \frac{1}{v_{x_s \rightarrow f_i}} + \left[\frac{1}{C_{x_s}^2} \left(v_i + \sum_{x_b \in \mathcal{X}_i \setminus x_s} C_{x_b}^2 v_{x_b \rightarrow f_i} \right) \right]^{-1}. \quad (77)$$

Consider the second term on the right-hand side of (77). Recall that it represents the inverse of the variance $v_{f_i \rightarrow x_s}$ of the message from the factor node f_i to the variable node x_s , as defined by (19b). Therefore, we have demonstrated that by applying WLS on the factor graph in Fig. 15, one can obtain the expression for the variance of the message from the factor node f_i to the variable node x_s .

For the AC SE that deals with non-linear measurement functions, it is possible to define a linear approximation of the variance-covariance matrix at a given point \mathbf{x}_i using the Gauss-Newton method (11a):

$$\mathbb{V}(\mathbf{x}_i) = [\mathbf{J}(\mathbf{x}_i)^T \mathbf{W} \mathbf{J}(\mathbf{x}_i)]^{-1}. \quad (78)$$

⁶Note that the measurement function of the factor node f_i is given by (18), while for all other factor nodes $f_b \in \mathcal{F}_{eq}$, it is equal to $h_b(x_b) = x_b$.

It can be shown, using (78), that the variance $v_{f_i \rightarrow x_s}$ is governed by (19b) where the coefficients C_{x_p} , $x_p \in \mathcal{X}_i$ are defined by Jacobian elements (see Appendix A and C for details):

$$C_{x_p} = \left. \frac{\partial h_i(\cdot)}{\partial x_p} \right|_{\substack{x_s = z_{f_i \rightarrow x_s} \\ \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}}}. \quad (79)$$

Note that the coefficients above are evaluated at the point $\mathbf{x}_i = (x_s, \mathbf{x}_b)$, where the values in \mathbf{x}_i represent the mean-values of the corresponding messages.

To summarize, the message evaluation for the AC-BP is governed by (72) and (19b), where coefficients are obtained using (79).

Marginal inference: The marginal of the state variable x_s is governed by (21).

Iterative AC-BP algorithms Here, the *indirect factor nodes* $\mathcal{F}_{ind} \subset \mathcal{F}$ include measurements of power flows, power injections and currents. The *direct factor nodes* $\mathcal{F}_{dir} \subset \mathcal{F}$ include measurements of voltage magnitudes and angles.

The AC-BP algorithms are presented in Algorithm 3. Note that, the initialization step for the DC-BP and AC-BP is different. This is due to the fact that the variance of the message from a factor node to a variable node for the AC-BP depends not only on the mean values of incoming messages, but also on the mean value of the message whose variance is being calculated.

Algorithm 3 The AC-BP

```

1: procedure INITIALIZATION  $\tau = 0$ 
2:   for Each  $f_s \in \mathcal{F}_{loc}$  do
3:     send  $\mu_{f_s \rightarrow x_s}^{(0)}$  to incidence  $x_s \in \mathcal{X}$ 
4:   end for
5:   for Each  $x_s \in \mathcal{X}$  do
6:     send  $\mu_{x_s \rightarrow f_i}^{(0)} = \mu_{f_s \rightarrow x_s}^{(0)}$ , to incidence  $f_i \in \mathcal{F}_{id}$ 
7:   end for
8:   for Each  $f_i \in \mathcal{F}_{ind}$  do
9:     send  $\mu_{f_i \rightarrow x_s}^{(0)} = \mu_{x_s \rightarrow f_i}^{(0)}$  to incidence  $x_s \in \mathcal{X}$ 
10:  end for
11: end procedure
12: procedure ITERATION LOOP  $\tau = 1, 2, \dots$ 
13:   while stopping criterion is not met do
14:     for Each  $f_i \in \mathcal{F}_{ind}$  do
15:       Compute  $\mu_{f_i \rightarrow x_s}^{(\tau)}$  using (72)*, (19b)*
16:     end for
17:     for Each  $x_s \in \mathcal{X}$  do
18:       Compute  $\mu_{x_s \rightarrow f_i}^{(\tau)}$  using (14)
19:     end for
20:   end while
21: end procedure
22: procedure OUTPUT
23:   for Each  $x_s \in \mathcal{X}$  do
24:     Compute  $\hat{x}_s, v_{x_s}$  using (21)
25:   end for
26: end procedure

```

*Incoming messages are obtained in previous iteration $\tau - 1$

APPENDIX C

THE AC-BP ALGORITHM: DETAILED MESSAGE DERIVATION

Here we present an example of evaluation of the message from a factor node to a variable node for the AC-BP algorithm. We consider a simple model containing buses i and j , with the active power flow measurement $M_i \equiv M_{P_{ij}}$ at the branch (i, j) . The mean z_i , variance v_i and the measurement function $h_i(\theta_i, V_i, \theta_j, V_j)$ defined as (62a) is associated with the active power flow measurement M_i . The corresponding factor graph is shown in Fig. 16.

Further, all incoming messages from variable nodes to the factor node f_i have Gaussian form. Therefore, these messages, denoted as $\mu_{\theta_i \rightarrow f_i}(\theta_i)$, $\mu_{V_i \rightarrow f_i}(V_i)$, $\mu_{\theta_j \rightarrow f_i}(\theta_j)$ and $\mu_{V_j \rightarrow f_i}(V_j)$, are represented by their mean-variance pair $(z_{\theta_i \rightarrow f_i}, v_{\theta_i \rightarrow f_i})$, $(z_{V_i \rightarrow f_i}, v_{V_i \rightarrow f_i})$, $(z_{\theta_j \rightarrow f_i}, v_{\theta_j \rightarrow f_i})$ and $(z_{V_j \rightarrow f_i}, v_{V_j \rightarrow f_i})$, respectively (Fig. 16a - Fig. 16d).

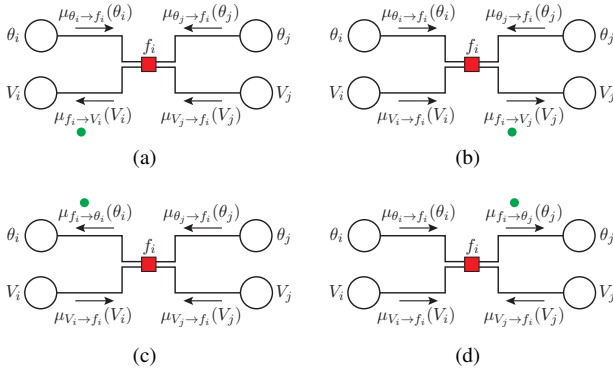


Fig. 16. Messages from factor node f_i to variable nodes: V_i (subfigure a), V_j (subfigure b), θ_i (subfigure c) and θ_j (subfigure d).

According to assumption (see Section Appendix B), the messages from the factor node f_i to variable nodes have Gaussian form: $\mu_{f_i \rightarrow V_i}(V_i)$, $\mu_{f_i \rightarrow V_j}(V_j)$, $\mu_{f_i \rightarrow \theta_i}(\theta_i)$ and $\mu_{f_i \rightarrow \theta_j}(\theta_j)$ with their mean-variance pair $(z_{f_i \rightarrow V_i}, v_{f_i \rightarrow V_i})$, $(z_{f_i \rightarrow V_j}, v_{f_i \rightarrow V_j})$, $(z_{f_i \rightarrow \theta_i}, v_{f_i \rightarrow \theta_i})$ and $(z_{f_i \rightarrow \theta_j}, v_{f_i \rightarrow \theta_j})$. In the following, we consider calculation of each of these messages.

- The message $\mu_{f_i \rightarrow V_i}$ (Fig. 16a): Let us first consider the mean $z_{f_i \rightarrow V_i}$. The equation (19a) for the active power flow measurement boils down to (72b):

$$a\mathbb{E}[V_i^2 | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b\mathbb{E}[V_i | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + c = 0,$$

where: $\mathbf{x}_b = (\theta_i, \theta_j, V_j)$ and $\mathbf{z}_{\mathbf{x}_b \rightarrow f_i} = (z_{\theta_i \rightarrow f_i}, z_{\theta_j \rightarrow f_i}, z_{V_j \rightarrow f_i})$, with coefficients:

$$\begin{aligned} a &= g_{ij} + g_{si} \\ b &= -z_{V_j \rightarrow f_i}(g_{ij} \cos z_{\theta_{ij} \rightarrow f_i} + b_{ij} \sin z_{\theta_{ij} \rightarrow f_i}) \\ c &= -z_i, \end{aligned}$$

where $z_{\theta_{ij} \rightarrow f_i}$ is determined as $z_{\theta_i \rightarrow f_i} - z_{\theta_j \rightarrow f_i}$. Due the fact that the conditional expected value $\mathbb{E}[V_i | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}]$ represents the mean $z_{f_i \rightarrow V_i}$, we can write:

$$a(z_{f_i \rightarrow V_i}^2 + v_{f_i \rightarrow V_i}) + bz_{f_i \rightarrow V_i} + c = 0.$$

The mean $z_{f_i \rightarrow V_i}$ follows from the quadratic equation, where we selected a solution using (73).

The variance $v_{f_i \rightarrow V_i}$ is determined using (19b) as:

$$\sigma_{f_i \rightarrow V_i}^2 = \frac{1}{C_{V_i}^2}(v_i + C_{\theta_i}^2 v_{\theta_i \rightarrow f_i} + C_{\theta_j}^2 v_{\theta_j \rightarrow f_i} + C_{V_j}^2 v_{V_j \rightarrow f_i}),$$

where coefficients are defined according to Jacobian elements of the measurement function $h_i(\cdot)$:

$$\begin{aligned} C_{\theta_i} &= \left. \frac{\partial h_i(V_i, \mathbf{x}_b)}{\partial \theta_i} \right|_{V_i=z_{f_i \rightarrow V_i}, \mathbf{x}_b=\mathbf{z}_{\mathbf{x}_b \rightarrow f_i}} & C_{\theta_j} &= \left. \frac{\partial h_i(V_i, \mathbf{x}_b)}{\partial \theta_j} \right|_{V_i=z_{f_i \rightarrow V_i}, \mathbf{x}_b=\mathbf{z}_{\mathbf{x}_b \rightarrow f_i}} \\ C_{V_i} &= \left. \frac{\partial h_i(V_i, \mathbf{x}_b)}{\partial V_i} \right|_{V_i=z_{f_i \rightarrow V_i}, \mathbf{x}_b=\mathbf{z}_{\mathbf{x}_b \rightarrow f_i}} & C_{V_j} &= \left. \frac{\partial h_i(V_i, \mathbf{x}_b)}{\partial V_j} \right|_{V_i=z_{f_i \rightarrow V_i}, \mathbf{x}_b=\mathbf{z}_{\mathbf{x}_b \rightarrow f_i}} \end{aligned}$$

- The message $\mu_{f_i \rightarrow V_j}$ (Fig. 16b): The mean $z_{f_i \rightarrow V_j}$ is defined according to (72a) as:

$$a\mathbb{E}[V_j | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b = 0,$$

where: $\mathbf{x}_b = (\theta_i, V_i, \theta_j)$ and $\mathbf{z}_{\mathbf{x}_b \rightarrow f_i} = (z_{\theta_i \rightarrow f_i}, z_{V_i \rightarrow f_i}, z_{\theta_j \rightarrow f_i})$, with coefficients:

$$\begin{aligned} a &= z_i - z_{V_i \rightarrow f_i}^2(g_{ij} + g_{si}) \\ b &= z_{V_i \rightarrow f_i}(g_{ij} \cos z_{\theta_{ij} \rightarrow f_i} + b_{ij} \sin z_{\theta_{ij} \rightarrow f_i}). \end{aligned}$$

Due the fact that the conditional expected value $\mathbb{E}[V_j | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}]$ represents the mean $z_{f_i \rightarrow V_j}$, we obtain:

$$az_{f_i \rightarrow V_j} + b = 0.$$

The variance $v_{f_i \rightarrow V_j}$ is determined using (19b) as:

$$v_{f_i \rightarrow V_j} = \frac{1}{C_{V_j}^2}(v_i + C_{\theta_i}^2 v_{\theta_i \rightarrow f_i} + C_{V_i}^2 v_{V_i \rightarrow f_i} + C_{\theta_j}^2 v_{\theta_j \rightarrow f_i}),$$

where coefficient are defined according to Jacobian elements of the measurement function $h_i(\cdot)$.

- The messages $\mu_{f_i \rightarrow \theta_i}$ and $\mu_{f_i \rightarrow \theta_j}$ (Fig. 16c and Fig. 16d): Means $z_{f_i \rightarrow \theta_i}$ and $z_{f_i \rightarrow \theta_j}$ are defined according to (72c):

$$a\mathbb{E}[\sin^2 x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + b\mathbb{E}[\sin x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] + c = 0,$$

where: $\mathbf{x}_b = (V_i, \theta_j, V_j)$ and $\mathbf{z}_{\mathbf{x}_b \rightarrow f_i} = (z_{V_i \rightarrow f_i}, z_{\theta_j \rightarrow f_i}, z_{V_j \rightarrow f_i})$ for the message $\mu_{f_i \rightarrow \theta_i}$, $\mathbf{x}_b = (\theta_i, V_i, V_j)$ and $\mathbf{z}_{\mathbf{x}_b \rightarrow f_i} = (z_{\theta_i \rightarrow f_i}, z_{V_i \rightarrow f_i}, z_{V_j \rightarrow f_i})$ for the message $\mu_{f_i \rightarrow \theta_j}$, and $x_s \in \{\theta_i, \theta_j\}$. Due the fact that the all variables and messages preserve Gaussian distribution, the conditional expectations of sine functions are equal to $\mathbb{E}[\sin^2 x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] = \sin^2 z_{f_i \rightarrow x_s}$ and $\mathbb{E}[\sin x_s | \mathbf{x}_b = \mathbf{z}_{\mathbf{x}_b \rightarrow f_i}] = \sin z_{f_i \rightarrow x_s}$, which allows us to compute the mean:

$$a \sin^2 z_{f_i \rightarrow x_s} + b \sin z_{f_i \rightarrow x_s} + c = 0.$$

To simplify expressions, we introduce coefficients $a = A^2 +$

B^2 , $b = -2BC$ and $c = -A^2 + C^2$:

$$A = g_{ij} \cos z_{\theta_j \rightarrow f_i} - b_{ij} \sin z_{\theta_j \rightarrow f_i}, \quad x_s \equiv \theta_i$$

$$A = g_{ij} \cos z_{\theta_i \rightarrow f_i} + b_{ij} \sin z_{\theta_i \rightarrow f_i}, \quad x_s \equiv \theta_j$$

$$B = g_{ij} \sin z_{\theta_j \rightarrow f_i} + b_{ij} \cos z_{\theta_j \rightarrow f_i}, \quad x_s \equiv \theta_i$$

$$B = g_{ij} \sin z_{\theta_i \rightarrow f_i} - b_{ij} \cos z_{\theta_i \rightarrow f_i}, \quad x_s \equiv \theta_j$$

$$C = \frac{z_{V_i \rightarrow f_i}^2 (g_{ij} + g_{si}) - z_i}{z_{V_i \rightarrow f_i} z_{V_j \rightarrow f_i}}, \quad x_s \in \{\theta_i, \theta_j\}$$

The variance $v_{f_i \rightarrow \theta_i}$ is determined using (19b) as:

$$v_{f_i \rightarrow \theta_i} = \frac{1}{C_{\theta_i}^2} (v_i + C_{V_i}^2 v_{V_i \rightarrow f_i} + C_{\theta_j}^2 v_{\theta_j \rightarrow f_i} + C_{V_j}^2 v_{V_j \rightarrow f_i}),$$

where coefficients are defined, as above, by calculating Jacobian elements of the measurement function $h_i(\cdot)$.

The variance $v_{f_i \rightarrow \theta_j}$ is determined according to (19b) as:

$$v_{f_i \rightarrow \theta_j} = \frac{1}{C_{\theta_j}^2} (v_i + C_{\theta_i}^2 v_{\theta_i \rightarrow f_i} + C_{V_i}^2 v_{V_i \rightarrow f_i} + C_{V_j}^2 v_{V_j \rightarrow f_i}),$$

where coefficient follow Jacobian elements of the measurement function $h_{P_i}(\cdot)$.

Using the same methodology, it is possible to define corresponding equations for means and variances for every type of measurement functions.

APPENDIX D

THE GN-BP ALGORITHM: TOY EXAMPLE

An illustrative example presented in Fig. 5a will be used to provide a step-by-step presentation of the proposed algorithm.

Input data for SE from measurement devices are Gaussian-type functions represented by means and variances: $\{z_{V_1}, z_{\theta_2}, z_{\theta_3}, z_{P_{12}}, z_{P_3}\}$ and $\{v_{V_1}, v_{\theta_2}, v_{\theta_3}, v_{P_{12}}, v_{P_3}\}$.

The corresponding factor graph is given in Fig. 14, where indirect factor nodes (red squares) are $f_{r_{P_{12}}}$ and $f_{r_{P_3}}$, while direct factor nodes (orange squares) are $f_{r_{V_1}}$, $f_{r_{\theta_2}}$ and $f_{r_{\theta_3}}$. The slack factor node $f_{r_{\theta_1}}$ (yellow square) corresponds to the slack or reference bus. The initialization (green square) and virtual (blue square) factor nodes are $f_{r_{V_2}}$ and $f_{r_{V_3}}$, respectively.

Each variable node in the initialization step of the algorithm has the corresponding local factor node. Local factor nodes only send, but do not receive, the messages to the incident variable nodes. Direct and virtual factor nodes always repeat the same message to their corresponding variable nodes through iterations.

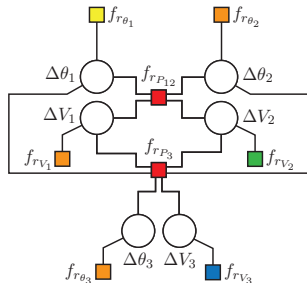


Fig. 17. Factor graph of the bus/branch model with given measurement configuration shown in Fig. 5a.

Algorithm Initialization

- 1) The AC SE in electric power systems assumes "flat start" or a priori given values of state variables:

$$\mathbf{x}^{(\nu=0)} = [\theta_1 \ \theta_2 \ \theta_3 \ V_1 \ V_2 \ V_3]^{(\nu=0)}.$$

- 2) The residual of the slack factor node is set to $r_{\theta_1} = 0$ with variance $v_{\theta_1} \rightarrow 0$.
- 3) The value of initialization factor nodes and virtual factor nodes are set to $r_{V_2} \rightarrow 0$ and $r_{V_3} \rightarrow 0$, with variances $v_{V_2} \rightarrow \infty$ and $v_{V_3} \rightarrow \infty$.

Iterate - Outer Loop: $\nu = 0, 1, 2, \dots; \tau = 0$

- 4) Each direct factor node computes residual, e.g.:

$$r_{V_1}^{(\nu)} = z_{V_1} - V_1^{(\nu)}.$$

- 5) Local factor nodes send messages represented by a triplet (residual, variance, state variable), to incident variable nodes, e.g.:

$$\mu_{f_{r_{\theta_1}} \rightarrow \Delta \theta_1}^{(\nu)} := (r_{\theta_1}, v_{\theta_1}, \theta_1^{(\nu)})$$

$$\mu_{f_{r_{V_1}} \rightarrow \Delta V_1}^{(\nu)} := (r_{V_1}^{(\nu)}, v_{V_1}, V_1^{(\nu)}).$$

- 6) Variable nodes forward the incoming messages received from local factor nodes along remaining edges, e.g.:

$$\begin{aligned} \mu_{\Delta \theta_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)} &= \mu_{f_{r_{\theta_1}} \rightarrow \Delta \theta_1}^{(\nu)} := (r_{\theta_1}^{(\nu)}, v_{\theta_1}, \theta_1^{(\nu)}) \\ &:= (r_{\Delta \theta_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)}, v_{\Delta \theta_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)}, \theta_1^{(\nu, \tau)}) \end{aligned}$$

$$\begin{aligned} \mu_{\Delta \theta_1 \rightarrow f_{r_{P_3}}}^{(\nu, \tau)} &= \mu_{f_{r_{\theta_1}} \rightarrow \Delta \theta_1}^{(\nu)} := (r_{\theta_1}^{(\nu)}, v_{\theta_1}, \theta_1^{(\nu)}) \\ &:= (r_{\Delta \theta_1 \rightarrow f_{r_{P_3}}}^{(\nu, \tau)}, v_{\Delta \theta_1 \rightarrow f_{r_{P_3}}}^{(\nu, \tau)}, \theta_1^{(\nu, \tau)}). \end{aligned}$$

- 7) Indirect factor nodes compute residuals, e.g.:

$$r_{P_{12}}^{(\nu)} = z_{P_{12}} - h_{P_{12}}(\theta_1^{(\nu)}, \theta_2^{(\nu)}, V_1^{(\nu)}, V_2^{(\nu)}).$$

- 8) Indirect factor nodes compute appropriate Jacobian elements associated with state variables, e.g.:

$$C_{P_{12}, \Delta \theta_1}^{(\nu)} = \frac{\partial h_{P_{12}}(\cdot)}{\partial \theta_1} = V_1^{(\nu)} V_2^{(\nu)} (g_{12} \sin \theta_{12}^{(\nu)} - b_{12} \cos \theta_{12}^{(\nu)})$$

$$C_{P_{12}, \Delta V_2}^{(\nu)} = \frac{\partial h_{P_{12}}(\cdot)}{\partial V_2} = -V_1^{(\nu)} (g_{12} \cos \theta_{12}^{(\nu)} + b_{12} \sin \theta_{12}^{(\nu)}).$$

Iterate - Inner Loop: $\tau = 1, 2, \dots, \eta(\nu)$

- 9) Indirect factor nodes send messages as pairs along incident edges according to (53), e.g.:

$$\mu_{f_{P_{12}} \rightarrow \Delta \theta_2}^{(\tau)} := (r_{f_{P_{12}} \rightarrow \Delta \theta_2}^{(\tau)}, v_{f_{P_{12}} \rightarrow \Delta \theta_2}^{(\tau)})$$

$$\begin{aligned} r_{f_{P_{12}} \rightarrow \Delta \theta_2}^{(\tau)} &= \frac{1}{C_{P_{12}, \Delta \theta_2}^{(\nu)}} \left[r_{P_{12}}^{(\nu)} - C_{P_{12}, \Delta \theta_1}^{(\nu)} \cdot r_{\Delta \theta_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} \right. \\ &\quad \left. - C_{P_{12}, \Delta V_1}^{(\nu)} \cdot r_{\Delta V_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} - C_{P_{12}, \Delta V_2}^{(\nu)} \cdot r_{\Delta V_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} \right] \end{aligned}$$

$$\begin{aligned} v_{f_{P_{12}} \rightarrow \Delta \theta_2}^{(\tau)} &= \frac{1}{(C_{P_{12}, \Delta \theta_2}^{(\nu)})^2} \left[v_{P_{12}} + (C_{P_{12}, \Delta \theta_1}^{(\nu)})^2 \cdot v_{\Delta \theta_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} \right. \\ &\quad \left. + (C_{P_{12}, \Delta V_1}^{(\nu)})^2 \cdot v_{\Delta V_1 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} + (C_{P_{12}, \Delta V_2}^{(\nu)})^2 \cdot v_{\Delta V_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau-1)} \right]. \end{aligned}$$

- 10) Variable nodes send messages as pairs along incident edges to indirect factor nodes according to (14), e.g.:

$$\mu_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)} := (r_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)}, v_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)})$$

$$\frac{1}{v_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)}} = \frac{1}{v_{\theta_2}} + \frac{1}{v_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}}$$

$$r_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)} = \left(\frac{r_{\theta_2}^{(\nu)}}{v_{\theta_2}} + \frac{r_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}}{v_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}} \right) v_{\Delta\theta_2 \rightarrow f_{r_{P_{12}}}}^{(\nu, \tau)}.$$

Iterate - Outer Loop: $\nu = 0, 1, 2, \dots; \tau = \eta(\nu)$

- 11) Variable nodes compute marginals according to (21), e.g.:

$$p(\Delta\theta_2) \propto \mathcal{N}(\Delta\hat{\theta}_2^{(\nu)} | \Delta\theta_2, \hat{v}_{\theta_2}^{(\nu)})$$

$$\frac{1}{\hat{v}_{\Delta\theta_2}^{(\nu)}} = \frac{1}{v_{\theta_2}} + \frac{1}{v_{f_{r_{P_{12}}} \rightarrow \Delta\theta_2}^{(\tau)}} + \frac{1}{v_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}}$$

$$\Delta\hat{\theta}_2^{(\nu)} = \left(\frac{r_{\theta_2}^{(\nu)}}{v_{\theta_2}} + \frac{r_{f_{r_{P_{12}}} \rightarrow \Delta\theta_2}^{(\tau)}}{v_{f_{r_{P_{12}}} \rightarrow \Delta\theta_2}^{(\tau)}} + \frac{r_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}}{v_{f_{r_{P_3}} \rightarrow \Delta\theta_2}^{(\tau)}} \right) \hat{v}_{\Delta\theta_2}^{(\nu)}.$$

- 12) Variable nodes update the state variables, e.g.:

$$\theta_2^{(\nu+1)} = \theta_2^{(\nu)} + \Delta\hat{\theta}_2^{(\nu)}.$$

- 13) Repeat steps 4-13 until convergence.

Note that, after step 8, initialization factor nodes are removed from the factor graph. Also in each iteration, virtual factor nodes repeat the same message as in the initial step and messages from a virtual factor node to a variable node should not be included in calculation of the marginals.

APPENDIX E

SELECTION OF RANDOMIZED DAMPING PARAMETERS

In this appendix, we analyze the GN-BP convergence performance in order to select appropriate values for randomized damping parameters p and α_1 introduced in equation (37). We generate 1000 random measurement configurations with the redundancy equals to 3. The set of measurement values are corrupted by additive noise of both the low noise level v_1 and the high noise level v_4 . The number of inner iterations is the same as in Section VI-B.

We record the number of non-converging simulations and evaluate the MAD value, both of which will depend on the convergence parameters p and α_1 . To recall, the probability p defines a fraction of a factor node to variable node messages from the current iteration that are combined with the corresponding messages from the previous iteration. The weighting coefficient α_1 defines the ratio that determines how messages from the current and the previous iteration are combined. For example, $p = 0.2$ specifies that 20% of the messages from the current iteration will be combined with their values in the previous iteration, while 80% of messages are keeping the values calculated in the current iteration. Furthermore, if $\alpha_1 = 0.1$, then for the 20% of messages, the new value is obtained as a linear combination of the values calculated in the current and the previous iteration with

coefficients 0.1 and 0.9, respectively. In the following, we numerically investigate the randomized damping parameter pairs that lead to acceptable trade-off between the number of non-converging simulations and the rate of convergence.

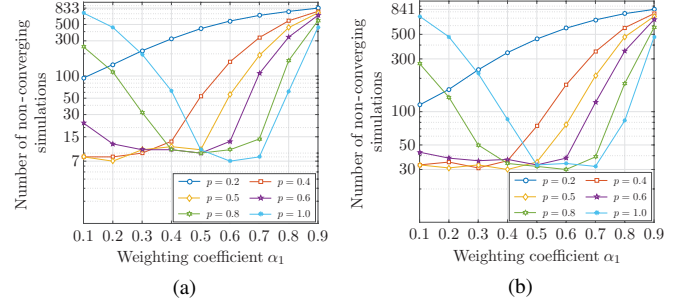


Fig. 18. The number of non-converging simulations (out of 1000 simulations in total) for the low noise level v_1 (subfigure a) and the high noise level v_4 (subfigure b).

Fig. 18 shows the number of non-converging simulations for low noise level v_1 and high noise level v_4 as a function of p and α_1 . In general, for the selection of p and α_1 for which only a small fraction of messages are combined with their values in the previous iteration, and that is the case for p close to zero or α_1 close to one, we observe a large number of non-converging simulations. This clearly demonstrates the necessity of using (37) to “slow down” the BP progress, thus increasing the algorithm stability and providing improved convergence.

For a fixed value of α_1 , we investigate the convergence rate of GN-BP for different values of p . We expect that, for any selected α_1 , the GN-BP will converge faster for smaller values of p , as lower p leads to a reduced “slow down” effect. For example, Fig. 19 shows median MAD values for $\alpha_1 = 0.5$ over the range of p values (where the median is calculated over convergent simulation outcomes only). Figure clearly shows that lower p leads to faster convergence. However, one needs to be careful with selection of p in order to avoid the combinations of p and α_1 that lead to large number of non-converging outcomes.

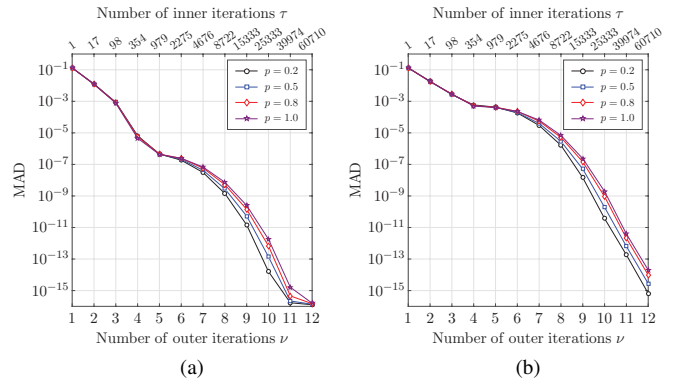


Fig. 19. The median MAD values for $\alpha_1 = 0.5$ over the range of p values for the low noise level v_1 (subfigure a) and the high noise level v_4 (subfigure b).

To select convergence parameters, we observed the median MAD values and the number of non-converging simulations. Table I lists the p and α_1 pairs that demonstrate good performance of both metrics of interest. From the table, we can select e.g., $p = 0.4$ and $\alpha_1 = 0.3$, to achieve good balance between the number of non-converging simulations and the convergence rate.

TABLE I
CONVERGENCE PERFORMANCES

Convergence parameters		Median MAD values per iteration (i.e. 7,8,9)						Number of non-converging simulations	
		Variance v_1			Variance v_2				
		7 [10 ⁻⁸]	8 [10 ⁻⁹]	9 [10 ⁻¹¹]	7 [10 ⁻⁵]	8 [10 ⁻⁶]	9 [10 ⁻⁸]		
p	α_1							v_1	v_2
0.4	0.1	5.547	5.123	9.942	5.402	4.776	10.411	8	33
0.4	0.2	5.218	4.206	9.427	4.848	3.851	8.775	8	35
0.4	0.3	4.522	3.271	6.697	4.663	3.134	6.593	9	31
0.5	0.2	5.929	5.340	13.782	5.634	5.177	14.042	7	31
0.5	0.3	5.351	4.543	10.061	5.160	4.847	9.514	10	33
0.5	0.4	4.928	3.754	7.316	4.826	3.605	8.060	11	30
0.5	0.5	4.264	2.835	5.309	3.971	2.887	5.096	10	35
0.6	0.5	4.762	4.034	7.354	4.528	3.179	6.902	9	33
0.8	0.5	5.850	5.179	14.215	5.503	5.140	14.454	9	32
0.8	0.6	4.887	3.823	8.595	4.644	3.948	8.800	10	30
1.0	0.6	5.590	5.231	13.882	5.456	4.963	13.086	7	34
1.0	0.7	4.593	3.638	7.097	4.500	3.613	7.084	8	32

There are two main causes of non-converging behavior. The main cause is that the number of inner iterations in each outer iteration follows the exponential rule ($\tau = \nu^q$). In general, the GN-BP performs very well with such an exponential increase of inner iterations, however, it suffers from insufficient number of inner iterations in the first few outer iterations. Our experiments show that increasing the number of inner iteration only in the first few outer iterations can dramatically reduce the number of non-converging simulations (see Section VI-D). The second cause is a combination of “weakly” connected variable nodes and measurements, especially if those nodes represent the buses incident to transformers. In that case, to prevent the non-converging behavior, it is sufficient to connect virtual factor nodes to such variable nodes. Therefore, with some adjustments, one can eliminate the non-converging cases and make the GN-BP algorithm fully reliable.

APPENDIX F

SCALABILITY AND COMPLEXITY OF GN-BP ALGORITHM

Scalability of the GN-BP algorithm: To demonstrate scalability of the GN-BP algorithm, we provide MAD values for the IEEE 30 and IEEE 118 bus test case. As before, we generate 1000 random measurement configurations with the redundancy equal to 3. The set of measurement values are corrupted by additive noise of both the low noise level v_1 and the high noise level v_4 . The convergence parameters are set to $p = 0.4$ and $\alpha_1 = 0.3$ and we use the exponential inner iteration scheme. From Fig. 20 and Fig. 21, we note that the GN-BP converges, however, the rate of convergence slightly decreases as the size of the system increases. As described in the previous subsection, one can use different inner iteration schemes to fine-tune the rate of convergence.

Complexity of the GN-BP algorithm: Let us first consider the complexity of the centralized SE based on the

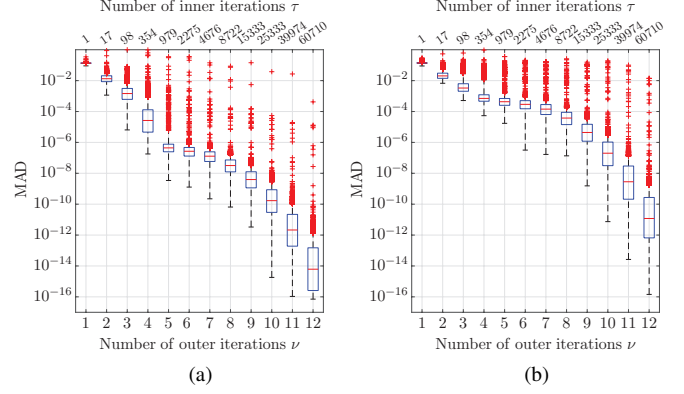


Fig. 20. The MAD values of the GN-BP for the IEEE 30 bus test case with the exponential inner iteration scheme ($\eta(\nu) = \nu^e$) for the low noise level v_1 (subfigure a) and high noise level v_4 (subfigure b).

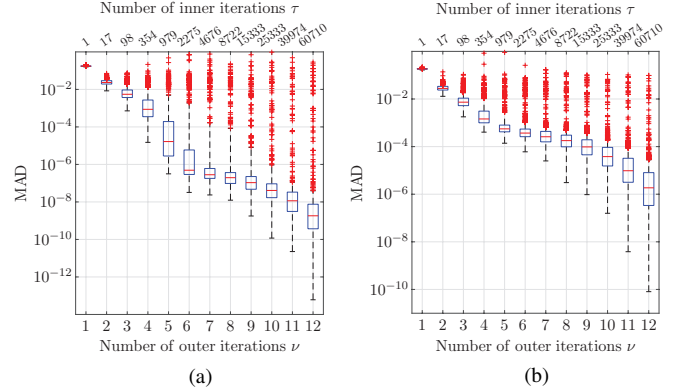


Fig. 21. The MAD values of the GN-BP for the IEEE 118 bus test case with the exponential inner iteration scheme ($\eta(\nu) = \nu^e$) for the low noise level v_1 (subfigure a) and high noise level v_4 (subfigure b).

Gauss-Newton method. For n state variables and k measurements, each iteration of the centralized SE involves a constant number of matrix multiplications and a matrix inversion whose complexity is of the order of $\mathcal{O}(kn^2)$ and $\mathcal{O}(n^3)$, respectively. Due to the fact that k is proportional to n for observable systems, and that the centralized SE takes constant number of iterations to converge, the overall complexity of the centralized SE scales as $\mathcal{O}(n^3)$. However, this can be reduced by employing matrix inversion techniques that exploit the sparsity of involved matrices. The resulting sparsity-aware centralized SE methods operate with complexity that scales as $\mathcal{O}(n^2)$ [41], [42].

The complexity of BP depends on the sparsity of the underlying factor graph, as the computational effort per iteration is proportional to the number of edges in the factor graph. For each of the k measurements, the degree (the number of incident edges) of the corresponding factor node is limited by a (typically small) constant. Indeed, for any type of measurements, the corresponding measurement function depends only on a few state variables corresponding to the buses in the local neighbourhood of the bus/branch where the measurement is taken. Thus as n and k grow large, the number of edges in the factor graph scales as

$\mathcal{O}(n)$, which corresponds to the computation complexity of BP *per iteration*. The scaling of the number of BP iterations ρ as n grows large is a challenging problem. Based on discussion in [43] for full matrices, the number of iterations is likely to scale with condition number of the underlying matrix, which for well-conditioned matrices may scale as low as $\mathcal{O}(1)$. However, we leave the more detailed analysis on the scaling of the number of GN-BP iterations for our future work.

To summarize, BP approach builds upon the factor graph structure that directly exploits the underlying system sparsity, thus achieving minimal complexity of $\mathcal{O}(n)$ per iteration, while the scaling of the number of iterations needs further study. In contrast to the optimized centralized methods whose complexity scales as $\mathcal{O}(n^2)$, the BP method can be flexibly distributed by arbitrarily segmenting the underlying factor graph into disjoint areas. In the extreme case of the fully-distributed BP algorithm (that we focus on in this paper), each factor graph node operates locally and independently. Thus, the SE problem is distributed across $\mathcal{O}(n)$ nodes, and if implemented to run in parallel, can be $\mathcal{O}(n)$ times faster than the centralized solution. In addition, for fully-distributed BP, none of the nodes need to store the system-level matrices (whose storage-size typically scales as $\mathcal{O}(n^2)$), and storing only constant-size set of local parameters is sufficient.

- [2] F. F. Wu, K. Moslehi, and A. Bose, "Power system control centers: Past, present, and future," *Proc. of the IEEE*, vol. 93, pp. 1890–1908, Nov 2005.
- [3] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.
- [4] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, part ii: Approximate model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 125–130, Jan 1970.
- [5] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Processing Magazine*, vol. 29, no. 5, 2012.
- [6] G. N. Korres, "A distributed multiarea state estimation," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 73–84, Feb 2011.
- [7] W. Jiang, V. Vittal, and G. T. Heydt, "Distributed state estimation using phasor measurement units," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1580–1589, Nov 2008.
- [8] L. Zhao and A. Abur, "Multi area state estimation using synchronized phasor measurements," *IEEE Transactions on Power Systems*, vol. 20, no. 2, 2005.
- [9] A. Minot, Y. Lu, and N. Li, "A distributed gauss-newton method for power system state estimation," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, July 2016, pp. 1–1.
- [10] D. Marelli, B. Ninness, and M. Fu, "Distributed weighted least-squares estimation for power networks," *IFAC-PapersOnLine*, vol. 48, no. 28, 2015.
- [11] X. Tai, Z. Lin, M. Fu, and Y. Sun, "A new distributed state estimation technique for power networks," in *2013 American Control Conference*, June 2013.
- [12] R. Ebrahimi and R. Baldick, "State estimation distributed processing [for power systems]," *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1240–1246, 2000.
- [13] A. J. Conejo, S. de la Torre, and M. Canas, "An optimization approach to multiarea state estimation," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 213–221, 2007.
- [14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [15] H. Zhu and G. Giannakis, "Power system nonlinear state estimation using distributed semidefinite programming," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1039–1050, 12 2014.
- [16] J. Matamoros, A. Tsitsimelis, M. Gregori, and C. Antn-Haro, "Multiarea state estimation with legacy and synchronized measurements," in *2016 IEEE International Conference on Communications*, May 2016, pp. 1–6.

REFERENCES

- [1] A. Abur and A. Expósito, *Power System State Estimation: Theory and Implementation*, ser. Power Engineering. Taylor & Francis, 2004.
- [17] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, 2013.
- [18] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1184–1194, 2013.
- [19] L. Xie, D. H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1154–1169, Sept 2012.
- [20] A. Gómez-Expósito, A. de la Villa Jaén, C. Gómez-Quiles, P. Rousseaux, and T. Van Cutsem, "A taxonomy of multi-area state estimation methods," *Electric Power Systems Research*, vol. 81, no. 4, pp. 1060–1069, 2011.
- [21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1988.
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [23] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 736–744, 2001.
- [24] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [25] Y. Hu, A. Kuh, T. Yang, and A. Kavcic, "A belief propagation based power distribution system state estimator," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 36–46, 2011.
- [26] Y. Weng, R. Negi, and M. Ilic, "Graphical model for state estimation in electric power systems," in *Smart Grid Communications, 2013 IEEE International Conference on*, Oct 2013, pp. 103–108.
- [27] T. Sui, D. E. Marelli, and M. Fu, "Convergence analysis of gaussian belief propagation for distributed state estimation," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015.
- [28] M. Cosovic and D. Vukobratovic, "Fast real-time DC state estimation in electric power systems using belief propagation," 2017. [Online]. Available: <http://arxiv.org/abs/1705.01376>
- [29] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.
- [30] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.
- [31] A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach*, ser. Kluwer international series in engineering and computer science. Springer US, 1999.
- [32] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- [33] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*, ser. A Wiley-Interscience publication. Wiley, 1996.
- [34] G. Elidan, I. McGraw, and D. Koller, "Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing," *ArXiv e-prints*, Jun. 2012.
- [35] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in gaussian graphical models of arbitrary topology," in *Advances in neural information processing systems*, 2000, pp. 673–679.
- [36] B. L. Ng, J. Evans, and S. Hanly, "Distributed downlink beamforming in cellular networks," in *2007 IEEE International Symposium on Information Theory*, June 2007, pp. 6–10.
- [37] C. Fan, X. Yuan, and Y. J. Zhang, "Scalable uplink signal detection in c-rans via randomized gaussian message passing," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.
- [38] M. Pretti, "A message-passing algorithm with damping," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 11, 2005.
- [39] P. Hansen, V. Pereyra, and G. Scherer, "Least squares data fitting with applications," Johns Hopkins University Press, 2012, ch. 9, p. 166.
- [40] M. Cosovic and D. Vukobratovic, "Distributed gauss-newton method for ac state estimation: A belief propagation approach," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2016, pp. 643–649.
- [41] M. K. Enns, W. F. Tinney, and F. L. Alvarado, "Sparse matrix inverse factors [power systems]," *IEEE Transactions on Power Systems*, vol. 5, no. 2, 1990.
- [42] F. L. Alvarado, "Computational complexity in power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. 95, no. 4, July 1976.
- [43] D. Bickson, "Gaussian Belief Propagation: Theory and Application," *ArXiv e-prints*, Nov. 2008.