Abstract—The state estimation algorithm estimates the values of the state variables based on the measurement model described as the system of equations. Prior to applying the state estimation algorithm, the existence and uniqueness of the solution of the underlying system of equations is determined through the observability analysis. If a unique solution does not exist, the observability analysis defines observable islands and further defines an additional set of equations (measurements) needed to determine a unique solution. For the first time, we utilise factor graphs and Gaussian belief propagation algorithm to define a novel observability analysis approach. The observable islands and placement of measurements to restore observability are identified by following the evolution of variances across the iterations of the Gaussian belief propagation algorithm over the factor graph. Due to sparsity of the underlying power network, the resulting method has the linear computational complexity (assuming a constant number of iterations) making it particularly suitable for solving large-scale systems. The method can be flexibly matched to distributed computational resources, allowing for determination of observable islands and observability restoration in a distributed fashion. Finally, we discuss performances of the proposed observability analysis using power systems whose size ranges between 1354 and 70000 buses.

Index Terms—Observability Analysis, Observable Islands, Observability Restoration, Belief Propagation, Factor Graphs, State Estimation, Large-Scale Systems

I. INTRODUCTION

Motivation: Deregulation of the electric power industry has led to the development of distributed control centres that operate over segments of a large interconnected power system. Such evolution of power systems calls for the distributed monitoring and control operations [1]. As a part of the energy management system, power system monitoring relies on the distributed, scalable and computationally efficient state estimation (SE) routines that include the network topology processor, observability analysis, SE algorithm, and bad data processing.

Typically, the observability analysis enacts before the SE algorithm with bad data processing. According to the available measurements, observability analysis determines whether the network is observable or unobservable. Within an unobservable network, the observability analysis identifies all observable islands that can be solved independently, and obtains a minimal set of additional measurements providing the observability of the entire system, which ensures a unique estimate of the power system state [2]. Over the past decades, comprehensive studies have been published that predominantly rely on distributed SE algorithms with bad data processing. In contrast, computationally efficient and distributed observability analysis received significantly less attention from the research community.

Literature Review: The identification of observable islands and observability restoration is traditionally carried out by applying numerical or topological methods. Numerical methods for the identification of observable islands and the observability restoration are mostly based on the matrix factorisation. For the identification of observable islands, there are several methods depending on the implementation approach: i) iterative methods that use the Jacobian [2], the gain [3], [4], and the Gram matrix [5]; ii) non-iterative methods that use the gain matrix with the inverse of its triangular matrix [6], and the Jacobian matrix based on the branch variable formulation [7]. Similar categorisation can be made for the numerical methods applied to the observability restoration: i) iterative methods that use the gain [3], [6], the Gram [5], and reduced network matrix [8]; ii) non-iterative methods that use the Jacobian matrix [2], the gain matrix with the inverse of its triangular matrix [9], and the Gram matrix of the reduced Jacobian [10], [11]. These approaches depend on the accurate computation of zero pivots. The major drawback of these methods is the sensitivity to the zero pivot threshold value, where the incorrect value may deteriorate observability analysis. Also, non-iterative methods usually require definition and the calculation of additional matrices as compared to the iterative methods. Methods that use the Gram matrices, even for the reduced system, are prone to the risk of losing sparsity.

Unlike numerical methods, topological methods are less analysed with the most of the research efforts focused on the identification of the observable islands. These methods do not use any floating point calculations in the analysis. The decision is strictly based on the logical operations, and therefore requires information about the network connectivity, measurement types and their locations [12, Sec. 4.7]. Typically, topological methods first process all flow measurements to define initial islands, moving to processing injection measurements in an iterative fashion, followed by the identification of observable islands [8], [13]–[15]. In the paper [16], authors propose a method based on the symbolic Jacobian matrix which considers only non-redundant measurements. Starting from a random bus, the proposed method utilise search technique to reveal observable islands through repetitive process.
Although topological methods can be adopted in a distributed framework, little work has been done in this area. However, numerical methods are also appealing for the observability analysis in the distributed framework, with many novel methods developed over the years. Several studies \cite{17,18} proposed the distributed observability analysis based on the numerical methods that require coordination among areas, where observability restoration is accomplished by a non-iterative approach. Authors in \cite{19} analysed the distributed observability in terms of a consensus algorithm, providing conditions which enable solving the system of equations.

**Belief Propagation Approach and Contributions:** In this paper, we present a novel observability analysis approach based on a probabilistic graphical model, complementing our previous work on Gaussian belief propagation (GBP) based SE \cite{20}. The proposed method can be understood as both numerical and topological, with several advantages over the current state-of-the-art approaches. Our contributions are summarised as follows:

- Compared to numerical methods, the proposed method does not utilise direct factorisation or inversion of matrices, avoiding inaccurate computation of zero pivots and incorrect choice of a zero threshold value.
- Due to the sparsity of the underlying factor graph, the proposed method has linear computational complexity per GBP iteration, making it particularly suitable for large-scale systems.
- In the multi-area scenario, the proposed method can be implemented over the non-overlapping multi-area scenario without the central coordinator. The GBP algorithm ensures measurement data privacy in the distributed architecture, where areas exchange only “beliefs” about specific state variables. The GBP algorithm within local areas can be implemented using synchronous scheduling, where all messages in a given iteration are updated using the output of the previous iteration as an input. Message updates between local areas are not necessarily performed in full synchronism with the scheduling applied within local areas, as this would incur excessive communication delay to each iteration. Thus, the GBP framework allows the algorithm in which asynchronous information exchange is possible \cite{21}.
- The method is straightforward to distribute and parallelize. In particular, even if implemented in the framework of centralised SE, it can be flexibly mapped to distributed computation resources (e.g., parallel processing on graphical-processing units).

The novelty of our work follows from the fact that, for the first time, we adopt the GBP algorithm to carry out the observability analysis. Moreover, we emphasise a particularly simplifying feature of the proposed method, where within the GBP framework, it is sufficient to track the evolution of the variance values only. In particular, the GBP algorithm convergence depends on the convergence of the means, while variances always converge \cite{22}. To leverage this property, we propose observability analysis based only on variances resulting in an algorithm that *always converges*. The resulting algorithm relies only on the graph structure and connections between nodes, and in contrast to numerical methods, it is immune to numerical issues.

**II. Observability Analysis**

This section provides a brief review of key concepts used for observability analysis in the power systems. Observability analysis is commonly performed on the linear decoupled measurement model \cite{23 Ch. 7}, which we adopt in this paper. In summary, the observability analysis considers the following linearised measurement model:

\[
Hx = z, \quad (1)
\]

where \(z \in \mathbb{R}^m\) is the vector of known values, \(x \in \mathbb{R}^n\) is the solution, while \(H \in \mathbb{R}^{m \times n}\) is a known matrix. The system is observable if and only if \(\text{rank}(H) = n - 1\),\( (2)\)

where the state variable corresponding to the slack bus is already included in \(1\). Ensuring the observability by condition \(2\) does not imply achieving a good estimation of the state variables. This is because numerical ill-conditioning may deteriorate the SE algorithm, but in the most cases, if the system is observable, it can be efficiently estimated \cite{24}. Note that the requirement posed by the condition \(2\) implicitly assumes using the least-squares approach in solving \(1\). In theory, requirement \(2\) is not sufficient for the system \(1\) to have a solution (the existence of the solution depends on the vector \(z\)). However, \(2\) guarantees invertibility of the symmetric matrix \(H^T H \in \mathbb{R}^{n \times n}\), that is at the heart of the least-squares approach, therefore making \(1\) solvable.

When the given set of measurements is not sufficient (i.e., \(\text{rank}(H) < n - 1\)), the observability analysis must identify all the possible observable islands that can be independently solved \(2\), where an observable island is defined as follows:

**Definition 1.** An observable island is a part of the power system for which the flows across all branches of the observable island can be calculated from the set of available measurements, independent of the values adopted for angular reference \cite{25 Sec. 7.1.1}.

In other words, the observability analysis divides the system into the set of islands \(S = \{s_1, \ldots, s_k\}\), where \(k\) is the number of islands. Formally, the set of state variables \(X = \{x_1, \ldots, x_n\}\) is divided into disjoint subsets \(X_i \subset X, i = 1, \ldots, k\). Each island encloses \(n_i = |X_i|\) state variables, where the number of *linearly independent* equations \(l_i\) corresponding to the island \(s_i\), is equal to \(l_i = n_i - 1\). Once the islands are determined, the observability analysis merges these islands in a way to protect previously-determined observable states from being altered by the new set of equations defined by the additional measurements. In general, this can be achieved by ensuring that the set of new measurements is a non-redundant set \cite{23 Sec. 7.3.2}, i.e., the set of equations must be linearly independent with regard to the global system. The aim of the observability restoration is to find this non-redundant set. The observability restoration finds exactly \(k - 1\) linearly independent equations, where the equation that defines the slack bus is known in advance.
III. BELIEF PROPAGATION BASED OBSERVABILITY ANALYSIS

We consider a factor graph approach for modelling the system of equations [25]. This approach starts with the construction of the factor graph with the set of variable nodes representing unknown variables, and the set of factor nodes corresponding to the set of equations. Note that a factor node connects to a variable node if and only if the unknown variable associated with the variable node is an argument of the corresponding equation associated with the factor node. Alongside the factor nodes that represent system equations (measurements), we introduce an additional set of virtual factor nodes defined as follows:

Definition 2. The virtual factor node is a singly-connected factor node attached to the variable node, with the variance set either to zero or to infinity.

A. Determination of Observable Islands

In our setup, from the set of equations, we retain a subset that corresponds to the set of the linearly independent equations defined by the power flow and the power injection measurement functions. We start with the construction of the factor graph with the set of variable nodes \( \mathcal{X} \) representing state variables, and the set of factor nodes \( \mathcal{F} \) corresponding to the linearly independent set of equations. Utilising the property that the GBP is robust to ill-conditioned scenarios caused by significant differences between measurement variances [20], [26], we augment the set of factor nodes \( \mathcal{F} \) with the set of virtual factor nodes \( \mathcal{F}_x \). Note that virtual factor nodes are equivalent to the pseudo-measurements of the voltage angles at the buses. Furthermore, if its variance tends to the infinity, the factor node has no influence on the rest of the graph.

Example 1 (Constructing a factor graph). In this toy example, using 6-bus model presented in Fig. 1 we demonstrate the conversion from a bus/branch model with a given measurement configuration to the corresponding factor graph. The linearised measurement Jacobian matrix can be formed as:

\[
H = \begin{bmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
  1 & -1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & -1 & 0 \\
  -1 & -1 & 3 & -1 & 0 & 0 \\
  0 & 0 & 0 & -1 & 2 & -1 \\
\end{bmatrix}
\]

The corresponding factor graph is given in Fig. 2 where the set of variable nodes \( \mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6\} \) is defined according to the vector of state variables \( x \). The set of factor nodes \( \mathcal{F} \) is defined by corresponding measurements, where in our example, active power flows \( M_{P_{12}}, M_{P_{45}}, \) active power injections \( M_{P_3}, M_{P_6} \) measurements are mapped into factor nodes \( \mathcal{F} = \{f_7, f_8, f_9, f_{10}\} \) (red boxes), respectively. The factor graph reflects the structure of the matrix \( H \), where each row of the matrix corresponds to one factor node, while columns define variable nodes. Finally, we augment the factor graph with the set of virtual factor nodes \( \mathcal{F}_x = \{f_1, f_2, f_3, f_4, f_5, f_6\} \) (orange and green boxes), where the factor node \( f_1 \) is attached to the slack bus.

Gaussian Belief-Propagation is a well-known algorithm for probabilistic inference on factor graphs [27]. It is used to efficiently calculate marginal distributions of random variables in a large-scale, sparse, linear systems of random variables corrupted by Gaussian noise. In this paper, we adapt GBP in order to solve the problem of observability analysis in power systems. We present GBP-based island detection in this subsection and extend it to GBP-based observability restoration in the next subsection.

The GBP-based island detection algorithm tracks the evolution of marginal variances of the state variables, establishing observability criteria in the relation to whether or not these variances converge to a unique fixed point. Let us assume, for the time being, that the system is a fully observable, and the Jacobian matrix is defined according to power flows and injections measurement functions. According to Definition 1 the values of the state variables corresponding to the buses are uniquely determined if at least one of the state variables in this set is declared known (i.e., introducing voltage angle pseudo-measurement). As a result, the Jacobian matrix has a full rank [23, Sec. 7.1.3], while the marginal GBP variances converge to a unique fixed point. Indeed, according to Lemma 1 of [28] and Theorem 2 of [29], variances always converge to a unique fixed point if the fixed point exists. However, when the system is unobservable, using GBP-based approach it is still possible to identify subsets of state variables that converge to a unique point [29]. We explore this feature for the identification of observable islands.

We formalise the proposed setup used to detect observable islands in Algorithm 1. We consider a factor graph with the set of variable nodes \( \mathcal{X} \), the set of factor nodes \( \mathcal{F} \), and the set of virtual factor nodes \( \mathcal{F}_x \) constructed as follows. The set \( \mathcal{F}_x \) contains \( n \) factor nodes, out of which a single one
has a variance set to zero and is attached to an arbitrary (randomly selected) variable node (we call this factor node a “probe”), and the remaining \(n-1\) factor nodes have variances set to infinity, and are attached to the remaining \(n-1\) variable nodes. After the initialisation (lines 1-9), the identification of observable islands procedure (lines 10-40) starts by selecting a random variable node from the set \(X^{(v)}\) to which a probe factor node is attached, where \(v\) denotes the number of passes through the procedure (line 12). The message-passing starts by setting initial GBP variances from variable nodes \(x_i \in X^{(v)}\) and factor nodes \(f_j \in F^{(v)}\) to finite positive values (line 13). The message-passing framework with marginal inference (lines 14-28) represents the main algorithm routine which determines the observable islands (lines 29-34), where the set \(F_i \setminus f_j\) represents factor nodes incident to the variable node \(x_i\), excluding the factor node \(f_j\), while the set \(X_j \setminus x_i\) represents variable nodes incident to the factor node \(f_j\), excluding the variable node \(x_i\). Upon the calculation of the marginal variances for all the variable nodes in \(X^{(v)}\), a node \(x_i \in X^{(v)}\) is proclaimed observable, i.e., \(x_i \in X_0^{(v)}\), if and only if its marginal variance \(\hat{v}_{x_i}\) is finite. Note that the resulting observable island depends on the position of the probe factor node. After the discovery of observable variable nodes, we consider a subgraph of the factor graph that contains the set of unobservable variable nodes \(X^{(v+1)} = X^{(v)} \setminus X_0^{(v)}\) and the set of all factor nodes \(F^{(v+1)}\) and \(F_{X}^{(v+1)}\) connected exclusively to variable nodes in \(X^{(v+1)}\) (lines 35-37). The procedure now repeats from the line 11. The next observable island is revealed by introducing a new randomly selected probe, and setting its virtual factor node variance to zero. This causes a change in marginal variances, in a way that all unobservable nodes within the next revealed observable island switch over into finite marginal variance values. The procedure is repeated until all observable islands are revealed.

**Example 2 (Identification of Observable Islands).** We demonstrate the GBP-based island detection algorithm, using the factor graph given in Example 1. We start by selecting the probe factor node \(f_1\) and setting its variance to zero, while variances of the rest of virtual factor nodes are set to infinity:

\[
\begin{align*}
\nu f_1 \to x_1 &\to 0 \\
\nu f_i \to x_i &\to \infty, \quad i = 2, \ldots, 6.
\end{align*}
\]

The algorithm starts with initial finite (e.g., equal to one) variance values:

\[
\nu x_i \to f_j = 1, \quad x_i \in X, \quad f_j \in F.
\]

The GBP iteration \(\tau = 1\) starts with computing variances from factor nodes \(F\) to variable nodes \(X\), using incoming variances from variable nodes \(X\) to factor nodes \(F\). For example, computation of the variance from the factor node \(f_0\) to the variable node \(x_1\) is computed as follows:

\[
\nu f_0 \to x_2 = \nu x_0 \to f_0 + \nu x_0 \to f_0 + \nu x_1 \to f_0 = 3.
\]

Next, the algorithm proceeds with computing variances from variable nodes \(X\) to factor nodes \(F\), using incoming variances from factor nodes \(F\) to variable nodes \(X\), for example:

\[
\frac{1}{\nu f_1 \to x_1} = \frac{1}{\nu f_1 \to x_1} + \frac{1}{\nu f_1 \to x_1} = v_{f_1 \to f_7} \to 0.
\]

When the algorithm converges, all variances from factor nodes \(F\) to variable nodes \(X\) are equal to infinity, except the variance from the factor node \(f_7\) to the variable node \(x_2\) that tends to zero \(\nu f_7 \to x_2 \to 0\). The marginal variance of variable nodes \(X\) can be obtained using variances from factor nodes \(F\) to variable nodes \(X\):

\[
\frac{1}{\nu x_1} = \frac{1}{\nu f_1 \to x_1} + \frac{1}{\nu f_0 \to x_1} \to 0.
\]

**Algorithm 1 Identification of Observable Islands**

1: procedure **Initialisation**
2: determine linearly independent set of equations
3: for each \(x_i \in X\) do
4: assign virtual factor node \(f_i \in F\) with infinite variance
5: end for
6: initialise counter \(\nu = 1\)
7: initialise the set of variable nodes \(X^{(v)} = X\)
8: initialise the set of factor nodes \(F^{(v)} = F\), \(F^{(v)} = F\)
9: end procedure
10: procedure **Identification of Observable Islands**
11: while \(X^{(v)} \neq \emptyset\) do
12: set a random factor node \(f_p \in F^{(v)}\) to zero variance (probe)
13: set all variances from \(x_i \in X^{(v)}\) to \(f_p \in F^{(v)}\) on finite values
14: procedure **Message-Passing Framework**
15: while stopping criterion is not met do
16: for each \(f_j \in F^{(v)}\) do
17: \(\nu v_{f_j \to x_i} = \nu x_6 \to f_j\)
18: end for
19: for each \(x_i \in X^{(v)}\) do
20: \(v_{f_0 \to x_i}^{-1} = \sum_{f_j \in F} (\nu v_{f_j \to x_i})^{-1}\)
21: end for
22: end while
23: end procedure
24: procedure **Marginal Variances**
25: for each \(x_i \in X^{(v)}\) do
26: \(\nu v_{x_i}^{-1} = \sum_{f_j \in F} (\nu v_{f_j \to x_i})^{-1}\)
27: end for
28: end procedure
29: initialise the set of observable variable nodes \(X_0^{(v)} = \emptyset\)
30: for each \(x_i \in X^{(v)}\) do
31: if \(\nu_{x_i}\) has finite variance then
32: the variable node is observable \(X_0^{(v)} \cup x_i\)
33: end if
34: end for
35: the set \(X_0^{(v)}\) form observable island \(s\)
36: form \(X^{(v+1)} = X^{(v)} \setminus X_0^{(v)}\)
37: form \(F^{(v+1)}\), \(F_{X}^{(v+1)}\) incident only to variable nodes \(X^{(v+1)}\)
38: increase counter \(\nu := \nu + 1\)
39: end while
40: end procedure

from factor nodes \(F \cup F_i\) to variable nodes \(X\), for example:

\[
\frac{1}{\nu x_1} = \frac{1}{\nu f_1 \to x_1} + \frac{1}{\nu f_0 \to x_1} + \frac{1}{\nu f_1 \to x_1} \to 0.
\]
Next, we introduce a new set of equations containing all the candidate measurements that are able to merge islands: 

\[ R_{1} = \begin{bmatrix} H_{ia} & I_{d\times d} \end{bmatrix} \begin{bmatrix} x \end{bmatrix} + \begin{bmatrix} z_{u} \\ z_{c} \\ z_{p} \end{bmatrix} = \begin{bmatrix} r \end{bmatrix}, \]  

where \( H \in \mathbb{R}^{h \times n} \), \( x \in \mathbb{R}^{n} \), \( z_{u} \in \mathbb{R}^{h} \), and \( h = n - k \). In addition, we introduce a new set of equations containing all the candidate measurements that are able to merge islands: 

\[ \begin{bmatrix} H_{ia} \\ H_{c} \\ H_{p} \end{bmatrix} x = \begin{bmatrix} z_{c} \\ z_{p} \end{bmatrix}, \]  

where \( H_{c} \in \mathbb{R}^{q \times n} \) and \( z_{c} \in \mathbb{R}^{q} \) represent measurements which were rejected during observable island identification, while \( H_{p} \in \mathbb{R}^{h \times n} \) and \( z_{p} \in \mathbb{R}^{b} \) describe pseudo-measurements. The system of equations described with \( H_{c} \) contains only linearly independent equations. Furthermore, we note that additional \( w = k - q - 1 \) equations among equations defined by pseudo-measurements are needed to achieve observability, as the equation that defines the slack bus is known in advance.

Let us introduce the vectors \( \alpha \), \( \beta \) and \( \gamma \) representing an uncorrelated noise with the independent and identically distributed entries, where each element follows a zero-mean Gaussian distribution with associated variances \( \alpha_{i} \), \( \beta_{i} \) and \( \gamma_{i} \). Hence, systems (10) and (11) can be merged together as:

\[ \begin{bmatrix} H_{ia} \\ H_{c} \\ H_{p} \end{bmatrix} x + \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} z_{u} \\ z_{c} \\ z_{p} \end{bmatrix}. \]  

Next, we apply a simple trick using the values of variances assigned to different measurements. Namely, we first set \( \alpha_{i} \to 0 \), \( i = 1, \ldots, h \), \( \beta_{i} \to 0 \), \( i = 1, \ldots, q \), and \( \gamma_{i} \to \infty \), \( i = 1, \ldots, b \). Then, we select exactly \( w \) linearly independent equations (with regards to the global system) from the set of pseudo-measurements. Without loss of generality, we choose the first \( w \) equations and we set \( \gamma_{i} \to 0 \), \( i = 1, \ldots, w \), i.e., we change the variance of \( w \) linearly independent measurements from infinity to zero. After additional \( w \) equations are “denoised”, the solution of (12) will be determined only by the equations associated to the variances that are set to zero. Note that the corresponding residuals are also equal to zero. In other words, each of the noiseless equations is a critical equation [12, Sec. 5.2]. However, the described scenario cannot be solved using the traditional weighted least-squares method, because we defined the ill-conditioned scenario caused by significant differences between variances. In contrast, the GBP approach is robust to ill-conditioned scenarios, and above-mentioned scenario can be solved [26].

Introducing the additional vector of measurement residuals \( r \in \mathbb{R}^{d} \), \( d = h + q + b \), we can expand (12) as follows:

\[ \begin{bmatrix} H_{ia} \\ H_{c} \\ H_{p} \end{bmatrix} x + \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} z_{u} \\ z_{c} \\ z_{p} \end{bmatrix}, \]  

where \( I \) is the identity matrix, \( 0_{d\times n} \) is the all-zero matrix, \( 0_{d\times d} \) is the vector of zeros, and the vector \( \delta \) represents an uncorrelated noise defined in the same way as vectors \( \alpha \), \( \beta \) and \( \gamma \). By applying the weighted least-squares method to (13) and inspecting the variance-covariance matrix, it is easy to show that the variances of the state variables \( r \) depend only on the variances related to the noise vector \( \delta \) in the case when the system (12) contains the set of linearly independent equations, and therefore (12) does not influence the residual values. In other words, if an equation described by a factor node \( f_{i} \) sends infinity variance to the corresponding residual described by a variable node \( r_{i} \), the equation is linearly independent.

For example, consider a part of the factor graph shown in Fig. 4 and focus on a single measurement that defines the factor node \( f_{i} \). Note that the message from a factor node to a variable node represents a “belief” about the variable node, and each message exchanged in GBP is completely represented...
using only two values: mean and variance. Messages from variable nodes \( \{x_1, \ldots, x_L\} \) to the factor node \( f_i \) carry a collective evidence of the rest of the factor graph, and according to these messages, the GBP algorithm computes the message \( \mu_{f_i \rightarrow r_i}(r_i) \) from the factor node \( f_i \) to the variable node \( r_i \). Next, we set the mean value of the message \( \mu_{f_i \rightarrow r_i}(r_i) \) from the singly-connected factor node \( f_a \) to the variable node \( r_i \) to zero, and its variance to a finite positive value. The marginal distribution of the variable node \( r_i \) obtained as the product of the incoming messages \( \mu_{f_i \rightarrow r_i}(r_i) \) and \( \mu_{f_a \rightarrow r_i}(r_i) \). The mean of the marginal will remain zero only if the variance of the message \( \mu_{f_i \rightarrow r_i}(r_i) \) tends to infinity. This is the case, when the corresponding equation is linearly independent and does not affect the residual value.

To summarize, the GBP-based algorithm decomposes the contribution of each factor node (equation) to a group of variable nodes (state variables and a measurement residual). This decomposition allows insight into the structure of the measurement residual, where the impact of each measurement can be observed. Hence, if one factor node \( f_i \) sends finite variance value to the residual \( r_i \), this means that the equation has an impact on the measurement residual value, i.e., \( r_i \neq 0 \), while infinite variance corresponds to the case when the equation does not have an impact to the measurement residual, i.e., \( r_i = 0 \).

In order to reduce numerical complexity, it is possible to avoid running GBP inference across the entire factor graph defined by \( \mathcal{F} \). Instead, we can transform the original factor graph into a considerably more compact version which is equivalent from the restoration viewpoint. The variable nodes from sets \( \mathcal{X}_i, i = 1, \ldots, k \) are connected by factor nodes \( \mathcal{F}_c \) and \( \mathcal{F}_p \) defined according to matrices \( \mathbf{H}_c \) and \( \mathbf{H}_p \), respectively. Also, each island \( s_i \) has \( l_i = n_i - 1 \) linearly independent equations, where revealing any of the state variables from the set \( \mathcal{X}_i \) reveals the entire observable island \( s_i \). This allows the construction of the factor graph, where all state variables \( \mathcal{X}_i \) enclosed by the island \( s_i \) are merged into a single variable node i.e., each island becomes the variable node. Thus, we observe the factor graph with a set of variable nodes \( \mathcal{S} = \{s_1, \ldots, s_k\} \) and the set of factor nodes \( \mathcal{F}_c \cup \mathcal{F}_p \). Same as before, we introduce the set of virtual factor nodes \( \mathcal{F}_v \) attached to the variable nodes from the set \( \mathcal{S} \). This strategy reveals an exciting finding: the factor graph defined by the variable nodes \( \mathcal{S} \), and factor nodes \( \mathcal{F}_c \) and \( \mathcal{F}_p \) equates to the matrix \( \mathbf{W}_{bc} \) given in \[10\]. However, in contrast to \[10\] where the Gram matrix is needed, we are solving the observability restoration problem directly over the matrix \( \mathbf{W}_{bc} \) using the GBP algorithm.

We formalise the proposed method for the observability restoration in Algorithm 2. The initialisation step (lines 1-13) creates the factor graph, where each factor node \( f_i \) from the set \( \mathcal{F}_c \) has the same variance equal to \( v_i \), i.e., variance \( v_i \) corresponds to the equation associated with the factor node \( f_i \). Each variable node \( s_i \in \mathcal{S} \) has the virtual factor node with variances set to infinity (lines 8-10). The algorithm continues observability restoration procedure (lines 14-44) until \( w = k - q - 1 \) linearly independent equations are found (line 11). The observability restoration procedure (lines 14-44) starts by selecting a factor node \( f_v \) from the set \( \mathcal{F}_p \) followed by changing its variance from the infinity to \( v_i \). In other words, we select one of the equations defined by pseudo-measurements that can merge islands (line 16). Then, we select one of the variable nodes incident to the factor node \( f_v \) and change the variance of its virtual factor node to zero. This virtual factor node has a role of a slack variable node that exists in the system (lines 17-18). The message-passing procedure (lines 20-29) commences after the initial GBP variances from variable nodes \( s_i \in \mathcal{S} \) to factor nodes \( f_j \in \mathcal{F}_c \cup \mathcal{F}_p \) are set to finite positive values (line 19), where the set \( \mathcal{F}_c \setminus f_j \) represents factor nodes incident to the variable node \( s_i \), excluding the

---

**Algorithm 2: Observability Restoration**

1: procedure **Initialisation**
2: for each \( f_i \in \mathcal{F}_c \) do
3: assign finite variance equal to \( v_i \)
4: end for
5: for each \( f_v \in \mathcal{F}_p \) do
6: assign infinite variance
7: end for
8: for each \( s_i \in \mathcal{S} \) do
9: assign virtual factor node \( f_v \in \mathcal{F}_v \) with infinite variance
10: end for
11: initialise \( w = k - q - 1 \)
12: initialise counter \( \nu = 1 \)
13: end procedure
14: procedure **Observability Restoration**
15: while \( w > 0 \) do
16: select \( \nu \)-th factor node \( f_v \in \mathcal{F}_p \), and assign variance \( v_i \)
17: find \( s_v \) as one of the variable nodes incident to the factor node \( f_v \)
18: assign zero variance to the virtual factor node attached to \( s_v \)
19: set all variances from \( s_v \in \mathcal{S} \) to \( f_v \in \mathcal{F}_c \cup \mathcal{F}_p \) on finite values
20: procedure **Message-Passing Framework**
21: while stopping criterion is not met do
22: for each \( f_j \in \mathcal{F}_c \cup \mathcal{F}_p \) do
23: \( \nu^{(r)}_{f_j \rightarrow s_v} = v_{f_j} + \sum_{s_k \in \mathcal{S}\setminus s_v} \nu^{(r-1)}_{s_k \rightarrow f_j} \)
24: end for
25: for each \( s_i \in \mathcal{S} \) do
26: \( \nu^{(r)}_{s_i \rightarrow f_j} = \sum_{f_n \in \mathcal{F}_c \setminus f_j} \nu^{(r-1)}_{f_n \rightarrow s_i} \)
27: end for
28: end while
29: end procedure
30: procedure **Linear Independence**
31: assign residual variable node \( r_v \) to the factor node \( f_v \)
32: compute \( v_{f_v \rightarrow r_v} \)
33: if \( v_{f_v \rightarrow r_v} \rightarrow \infty \) then
34: factor node \( f_v \) represents independent equation
35: \( w := w - 1 \)
36: elseif
37: factor node \( f_v \) represents dependent equation
38: assign infinite variance to \( f_v \)
39: end if
40: change variance of the virtual factor node of \( s_v \) to infinity
41: increase counter \( \nu := \nu + 1 \)
42: end procedure
43: end while
44: end procedure
factor node $f_j$, while the set $S_j \setminus s_i$ represents variable nodes incident to the factor node $f_j$, excluding the variable node $s_i$. The algorithm proceeds with the linear independence procedure (lines 30-42). The equation, described by the factor node $f_v$, is linearly dependent or independent, based on the variance value from factor node $f_v$ to the residual variable node $v_r$. The linear independence procedure marks an equation as linearly independent if the variance of factor node $f_v$ is finite positive value, while the variances of factor nodes $f_{12}$, $f_{13}$, $f_{14}$, and $f_{15}$ are set to infinity. We begin by setting the variances from the virtual factor nodes as follows:

$$v_{f_{i} \rightarrow s_{i}} \rightarrow 0, \quad v_{f_{i} \rightarrow s_{i}} \rightarrow \infty, \quad i = 2, 3.$$  

The GBP iterations are performed in a similar way as in Example 2 except that here we use variance $v_{i}$ to compute variances from factor nodes $f_{i} \in F_v \cup F_p$ to variable nodes $s_i \in S$. When the algorithm converges, we include the residual variable node $r_{12}$ incident to the factor node $f_{12}$ and compute variance $v_{f_{12} \rightarrow r_{12}}$ using variances into the factor node $f_{12}$:

$$v_{f_{12} \rightarrow r_{12}}^{(\tau_{\text{max}})} = v_{s_{1} \rightarrow f_{12}}^{(\tau_{\text{max}})} + v_{s_{3} \rightarrow f_{12}}^{(\tau_{\text{max}})} + v_{f_{12} \rightarrow r_{12}} \rightarrow \infty. \quad (15)$$

Thus, the factor node $f_{12}$ represents a linearly independent equation, and the system is fully observable using measurements $M_{P_1}$ and $M_{P_3}$ described by factor nodes $f_{11}$ and $f_{12}$, respectively.

IV. NUMERICAL RESULTS

In this section, we assess our proposed method against the numerical and topological state-of-the-art methods available in the literature. In our evaluation, we explore a large set of power systems with different bus configurations. We evaluate the performance of the proposed method using power systems with 1354, 2000, 10000, 25000 and 70000 buses [31], [32].

For each power system, we generate 100000 pseudo-measurement configurations per each number of islands, to control for the resulting number of observable islands, where we aim at producing approximately the same number of measurement configurations per each number of islands ranging between 2 and 33.

In all scenarios, we use box plots where we normalise execution time of each of the different state-of-the-art methods $t_{sa}$ by the execution time of the GBP-based approach $t_{bp}$. Hence, the case $t_{sa}/t_{bp} > 1$ corresponds to the case where the GBP-based method shows better performance (i.e., lower execution time), while $t_{sa}/t_{bp} < 1$ corresponds to the case where the GBP-based method shows worse performance (i.e., higher execution time) in comparison with the state-of-the-art method. The methods have been implemented using Julia programming language, while the source code of the GBP-based observability analysis can be found as the part of the JuliaGrid package.

A. Determination of Observable Islands

To evaluate the GBP-based island detection algorithm, we compare the execution time of the GBP-based method $t_{bp}$ against the numerical method based on the nodal variable formulation $t_{nm}$ [12] Sec. 4.6.2 and the topological method based on the multi-stage procedure $t_{nm}$ [13].

Fig. 7 shows distributions of the normalised execution times $t_{nm}/t_{bp}$ and $t_{nm}/t_{bp}$ for power systems with 1354 and 2000 buses. We split results into two groups corresponding to the particular range of the number of islands. For all experiments, the topological and GBP-based methods achieve significantly lower execution times compared to the numerical method. Both topological and the GBP-based methods result

2https://github.com/mcosovic/JuliaGrid.jl
in comparable execution times across all experiments with the topological method achieving marginally lower execution times. Note that median values of the execution time for the topological and GBP-based methods are between 0.78 ms and 1.37 ms, and between 1.38 ms and 2.96 ms, respectively.

Using a funnel-based approach, we omit numerical method from further experiments due their large execution times on power systems with a smaller number of buses, making it unpractical for larger systems. Fig. 8 shows box plots of the normalised execution time $t_{\text{bp}}/t_{\text{nm}}$ for the power systems with 10000, 25000 and 70000 buses with regard to the number of islands. The GBP-based method outperforms topological method for larger power systems. Also, for the fixed power system size (i.e., the number of buses), increasing the number of islands results in lower execution times for the GBP-based method compared to the topological method. Note that the execution time medians of the GBP-based method for 70000-bus power system are 92.97 ms and 123.43 ms for 2-17 and 18-33 group of islands, respectively.

### C. Observability Restoration

To evaluate the GBP-based observability restoration algorithm, we compare the execution time of the GBP-based method $t_{\text{bp}}$ against the numerical method based on the Gram matrix associated with a Jacobian matrix corresponding to a reduced network and measurement set $t_{\text{nm}}$.

![Fig. 7. The normalised execution time of the numerical and topological methods to the GBP-based method for power systems with 1354 and 2000 buses with regard to the number of islands.](image)

![Fig. 8. The normalised execution time of the topological method versus the GBP-based method for the power systems with 10000, 25000 and 70000 buses with regard to the number of islands.](image)

![Fig. 9. The normalised execution time of the numerical method versus the GBP-based method for 70000-bus power system with regard to the number of islands.](image)

The methods have been tested on a 64 bit Windows 10 with an Intel CoreTM i7-8850H 2.60 GHz CPU, and 32 GB of RAM. It is important to emphasise that the same GBP iteration scheme (i.e., stopping criterion threshold, the maximum number of iterations) was used throughout all scenarios. However, for the specific system, by tuning these parameters, execution times can be further improved.

---

3The methods have been tested on a 64 bit Windows 10 with an Intel CoreTM i7-8850H 2.60 GHz CPU, and 32 GB of RAM. It is important to emphasise that the same GBP iteration scheme (i.e., stopping criterion threshold, the maximum number of iterations) was used throughout all scenarios. However, for the specific system, by tuning these parameters, execution times can be further improved.
directly related to the sparsity of the underlying factor graph \cite{20, 28}.

For the second point, the scaling of the number of the GBP iterations as \( n \) grows large is a more challenging problem related to the convergence rate of the GBP algorithm. Based on a discussion in \cite{33} for full matrices, we know that the number of iterations is likely to scale with the condition number of the underlying matrix, which, for well-conditioned matrices may scale as low as \( O(1) \) (constant). In our method, we only track the evolution of the variance values, because variances converge much faster than means. Tracking only variances results in a significant simplification. In our framework, we expect the number of the GBP iterations to scale as a constant.

Another advantage of using the GBP algorithm is the possibility of distributing GBP method across disjoint areas by arbitrarily segmenting the underlying factor graph. In the extreme case of the fully-distributed GBP algorithm, each factor graph node operates locally and independently. In this scenario, the problem is distributed across \( O(n) \) nodes, and, if implemented to run in parallel, it can be \( O(n) \) times faster than the centralised solution.

V. CONCLUSIONS

We presented a novel computationally efficient GBP-based observability analysis suitable for large-scale data processing in electric power systems that can be easily implemented as part of a fully distributed architecture. In essence, proposed algorithms use only addition operations over the factor graph. Both, determination of observable islands and observability restoration algorithm, are implemented by tracking evolution of the variances, which results in the algorithm that always converges. While algorithms use floating point calculations, most of the time they operate as topological methods. Finally, for large-scale power systems, GBP-based method significantly outperforms numerical and topological state-of-the-art methods, reducing execution time of island detection procedure while being more robust in a search for correct non-redundant set of equations.

VI. ACKNOWLEDGMENT

This paper has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement number 856967.

REFERENCES

[1] L. Xie, D. H. Choi, S. Kar, and H. V. Poor, “Fully distributed state estimation for wide-area monitoring systems,” IEEE Trans. Smart Grid, vol. 3, no. 3, pp. 1154–1169, Sept. 2012.
[2] B. Gou, “Jacobian matrix-based observability analysis for state estimation,” IEEE Trans. Power Syst., vol. 21, no. 1, pp. 348–356, Feb. 2006.
[3] A. Monticelli and F. F. Wu, “Network observability: Identification of observable islands and measurement placement,” IEEE Trans. Power App. Syst., vol. PAS-104, no. 5, pp. 1035–1041, 1985.
[4] F. F. Wu and A. Monticelli, “Network observability: Theory,” IEEE Trans. Power App. Syst., vol. PAS-104, no. 5, pp. 1042–1048, 1985.
[5] M. C. de Almeida, E. N. Asada, and A. V. Garcia, “Power system observability analysis based on Gram matrix and minimum norm solution,” IEEE Trans. Power Syst., vol. 23, no. 4, pp. 1611–1618, 2008.
[6] B. Gou and A. Abur, “A direct numerical method for observability analysis," IEEE Trans. Power Syst., vol. 15, no. 2, pp. 625–630, 2000.
[7] A. G. Expósito and A. Abur, “Generalized observability analysis and measurement classification,” IEEE Trans. Power Syst., vol. 13, no. 3, pp. 1090–1095, 1998.
[8] G. C. Contaxis and G. N. Korres, “A reduced model for power system observability: Analysis and restoration,” IEEE Trans. Power Syst., vol. 3, no. 4, pp. 1411–1417, 1988.
[9] B. Gou and A. Abur, “An improved measurement placement algorithm for network observability,” IEEE Trans. Power Syst., vol. 16, no. 4, pp. 819–824, 2001.
[10] G. N. Korres, “A Gram matrix-based method for observability restoration,” IEEE Trans. Power Syst., vol. 26, no. 4, pp. 2569–2571, 2011.
[11] G. N. Korres and N. M. Manousakis, “Observability analysis and restoration for state estimation using SCADA and PMU data,” in Proc. PESGM, 2012, pp. 1–8.
[12] A. Abur and A. Expósito, Power System State Estimation: Theory and Implementation, ser. Power Engineering. Taylor & Francis, 2004.
[13] H. Horisberger, “Observability analysis for power systems with measurement deficiencies,” IFAC Proceedings Volumes, vol. 18, no. 7, pp. 51–58, 1985.
[14] G. R. Krumpholz, K. A. Clements, and P. W. Davis, “Power system observability: A practical algorithm using network topology,” IEEE Trans. Power App. Syst., vol. PAS-99, no. 4, pp. 1534–1542, 1980.
[15] K. A. Clements, G. R. Krumpholz, and P. W. Davis, “State estimator measurement system reliability evaluation – an efficient algorithm based on topological observability theory,” IEEE Trans. Power App. Syst., vol. PAS-101, no. 4, pp. 997–1004, 1982.
[16] I. W. Slutsker and J. M. Scudder, “Network observability analysis through measurement Jacobian matrix reduction,” IEEE Trans. Power Syst., vol. 2, no. 2, pp. 331–336, 1987.
[17] G. N. Korres, “A distributed multiarea state estimation,” IEEE Trans. Power Syst., vol. 26, no. 1, pp. 73–84, Feb. 2011.
[18] G. N. Korres and G. C. Contaxis, “Application of a reduced model to a distributed state estimator,” in Proc. PESGM, vol. 2, Jan. 2000, pp. 999–1004 vol.2.
[19] S. Kar and J. M. F. Moura, “Consensus + innovations distributed inference over networks: Cooperation and sensing in networked systems,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 99–109, May 2013.
[20] M. Cospovic and D. Vukobratovic, “Distributed Gauss–Newton method for state estimation using belief propagation,” IEEE Trans. Power Syst., vol. 34, no. 1, pp. 648–658, Jan. 2019.
[21] M. Cospovic and D. Vukobratovic, “Large-scale multi-area state estimation from phasor measurement units utilizing factor graphs,” in Proc. IEEE EUROCON, 2019, pp. 1–8.
[22] B. L. Ng, J. Evans, and S. Hanly, “Distributed downlink beamforming in cellular networks,” in Proc. IEEE ISIT, 2007, pp. 6–10.
[23] A. Monticelli, State Estimation in Electric Power Systems: A Generalized Approach, ser. Kluwer international series in engineering and computer science. Springer US, 1999.
[24] G. N. Korres, “Observability analysis based on echelon form of a reduced dimensional Jacobian matrix,” IEEE Trans. Power Syst., vol. 26, no. 4, pp. 2572–2573, 2011.
[25] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 498–519, 2001.
[26] M. Cospovic and D. Vukobratovic, “Fast real-time DC state estimation in electric power systems using belief propagation,” in Proc. IEEE SmartGridComm, 2017, pp. 207–212.
[27] 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.
[28] C. Fan, X. Yuan, and Y. J. Zhang, “Scalable uplink signal detection in CRANs via randomized Gaussian message passing,” IEEE Trans. Wireless Commun., vol. 16, no. 8, pp. 5187–5200, Aug. 2017.
[29] R. D. Yates, “A framework for uplink power control in cellular radio systems,” IEEE J. Sel. Areas Commun., vol. 13, pp. 1341–1347, 1995.
[30] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient erasure correcting codes,” IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 569–584, Feb. 2001.
[31] S. Flisounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, “Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions,” IEEE Trans. Power Syst., vol. 28, no. 4, pp. 4900–4917, Nov. 2013.
[32] A. B. Birchfield, T. Xu, K. M. Hegner, K. S. Shetye, and T. J. Overbye, “Grid structural characteristics as validation criteria for synthetic networks,” IEEE Trans. Power Syst., vol. 32, pp. 3258–3265, July 2017.
[33] D. Bickson, “Gaussian belief propagation: Theory and application,” arXiv preprint arXiv:0811.2518, 2008.