Identification of Edge Disconnections in Networks Based on Graph Filter Outputs

Shlomit Shaked and Tirza Routtenberg, Senior Member, IEEE.

Abstract—Graphs are fundamental mathematical structures used in various fields to model statistical and physical relationships between data, signals, and processes. In some applications, such as data processing in graphs that represent physical networks, the initial network topology is known. However, disconnections of edges in the network change the topology and may affect the signals and processes over the network. In this paper, we consider the problem of edge disconnection identification in networks by using concepts from graph signal processing (GSP).

We assume that the graph signals measured over the vertices of the network can be represented as white noise that has been filtered on the graph topology by a smooth graph filter. We develop the likelihood ratio test (LRT) to detect a specific set of edge disconnections. Then, we provide the maximum likelihood (ML) decision rule for identifying general scenarios of edge disconnections in the network. It is shown that the sufficient statistics of the LRT and ML decision rule are the graph-frequency energy levels in the graph spectral domain. However, the ML decision rule leads to a high-complexity exhaustive search over the edges in the network and is practically infeasible. Thus, we propose a low-complexity greedy method that identifies a single disconnected edge at each iteration. Moreover, by using the smoothness of the considered graph filter, we suggest a local implementation of the decision rule, which is based solely on the measurements at neighboring vertices. Simulation results demonstrate that the proposed methods outperform existing detection and identification methods on a synthetic dataset and for line outage identification in power systems from the IEEE 118-bus test case.

Index Terms—Network topology, identification of edge disconnections, Graph Signal Processing (GSP), likelihood ratio test (LRT), smooth graph filters

I. INTRODUCTION

Graph signals arise in a wide range of applications, ranging from physical networks, such as power, transportation, and communication systems, to data-driven networks, where the graph is used to represent relations among the observed data, such as in social networks. The emerging field of graph signal processing (GSP) extends classical signal processing methodologies, such as filtering and sampling, to the graph domain [1]–[5]. Based on this GSP theory, different graph signals are represented as white noise that has been filtered on the graph topology by a graph filter. Most GSP methods are based on the assumption that the underlying network topology is known. However, even if the original topology of the network is known, small changes in this topology may occur that significantly degrade the performance of GSP tasks [6]. In particular, edge disconnections, i.e., links between the graph vertices that have been dropped, is a common problem, especially in physical networks. For example, in electrical networks, the problem of identifying line outages that happen due to environmental factors, damages, aging, and malicious attacks, is a significant problem [7]–[10]. Additional examples are identifying traffic congestion and blockages in transportation networks [11], and detecting links that drop in wireless communication networks because of random blocking or fading [12]. Accordingly, a fundamental question is how to use measured graph signals to identify edge disconnections, where the original underlying network structure is known.

Numerous works in the literature have focused on (full) graph-topology learning. A popular approach in the context of graphical models is the graphical Lasso, which is based on relating between the inverse of the sample covariance matrix, i.e., the precision matrix, and the connectivity of the graph [13]–[15]. Recent GSP-based network inference frameworks are based on recovering a graph shift operator (GSO), which encodes direct relationships between the signal elements from observed indirect relationships generated by graph diffusion processes [16]–[19]. Other works consider inference of dynamic networks and nonlinear models [20]–[22]. However, the full topology identification methods are inefficient (in the sense of the amount of data required) and suboptimal (in terms of performance) for detecting a few specific topology disconnections, where the nominal topology is known.

Detection of graph topology changes with different prior knowledge on the topology has been investigated. The problem of detecting changes in a sequence of graphs by spectral methods has been considered in [23], [24]. Anomalous subgraph detection deals with detecting small, anomalous subgraphs embedded into background networks with known properties (such as the modularity matrix eigenvectors [25], [26]). However, the methods in [23]–[26] are based on data in the form of random graphs and do not consider data measured on the vertices of a graph. Recent works discuss detecting changes in graph topology from graph signals [27]–[29]. In [27], [28], matched subspace detectors have been developed to decide which one of two given graphs match better with a given dataset, under the assumptions of bandlimitedness in the graph Fourier domain but without making any assumption on the nature of the change itself. Using information regarding the nature of the change in the topology, as considered in this paper, is expected to improve the detection performance. Several edge exclusion tests have been proposed for general graphical models by using the partial spectral coherence [30], Matsuda test statistic [31], and a generalized likelihood ratio test (GLRT) [32], [33] for detecting edge exclusion. In [34], a new algorithm is proposed...
for learning the Laplacian matrix of the graph in Gaussian Markov random field (GMRF) models, where the connectivity is assumed to be known. The simulations in [34] show that this algorithm can be used to identify connectivity mismatches.

In this paper, we consider the problem of detecting disconnections in the topology based on graph signals that are represented as the output of smooth graph filters [35–37]. We formulate the hypothesis testing problem for deciding the status (connected/disconnected) of a specific subset of edges. We develop the likelihood ratio test (LRT) for this hypothesis testing and show that the LRT is a function of the graph-frequency energy levels in the graph spectral domain. In addition, for the noiseless GMRF model, the LRT is shown to be a graph-smoothness detector, which can be implemented based solely on local measurements. Next, we develop the maximum likelihood (ML) decision rule for the binary hypothesis testing problems of each subset of edges. The ML decision rule consists of parallel LR Ts of the M-ary hypothesis testing problem of identifying all the combinations of edge disconnections in the network. We show that the ML decision rule consists of parallel LRTs of the binary hypothesis testing problems of each subset of edges. However, the ML decision rule leads to an exhaustive search over the candidate removed edges, which requires testing a number of hypotheses that is exponential in the number of edges and is practically infeasible. Thus, we propose a new low-complexity greedy approach for the identification problem that identifies a single disconnected edge at each iteration.

In this section, we present the background for GSP in Subsection II-A and define the smooth graph filter in Subsection II-B. Then, we describe the considered measurement model as an output of a smooth GSP filter in Subsection II-C.

A. Background: Graph Signal Processing (GSP)

Consider an undirected, connected, weighted graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of vertices and edges, respectively. The matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix of the graph, where $N \triangleq |\mathcal{V}|$ is the number of vertices in the graph. If there is an edge $(i, j) \in \mathcal{E}$ connecting vertices $i$ and $j$, the entry $W_{i,j} > 0$ represents the weight of the edge; otherwise, $W_{i,j} = 0$ [2]. A common way to represent the graph topology is by the Laplacian matrix, defined by

$$\mathbf{L} \triangleq \text{diag}(\mathbf{W}) - \mathbf{W}.$$  

(1)

The Laplacian matrix is a real PSD matrix. Hence, its associated eigenvalue decomposition (EVD) is given by

$$\mathbf{L} = \mathbf{U}^{(L)} \mathbf{\Lambda}^{(L)} (\mathbf{U}^{(L)})^T,$$

(2)

where the columns of $\mathbf{U}^{(L)} \in \mathbb{R}^{N \times N}$ are the eigenvectors of $\mathbf{L}$ and $(\mathbf{U}^{(L)})^{-1} = (\mathbf{U}^{(L)})^T$. The matrix $\mathbf{\Lambda}^{(L)} \in \mathbb{R}^{N \times N}$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{L}$, such that $0 = \lambda_1^{(L)} < \lambda_2^{(L)} \leq \ldots \leq \lambda_N^{(L)}$. For the sake of simplicity of presentation, the analysis and discussions in this paper are under the assumption that the graphs are connected graphs, i.e. they have a single zero eigenvalue [39].

A graph signal $\mathbf{a} : \mathcal{V} \to \mathbb{R}^N$ is an $N$-dimensional vector measured over the vertices of the graph. The graph Fourier transform (GFT) of the graph signal $\mathbf{a}$ is defined as [1], [2]:

$$\tilde{\mathbf{a}}^{(L)} = (\mathbf{U}^{(L)})^T \mathbf{a}.$$  

(3)

Similarly, the inverse GFT (IGFT) is obtained by left multiplication of $\tilde{\mathbf{a}}^{(L)}$ by $\mathbf{U}^{(L)}$. The graph Laplacian quadratic form, also named the Dirichlet energy, is defined as

$$Q_L(\mathbf{a}) \triangleq \mathbf{a}^T \mathbf{L} \mathbf{a} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} W_{i,j} (a_i - a_j)^2 = \sum_{(i,j) \in \mathcal{E}} W_{i,j} (a_i - a_j)^2,$$

(4)

where $\mathcal{N}_i$ is the set of vertices connected to vertex $i$ by an edge in $\mathcal{E}$, and the first equality in (4) is obtained by substituting (1). A smooth graph signal is signals with “small” Dirichlet energy in (4). Intuitively, since the weights are non-negative, a smooth graph signal is considered to be a good match with the graph if the signal values are close to their neighbors’ values [2], as described on the right-hand side (r.h.s.) of (4).

Substitution of (2) and (3) in (4) results in

$$Q_L(\mathbf{a}) = (\tilde{\mathbf{a}}^{(L)})^T \mathbf{\Lambda}^{(L)} \tilde{\mathbf{a}}^{(L)} = \sum_{n=1}^{N} \lambda_n^{(L)} (\tilde{a}_n^{(L)})^2,$$

(5)

where $\tilde{a}_n^{(L)}$ is the $n$th element of the GFT defined in (3), $n = 1, \ldots, N$. Therefore, a smooth graph signal with a small $Q_L(\mathbf{a})$ is associated with GFT coefficients that have a decaying behavior [2], [40].

II. MEASUREMENT MODEL: OUTPUT OF A GSP FILTER

In this section, we present the background for GSP in Subsection II-A and define the smooth graph filter in Subsection II-B. Then, we describe the considered measurement model as an output of a smooth GSP filter in Subsection II-C.

A. Background: Graph Signal Processing (GSP)

Consider an undirected, connected, weighted graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of vertices and edges, respectively. The matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix of the graph, where $N \triangleq |\mathcal{V}|$ is the number of vertices in the graph. If there is an edge $(i, j) \in \mathcal{E}$ connecting vertices $i$ and $j$, the entry $W_{i,j} > 0$ represents the weight of the edge; otherwise, $W_{i,j} = 0$ [2]. A common way to represent the graph topology is by the Laplacian matrix, defined by

$$\mathbf{L} \triangleq \text{diag}(\mathbf{W}) - \mathbf{W}.$$  

(1)

The Laplacian matrix is a real PSD matrix. Hence, its associated eigenvalue decomposition (EVD) is given by

$$\mathbf{L} = \mathbf{U}^{(L)} \mathbf{\Lambda}^{(L)} (\mathbf{U}^{(L)})^T,$$

(2)

where the columns of $\mathbf{U}^{(L)} \in \mathbb{R}^{N \times N}$ are the eigenvectors of $\mathbf{L}$ and $(\mathbf{U}^{(L)})^{-1} = (\mathbf{U}^{(L)})^T$. The matrix $\mathbf{\Lambda}^{(L)} \in \mathbb{R}^{N \times N}$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{L}$, such that $0 = \lambda_1^{(L)} < \lambda_2^{(L)} \leq \ldots \leq \lambda_N^{(L)}$. For the sake of simplicity of presentation, the analysis and discussions in this paper are under the assumption that the graphs are connected graphs, i.e. they have a single zero eigenvalue [39].

A graph signal $\mathbf{a} : \mathcal{V} \to \mathbb{R}^N$ is an $N$-dimensional vector measured over the vertices of the graph. The graph Fourier transform (GFT) of the graph signal $\mathbf{a}$ is defined as [1], [2]:

$$\tilde{\mathbf{a}}^{(L)} = (\mathbf{U}^{(L)})^T \mathbf{a}.$$  

(3)

Similarly, the inverse GFT (IGFT) is obtained by left multiplication of $\tilde{\mathbf{a}}^{(L)}$ by $\mathbf{U}^{(L)}$. The graph Laplacian quadratic form, also named the Dirichlet energy, is defined as

$$Q_L(\mathbf{a}) \triangleq \mathbf{a}^T \mathbf{L} \mathbf{a} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} W_{i,j} (a_i - a_j)^2 = \sum_{(i,j) \in \mathcal{E}} W_{i,j} (a_i - a_j)^2,$$

(4)

where $\mathcal{N}_i$ is the set of vertices connected to vertex $i$ by an edge in $\mathcal{E}$, and the first equality in (4) is obtained by substituting (1). A smooth graph signal is signals with “small” Dirichlet energy in (4). Intuitively, since the weights are non-negative, a smooth graph signal is considered to be a good match with the graph if the signal values are close to their neighbors’ values [2], as described on the right-hand side (r.h.s.) of (4).

Substitution of (2) and (3) in (4) results in

$$Q_L(\mathbf{a}) = (\tilde{\mathbf{a}}^{(L)})^T \mathbf{\Lambda}^{(L)} \tilde{\mathbf{a}}^{(L)} = \sum_{n=1}^{N} \lambda_n^{(L)} (\tilde{a}_n^{(L)})^2,$$

(5)

where $\tilde{a}_n^{(L)}$ is the $n$th element of the GFT defined in (3), $n = 1, \ldots, N$. Therefore, a smooth graph signal with a small $Q_L(\mathbf{a})$ is associated with GFT coefficients that have a decaying behavior [2], [40].
B. Smooth graph filter

Some of these graph signals may be represented as white noise that has been filtered on the graph topology by a graph filter. A graph filter is defined as follows [2]:

\[ h(L) \triangleq U(L)h(A(L))(U(L))^T \]

where \( h(\cdot) \) is the transfer function of the filter. A graph filter is a linear operator applied on an input graph signal, \( a_{in} \), such that the output graph signal, \( a_{out} \), satisfies [2, 35, 41]

\[ a_{out} = h(L)a_{in}. \]  

(7)

In this paper, we consider smooth graph filters. Table I presents smooth graph filters that are commonly used in the GSP and network science literature.

| Graph Filter                  | \( h(\lambda) \) | Covariance matrix of \( a_{out} \) for \( Cov(a_{in}) = 1 \) |
|-------------------------------|-------------------|-----------------------------------------------------------|
| GMRF with a Laplacian precision matrix [35, 36] | \( \frac{1}{\lambda} \lambda \neq 0 \) \( \lambda = 0 \) (8) | \( L^+ \) |
| Regularized Laplacian (Tikhonov) [35, 43] | \( \frac{1}{1 + \alpha \lambda} \alpha > 0 \) (9) | \( (I + \alpha L)^{-2} \) |
| Heat Diffusion Kernel [13, 35, 43] | \( \exp(-\tau \lambda) \), \( \tau > 0 \) (10) | \( \exp(-2\tau L) \) |

TABLE I: Examples of smooth graph filters.

In the following, we give a formal definition of smooth graph filters, which states that the expected Dirichlet energy of the output graph signal is lower than that of the input graph signal.

**Definition 1.** Let the elements of the input graph signal, \( a_{in} \), be independent and identically distributed (i.i.d.) zero-mean Gaussian random variables. Then, \( h(\cdot) \) from (6) is a smooth graph filter if

\[ \frac{E[Q_L(a_{out})]}{E[Q_L(a_{in})]} < 1, \]  

(11)

where the Dirichlet energy, \( Q_L(\cdot) \), is defined in (3) and \( a_{out} \) is given in (7).

By using (4), it can be verified that smoothness according to Definition 1 is satisfied for the graph filters in (9) and (10). In addition, for the GMRF graph filter from (8), (11) holds under the assumption that the Laplacian matrix satisfies \( \text{Tr}(L) > 1 \). Smooth graph signal corresponds to slow variations within the neighboring/connected vertices [2, 44]. Similarly, it can be shown that these filters are low-pass graph filters [45, 46].

C. Measurement model

In this paper, we consider a graph signal model as an output of a smooth graph filter, \( h(L) \), in the form given in (7), where the input graph signal is white Gaussian noise. This model is presented in [35–37] and it is shown that many signals over networks could be represented by this model, e.g. in power systems [40–43]. Thus, the measurement model is given by

\[ y[m] = h(L)x[m] + w[m], \ m = 1 \ldots M, \]  

(12)

where \( m \) is a time index, \( \{x[m]\}_{m=1}^M \), is a sequence of i.i.d. Gaussian random vectors with zero mean and a diagonal covariance matrix, \( \sigma_x^2 I \), i.e. \( x[m] \sim N(0, \sigma_x^2 I) \), \( m = 1, \ldots, M \), where \( \sigma_x^2 > 0 \) is assumed to be known. The graph filter, \( h(\cdot) \), is assumed to be known. The noise \( \{w[m]\}_{m=1}^M \), is a sequence of i.i.d. zero–mean, Gaussian random vectors with a covariance matrix \( \sigma_w^2 I \), where \( \sigma_w^2 > 0 \) is known, i.e. \( w[m] \sim N(0, \sigma_w^2 I) \), \( m = 1 \ldots M \). It is also assumed that the sequences \( \{x[m]\}_{m=1}^M \) and \( \{w[m]\}_{m=1}^M \) are mutually independent.

Under these assumptions, the output graph signal, \( y[m] \), \( m = 1, \ldots, M \), from (12), is a sequence of i.i.d. zero–mean, Gaussian random vectors with the covariance matrix

\[ \Sigma(L) \triangleq \sigma_x^2 h^2(L) + \sigma_w^2 I = \sigma_x^2 U(L)h^2(A(L))(U(L))^T + \sigma_w^2 I, \]  

(13)

where we use the fact that \( h(L)h^T(L) = h^2(L) \) since \( L \) is a symmetric matrix, and the last equality stems from (6). As a result, the log-likelihood of the augmented output vector of \( M \) time samples, \( \hat{y} \triangleq [y^T[1], \ldots, y^T[M]]^T \), is

\[ \log f(y; L) = -\frac{MN}{2} \log(2\pi) - \frac{M}{2} \log \left( \sigma_x^2 h^2(L) + \sigma_w^2 I \right) + \frac{M}{2} \text{Tr} \left( \sigma_x^2 h^2(L) + \sigma_w^2 I \right) S_y, \]  

(14)

where the sample covariance matrix is given by

\[ S_y \triangleq \frac{1}{M} \sum_{m=1}^M y[m]y^T[m]. \]  

(15)

The use of the pseudo-inverse and the pseudo-determinant in the log-likelihood in (14) is since the covariance matrix in (13) may be a singular matrix. For example, for the GMRF graph filter in (8) from Table I with \( \sigma_w^2 = 0 \), the matrix in (13) is reduced to

\[ \Sigma(L) = \sigma_x^2 h^2(L) + \sigma_w^2 I = \sigma_x^2 L^+, \]  

(16)

which is a singular matrix. More details about the use of a singular covariance matrix for the Gaussian distribution can be found in [49, 50].

It can be seen from (13) that in the considered model the graph topology determines the covariance matrix of the output graph signal. In fact, the graph filter, \( h(L) \), “colors” the input graph signal using the network connectivity. Therefore, when there are edge disconnections it affects the covariance matrix for the graph signal. In this work, we assume that the measurements are obtained from the model in (12) and our goal is to detect (Section III) and localize (Section IV) edge disconnection based on this model.

III. DETECTION OF A TOPOLOGY CHANGE

In this section, we develop the LRT for detecting the status (connected/disconnected) of a specific set of edges, based on the graph filter output model described in Subsection II-C.

In this section, we assume that the exact topology of the
graph is known under both hypotheses. In Subsection III-A and Subsection III-B, we present the hypothesis testing of this problem and develop the appropriate LRT, respectively. In Subsection III-C, we present the LRT in the graph spectral domain. Then, in Subsection III-D, we develop the LRT for the special case of a GMRF graph filter.

A. Problem formulation: Detection

We consider two candidate graphs, \( G^{(0)} = (V, \mathcal{E}^{(0)}, \mathbf{W}^{(0)}) \) and \( G^{(k)} = (V, \mathcal{E}^{(k)}, \mathbf{W}^{(k)}) \), that are associated with the Laplacian matrices \( \mathbf{L}^{(0)} \) and \( \mathbf{L}^{(k)} \), respectively. In fact, in this section the index \( k \) can be treated as \( k = 1 \), and the use of a general index \( k \) is only for the sake of simplicity of presentation for the case of M-ary hypothesis testing in the following sections. The graph \( G^{(0)} = (V, \mathcal{E}^{(0)}, \mathbf{W}^{(0)}) \) is the initial graph model of the network, representing the normal condition scenario with the associated known Laplacian matrix, \( \mathbf{L}^{(0)} \).

The graph \( G^{(k)} \) is a subgraph of \( G^{(0)} \), which is obtained by removing a series of edges, \( C(k) \), from the edge set, \( \mathcal{E}^{(0)} \). Thus, the set of edges satisfies \( \mathcal{E}^{(k)} = \mathcal{E}^{(0)} \setminus C(k) \). The matrix \( \mathbf{E}^{(i,j)} \), which corresponds to a single-edge disconnection of the edge \((i,j) \in \mathcal{E}^{(0)} \), is defined as

\[
\mathbf{E}^{(i,j)} \triangleq I^{(0)}(e_i e_j^T + e_j e_i^T - e_i e_i^T - e_j e_j^T).
\]

Accordingly, the Laplacian matrix, \( \mathbf{L}^{(k)} \), can be written as the sum of single-edge disconnections as follows:

\[
\mathbf{L}^{(k)} \triangleq \mathbf{L}^{(0)} - \sum_{(i,j) \in C^{(k)}} \mathbf{E}^{(i,j)}.
\]

Thus, the sufficient statistics of the LRT in the time domain is the sample covariance matrix, \( \mathbf{S}_y \).

C. Interpretation of the LRT in the graph spectral domain

In this subsection, we present the GSP form of the LRT from (22) based on its projection onto the graph spectral domain. For the sake of simplicity, we assume that \( \Sigma(L^{(k)}) \) and \( \Sigma(L^{(0)}) \), as defined in (13), are non-singular matrices. Then, by applying the matrix inversion lemma (see, e.g. Eq. (1) in [55]) on the r.h.s. of (13) after substituting \( \mathbf{L} = \mathbf{L}^{(k)} \), we obtain that the inverse of the covariance matrix satisfies

\[
(\Sigma(L^{(k)}))^{-1} = (\Sigma^{2}h^{2}(L^{(k)}) + \Sigma_x^{2}I)^{-1}
\]

\[
= \frac{1}{\Sigma_x^{2}}I - \frac{\Sigma_x^{2}}{\Sigma_x^{2}}U^{(L^{(k)})}h(\Lambda^{(L^{(k)})})^{-1}h(\Lambda^{(L^{(k)})})^{T}U^{(L^{(k)})}.
\]

Moreover, by using the GFT definition in (3), we obtain that the GFT of the output graph signal at time \( m \) with respect to (w.r.t.) the Laplacian matrix \( \mathbf{L}^{(k)} \) is

\[
\hat{\psi}_n^{(L^{(k)})}[m] = (U(L^{(k)}))^{T}y[m], m = 1, \ldots, M.
\]

Thus, the \( n \)-th element of the mean-squared GFT of the output graph signal, is defined as follows:

\[
\psi_n^{(L^{(k)})} \triangleq \frac{1}{M} \sum_{m=1}^{M} ([\hat{\psi}_n^{(L^{(k)})}[m]]^2, n = 1, \ldots, N.
\]

The expression in (27) can be interpreted as the \( n \)-th graph-frequency energy level. By substituting (15), (25), and (27) for general \( k \) and for \( k = 0 \) in (23), the LRT can be rewritten as

\[
l(y|\mathbf{L}^{(k)}) = \frac{\sigma_x^{2}}{\sigma_w^{2}} \left( \sum_{n=1}^{N} \frac{\sigma_w^{2}}{\sigma_x^{2}} + \frac{\sigma_x^{2}}{\sigma_w^{2}}h^{2}(\Lambda^{(L^{(0)})}) \psi_n^{(L^{(0)})} - \frac{\sigma_x^{2}}{\sigma_w^{2}} + \frac{\sigma_x^{2}}{\sigma_w^{2}}h^{2}(\Lambda^{(L^{(0)})}) \psi_n^{(L^{(0)})} \right),
\]
where, since \( h(\Lambda^{(L)}) \) is a diagonal matrix, we have
\[
\left[ h(\Lambda^{(L)})(\sigma_w^2 I + \sigma_x^2 h^2(\Lambda^{(L)}))^{-1} h(\Lambda^{(L)}) \right]_{i,j} = \begin{cases} \frac{h^2(\Lambda^{(L)})}{\sigma_w^2 + \sigma_x^2 h^2(\Lambda^{(L)})^2}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}
\] (29)
for any Laplacian matrix, \( L \). It can be seen from (28) that the sufficient statistic for the LR T is the graph-frequency energy levels, \( \{\psi_n^{(0)}, \psi_n^{(k)}\}_{n=1}^N \). Moreover, for smooth graph filters, as defined in Definition 1, the weights of the graph-frequency energy levels, \( \{\psi_n^{(0)}, \psi_n^{(k)}\}_{n=1}^N \)
includes only measurements that are measured at the vertices. In particular, \( \psi_n^{(0)} \) is a smooth
signal at neighboring vertices.

D. LRT for the GMRF model with a Laplacian precision

The Gaussian graphical model (GGM) is a prominent model that aims to handle large amounts of data collected in various applications and estimate the structure behind the data [56]. Under this model, there is an exact correspondence between the non-zero entries in the precision matrix and the existence of an edge between the relevant vertices [42]. In particular, the use of the graph filter in [29] leads to a GMRF model with the Laplacian as the precision matrix [35]. In this subsection, we discuss the LRT for the special case of a GMRF model.

Consider the GMRF in a noiseless scenario, \( \sigma_w^2 = 0 \), then by substituting (15) and (16) in (23) and using (29), the Laplacian as the precision matrix [35]. In this subsection, we discuss the LRT for the special case of a GMRF model.

Fig. 1: Illustration of the subsets: A graph \( G^{(0)} = (\mathcal{V}, E^{(0)}, W^{(0)}) \) with \( N = 7 \) vertices and \( |E^{(0)}| = 9 \) edges. The graph \( G^{(k)} \) is obtained by removing the series of edges, \( C^{(k)} = \{(1, 4), (4, 6), (3, 5)\} \) that are associated with the subset of vertices, \( S^{(k)} = \{1, 3, 4, 5, 6\} \). If we assume that all the weights are equal to 1, then, according to (19), in this example
\[
E^{(k)} = \sum_{(i,j) \in S^{(k)}} E^{(i,j)} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & -1 & 0 \\
-1 & 0 & 0 & 2 & 0 & -1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
\]
In Fig. 1 we illustrate the notations used in this paper for the different subsets over an arbitrary graph.

By using Definition 2, we can verify that the terms of the matrix \( E^{(k)} \) from (19) satisfy
\[
E^{(k)}_{m,l} = 0, \text{ if } m \notin S^{(k)} \text{ and/or } l \notin S^{(k)}. \tag{33}
\]
Hence, by substituting (18) in the first row of (30), we obtain
\[
l(y|L^{(k)}) = \frac{1}{\sigma_x^2} \text{Tr}(E^{(k)} S_y) = \frac{1}{\sigma_x^2} \sum_{i=1}^N \sum_{j=1}^N E^{(k)}_{i,j} S_y_{i,j} \\
= \frac{1}{\sigma_x^2} \sum_{(i,j) \in S^{(k)}} \sum_{j \in S^{(k)}} E^{(k)}_{i,j} S_y_{j,i} \\
= \frac{1}{\sigma_x^2} \text{Tr}([E^{(k)}]_{S^{(k)}} S_y_{S^{(k)}}), \tag{34}
\]
where the third equality is obtained by substituting (33) and the last equality is obtained by using the trace operator properties. From the observations in (31) and (34), it can be concluded that the measurements at all vertices that do not belong to \( S^{(k)} \) do not affect the LRT and are non-informative for the hypothesis testing in (20). This conclusion is aligned with the Hammersley-Clifford theorem [57], which states that a probability density function (pdf) that satisfies Markov properties w.r.t an undirected graph \( G \), such as the GMRF model, can be factorized into positive functions defined on cliques that cover all the vertices and edges of the graph. As a result, the LRT for the GMRF model in (34) is the difference between two log-likelihood functions that satisfy Markov properties and is only evaluated at the vertices that are in the subtraction of the cliques under hypothesis \( \mathcal{H}_1 \) from the cliques under \( \mathcal{H}_0 \), which are exactly the vertices in \( S^{(k)} \).
IV. IDENTIFICATION OF EDGE DISCONNECTIONS

In this section, we investigate the problem of identifying general edge disconnections in a network that are not limited to a specific set of edges. In Subsection IV-A, we formulate the M-ary hypothesis testing of the edge disconnections identification problem. In Subsection IV-B, we develop the ML decision rule for this hypothesis-testing problem. In Subsection IV-C, we discuss the properties of the ML decision rule and its computational complexity.

A. Problem formulation: Identification

We consider the following identification problem: the graph \( G^{(0)} = (\mathcal{V}, \mathcal{E}^{(0)}, \mathbf{W}^{(0)}) \) is the initial graph model of the network, representing the normal condition scenario with the associated known Laplacian matrix, \( \mathbf{L}^{(0)} \). Our purpose is to identify the graph topology from a set of possible graphs, \( \{G^{(k)} = (\mathcal{V}, \mathcal{E}^{(k)}, \mathbf{W}^{(k)})\}_{k=0}^{K} \), where \( \mathbf{L}^{(k)} \) represents the Laplacian matrix of the corresponding graph \( G^{(k)} \). Each graph, \( G^{(k)}, k = 1, \ldots, K \), is a subgraph of \( G^{(0)} \) with \( \mathcal{E}^{(k)} = \mathcal{E}^{(0)} \setminus \mathcal{C}^{(k)} \), where \( \mathcal{C}^{(k)} \) is the kth set of edge disconnections, as described for the binary hypothesis testing in Subsection III-A.

The identification problem between \( K+1 \) possible hypotheses based on \( M \) graph signals from \( \{y[m], m = 1, \ldots, M\} \), can be stated as an M-ary hypothesis testing problem:

\[
\mathcal{H}_0: \quad y[m] = h(\mathbf{L}^{(0)})x[m] + w[m], \\
\mathcal{H}_k: \quad y[m] = h(\mathbf{L}^{(k)})x[m] + w[m], \quad m = 1, \ldots, M, \quad (35)
\]

for \( k = 1, \ldots, K \), where \( \mathbf{L}^{(k)} \) is defined in (13). This problem is a simple multiple hypothesis testing problem, where all the parameters under each hypothesis are known.

B. ML decision rule

The ML decision rule maximizes the log-likelihood of the \( K+1 \) hypotheses in (35) and is given by (51)

\[
\xi(y) = \arg \max_{0 \leq k \leq K} \log f(y; \mathbf{L}^{(k)}), \quad (36)
\]

where \( \log f(y; \mathbf{L}^{(k)}) \) is the log-likelihood under hypothesis \( \mathcal{H}_k \), which is obtained by substituting \( \mathbf{L} = \mathbf{L}^{(k)} \) in (14), for \( k = 0, 1, \ldots, K \). By dividing each of the \( K+1 \) log-likelihood functions on the r.h.s. of (36) by the positive log-likelihood of the null hypothesis, \( \log f(y; \mathbf{L}^{(0)}) \), (36) can be rewritten as

\[
\xi(y) = \arg \max_{0 \leq k \leq K} \left\{ \log \left( \frac{f(y; \mathbf{L}^{(k)})}{f(y; \mathbf{L}^{(0)})} \right) \right\}. \quad (37)
\]

By substituting (13) for each likelihood function in (37) and removing constant terms, the ML decision rule satisfies

\[
\xi(y) = \arg \max_{0 \leq k \leq K} l(y|\mathbf{L}^{(k)}) - \rho(\mathbf{L}^{(k)}), \quad (38)
\]

where \( l(y|\mathbf{L}^{(k)}) \) and \( \rho(\mathbf{L}^{(k)}) \) are defined in (24) and (24), respectively. It can be seen from (38) that under hypothesis \( \mathcal{H}_0 \) we obtain \( l(y|\mathbf{L}^{(0)}) - \rho(\mathbf{L}^{(0)}) = 0 \).

The term \( l(y|\mathbf{L}^{(k)}) \) is the LRT for the binary hypothesis testing between \( \mathcal{H}_0 \) and \( \mathcal{H}_k \), i.e. detecting a specific edge disconnection \( \mathcal{C}^{(k)} \), as described in (20). Thus, the ML decision rule in (38) consists of two stages: first, implementing \( K \) binary LRTs in parallel, where each LRT is summed with an appropriate penalty function, \(-\rho(\mathbf{L}^{(k)})\), which is independent of the measurement vector, \( y \). Second, the ML decision rule selects the hypothesis associated with the maximal values, where 0 is the value of the null hypothesis, \( \mathcal{H}_0 \). The ML decision rule is illustrated in Fig. 2.

C. Remarks on the ML decision rule

1) Penalty function interpretation: The data-independent term, \( \rho(\mathbf{L}^{(k)}) \) in (24), can be interpreted as a penalty function on the considered topology change. In order to demonstrate this interpretation, the following proposition describes the order relation between the penalty functions.

**Proposition 1.** Consider two connected graphs, \( G^{(k_1)} \) and \( G^{(k_2)} \), associated with the Laplacian matrices, \( \mathbf{L}^{(k_1)} \) and \( \mathbf{L}^{(k_2)} \), respectively, and assume the following conditions:

C.1) The matrices \( \Sigma(\mathbf{L}^{(k_1)}) \) and \( \Sigma(\mathbf{L}^{(k_2)}) \), defined in (13), are non-singular matrices.

C.2) \( \mathcal{C}^{(k_2)} \) is a proper subset of \( \mathcal{C}^{(k_1)} \), i.e. \( \mathcal{C}^{(k_2)} \subset \mathcal{C}^{(k_1)} \).

C.3) the graph filter, \( h(\lambda) \), is a monotonic decreasing function of \( \lambda \) for any \( \lambda > 0 \).

Then,

\[
\rho(\mathbf{L}^{(k_2)}) \leq \rho(\mathbf{L}^{(k_1)}). \quad (39)
\]

**Proof:** The proof appears in Appendix A.

It can be seen that the graph filters from (8)-(10) in Table I are all monotonic decreasing functions for \( \lambda > 0 \) and, thus, satisfy the conditions of Proposition 1. According to this proposition, in the case of nested subsets of edge disconnections, the ML decision rule in (38) imposes a larger penalty on models with a larger number of edge disconnections, in order to avoid overfitting (58).

2) Interpretation in the graph spectral domain: Similar to in Subsection III-C, the ML decision rule from (35) can be written in the graph spectral domain using the result in (28) for the \( k \)th hypothesis. That is, the ML decision rule in (38)
can be written as
\[
\xi(y) = \arg\max_{0 \leq k \leq K} \frac{\sigma_w^2}{\sigma_w^2 + \sigma_w^2 \sum_{n=1}^{N} \frac{h^2(\lambda_n^{(L_k)})}{\lambda_n^{(L_k)}} \psi_n^{(L_k)}} - \frac{h^2(\lambda_n^{(L_0)}) \psi_n^{(L_0)}}{\sigma_w^2 + \sigma_w^2 \lambda_n^{(L_0)}} - \rho(L_k),
\]
where \(\rho(L_k)\) is defined in (24), and the \(n\)th graph-frequency energy level w.r.t. \(L_k\), \(\psi_n^{(L_k)}\), is defined in (27). It should be noted that the representation in the graph spectral domain emphasizes the fact that the ML decision rule only requires the evaluation of the \(NK\) scalar parameters, \(\psi_n^{(L_k)}\), for any \(k = 0, \ldots, K\), \(n = 1, \ldots, N\). Thus, in contrast with covariance matrix estimation problems, there is no need to assume that the sample covariance matrix from (15), \(S\), is only a function of the measurements and the second-order local statistics and searching only in the neighborhood.

3) GMRF model with a Laplacian precision: Similar to Subsection II-D, the following proposition states that for the special case of a noiseless GMRF with a Laplacian precision matrix, the ML decision rule can be calculated locally for each candidate set \(k = 1, \ldots, K\), by observing only the related measurements in \(S(k)\), defined in Definition 2.

Proposition 2. Consider a connected graph, \(G^{(k)}\), associated with the Laplacian matrix, \(L_k\). Then, for the noiseless GMRF filter in (5), the term \(l(y|L^{(k)}) - \rho(L^{(k)})\) on the r.h.s. of (33) is only a function of the measurements and the second-order statistics that are associated with the vertices in \(S(k)\).

Proof: It was shown in (24) that \(l(y|L^{(k)})\) is only a function of the measurements that are measured at vertices in \(S(k)\). In addition, in Appendix B we prove that under the assumptions of Proposition 1, \(\rho(L^{(k)})\) is only a function of the matrices \(\Sigma(L^{(k)})\Sigma^{(k)}\) and \(\Sigma(L^{(0)})\Sigma^{(0)}\), i.e. of the second-order statistics of the graph signal over the vertices in \(S(k)\). Thus, the \(k\)th term in (33) depends only on the observation measured at the vertices in \(S(k)\) and their statistics. \(\blacksquare\)

4) Computational complexity: The ML decision rule in (33) is based on evaluating the likelihood for each of the \(K\) candidate hypotheses and then selecting the maximum among them. In the general worst-case scenario, where any of the edges can disconnect, we have
\[
K = \sum_{r=1}^{r_{\text{max}}} \binom{\mathcal{E}(0)}{r},
\]
where \(r_{\text{max}}\) denotes the considered maximum number of possible edge disconnections in the network. Such a combinatorial optimization problem suffers from high computational complexity due to the exhaustive search over the set of possibilities for each suspicious edge (connected/disconnected). Moreover, the calculation of \(l(y|L^{(k)})\) and \(\rho(L^{(k)})\) in (23) and (24), respectively, requires complexity of \(N\)-dimensional matrix inversion. Thus, the computational complexity of the ML decision rule grows exponentially with the graph size and it would be impractical for practical networks. In the next section, we develop efficient low-complexity methods to deal with the edge disconnections identification problem.

V. GREEDY APPROACHES FOR IDENTIFYING EDGE DISCONNECTIONS

In this section, we propose low-complexity greedy approaches for edge disconnections identification. We first present the basic greedy method in Subsection V-A. Then, in Subsection V-B we present a modification: a neighboring strategy that is combined with the greedy method to reduce the computational complexity even further. Some remarks on these approaches are discussed in Section V-C.

A. Greedy approach

We assume a nominal topology, where only a small percentage of edges may be disconnected, i.e. \(|\mathcal{C}(k)| \ll |\mathcal{E}(0)|\), \(k = 1, \ldots, K\), and the matrices \(E^{(k)}\), \(k = 1, \ldots, K\), are sparse. A commonly-used heuristic for combinatorial problems over graphs is a greedy algorithm [59], which starts with an empty set and then, iteratively, in each step, adds the edge which maximizes the objective. In the considered identification problem, we propose a greedy approach, given in Algorithm 1 that starts with \(C = \emptyset\) at the first iteration. Then, at the \(l\)th iteration, we test all the available edges in the graph and choose the edge that maximizes the marginal likelihood in (33) for a single edge:
\[
\hat{k} = \arg\max_{k=1}^{K} l(y|\mathbf{L}^l - E^{(k)}) - \rho(\mathbf{L}^l - E^{(k)}),
\]
where \(\mathbf{L}^l\) and \(\mathbf{L}^l\) are the edge set and the Laplacian matrix at the \(l\)th iteration that are initialized by \(\mathbf{L}^0 = \mathbf{L}^{(0)}\), respectively, where in the left terms, “0” denotes the iteration index. The functions \(l(\cdot)\) and \(\rho(\cdot)\) in (42) are defined in (23) and (24), respectively. Afterwards, we test if the likelihood ratio of the chosen edge, \(\hat{k}\), is higher than zero:
\[
l(y|\mathbf{L}^l - E^{(k)}) - \rho(\mathbf{L}^l - E^{(k)}) > 0.
\]
If (43) is satisfied, then: 1) we add \(\hat{k}\) to the edge disconnections set, \(\mathcal{C}^l\); 2) we update the edge set to \(\mathbf{E}^{l+1}\) by removing \(\hat{k}\) from the current edge set, \(\mathbf{E}^l\); and 3) we modify the Laplacian matrix \(\mathbf{L}^l\), such that \(\mathbf{L}^{l+1}\) excludes the \(\hat{k}\) edge. Otherwise, the algorithm stops. The rationale behind the stop condition in (43) is that the zero value is associated with the null hypothesis, \(H_0\) of no disconnections, since the likelihood ratio satisfies \(l(y|\mathbf{L}^{(0)}) - \rho(\mathbf{L}^{(0)}) = 0\). If the maximum of edge disconnections, \(r_{\text{max}}\), is known, then it can be used as an additional stopping condition. In addition, a restriction to a specific set of possible edge disconnections, \(\mathcal{C}^{(k)}\), \(k = 0, 1, \ldots, K\), can be done by adding a projection step at the end of Algorithm 1.

B. Greedy approach with a neighboring strategy

The computational complexity of the greedy algorithm in Algorithm 1 may still be too high for large networks. This is due to the fact that, according to (23) and (24), evaluating the r.h.s. on (42) at each iteration, requires computing around \(|\mathcal{E}(0)|\) \(N\)-dimensional inverse matrices. In this section we propose additional simplifications by computing the local statistics and searching only in the neighborhood.
of the suspected edges. The neighboring strategy is inspired by the local property of the GMRF filter in Proposition 2, which states that we only need to consider the vertices in \( S^{(k)} \). Here, we use the vertices in the \( \beta \)-neighborhood of the suspicious edges for general smooth graph filters that are expected to have similar values at neighboring vertices. The tunable parameter \( \beta \) is the number of neighbors taken into consideration, providing a trade-off between the identification accuracy and the computational cost.

The neighboring strategy comes to ease the greedy algorithm’s complexity by: 1) calculating the \( \beta \)-local ML decision rule for a single edge, where for a given candidate edge, \( (i,j) \in \hat{E}^l \), we calculate the likelihood ratio of the measurements in the set \( \mathcal{N}((i,j),\beta) \triangleq N(i,\beta) \cup N(j,\beta) \), where \( N(i,\beta) \) is the set of vertices connected to vertex \( i \) by a path of at most \( \beta \) edges; and 2) building the new set \( E^{l+1} \) of the suspicious edges for the \( l+1 \) iteration, where for sparse graphs, \( E^{l+1} \) is significantly smaller than \( E^0 \), which is required for searching over all the edges in the graph. This two-step approach is as follows.

1) Calculating the \( \beta \)-local ML decision rule: In order to compute the likelihood locally, the greedy iteration from (42) is replaced by the local ML decision rule that we define by

\[
\hat{k} = \arg \max_{k=(i,j)} \Phi_1(\mathbf{y},\hat{L}^i, E^{(i,j)}, \mathcal{N}((i,j),\beta)) - \Phi_2(\hat{L}^i, E^{(i,j)}, \mathcal{N}((i,j),\beta)),
\]

where

\[
\Phi_1(\mathbf{y}, \mathbf{L}, \mathbf{E}, S) \triangleq \text{Tr}\left( \sigma_x^2 [h^2(\mathbf{L})]_S + \sigma_w^2 \mathbf{I}^+ | S_Y | S \right) - \text{Tr}\left( \sigma_x^2 [h^2(\mathbf{L} - \mathbf{E})]_S + \sigma_w^2 \mathbf{I}^+ | S_Y | S \right)
\]

and

\[
\Phi_2(\mathbf{E}, \mathbf{L}, S) \triangleq \log \left( \frac{\sigma_x^2 [h^2(\mathbf{L} - \mathbf{E})]_S + \sigma_w^2 \mathbf{I}^+}{\sigma_x^2 [h^2(\mathbf{L})]_S + \sigma_w^2 \mathbf{I}^+} \right).
\]

The expressions in (45) and (46) are the local equivalent to the expressions in (23) and (24), respectively. For the special case where \( \mathcal{N}((i,j),\beta) = \mathcal{V} \), which is obtained by taking a maximal value of \( \beta \), (44) is reduced to (42) from the greedy approach in Algorithm 1.

2) Building a suspicious edge set: In order to build a new subset of suspicious edges, we initialize this set to \( \hat{E}^0 = E^0 \).

Then, in the \( l \)th iteration, we determine the search edge set for the \( l+1 \) iteration by

\[
\hat{E}^{l+1} = \{E^0 \cup E'_l \} \setminus \hat{C}^l,
\]

where

\[
E'_l = \left\{ (i,j) \in \hat{E}^l | \Phi_1(\mathbf{y},\hat{L}^i, E^{(i,j)}, \mathcal{N}((i,j),\beta)) - \Phi_2(\hat{L}^i, E^{(i,j)}, \mathcal{N}((i,j),\beta)) > 0 \right\}.
\]

The expressions in (45) and (46) are defined in (45) and (46), respectively, and

\[
E_{\beta} = \left\{ (i,j) \in E^0 \big| \exists (u,v) \in \{ \{ \mathcal{P}(i,\beta) \cup \mathcal{P}(j,\beta) \} \cap E'_l \} \right\}
\]

in which \( \mathcal{P}(i,\beta) \) is the set of edges in the shortest paths between the vertex \( i \) and the vertices in \( N(i,\beta) \). That is, we include in the search set in (47) only edges: a) with a non-negative \( \beta \)-local likelihood ratio, which is the term on the r.h.s. in (44), i.e. edges in \( E'_l \), defined in (48); and b) edges that are in the \( \beta \)-neighborhood of such edges as in a), i.e. edges in \( E_{\beta} \), defined in (49). The rationale behind this set is that the zero value is associated with the null hypothesis (of no disconnections). Thus, suspicious edges should have non-negative values or at least be in the local neighborhood of such edges.

The two-step greedy algorithm with neighboring strategy is summarized in Algorithm 2.

C. Remarks

1) Computational complexity: The computational complexity of the proposed greedy algorithms is significantly lower than those of the original ML decision rule due to the following reasons. First, in the general worst-case scenario, where any of the edges may be disconnected, the ML decision rule from (38) requires \( K \) calculations of the likelihood ratio, where \( K \) is a combinatorial term defined in (41). In contrast, Algorithms 1 and 2 are based on a search approach described in (42) and (44), respectively, that are performed over \( r_{max} \times |E^0| \) (or less) possibilities, which is significantly smaller than \( K \) for large networks. Second, if we assume that the considered graphs are sparse with a small degree, such that \( |E^0| \ll \frac{N(N-1)}{2} \), then, for small values of \( \beta \), most of the edges have small sets of their \( \beta \)-local neighborhood, that are used in Algorithm 2. In particular, the computational complexity of inverting matrices in Algorithm 2 is \( \mathcal{O}(|\mathcal{N}((i,j),\beta)|^3) \), since we have smaller matrices in (45) and (46), where the size depends on the dimension of the \( \beta \)-neighborhood of the specific edge \((i,j)\). This is in contrast with the computational complexity of the ML decision rule and Algorithm 1 that require computing the inverse of the \( N \)-dimensional matrices.

Algorithm 1: Greedy identification

Input:
- Sample covariance matrix, \( \mathbf{S}_y \)
- Initial Laplacian matrix, \( \mathbf{L}^{(0)} \), and its edge set, \( \mathcal{E}^{(0)} \)
- Signal and noise variances, \( \sigma_x^2 \) and \( \sigma_w^2 \)
- Graph filter, \( h(\cdot) \)
- Optional: Sparsity level, \( r_{max} \).

Output: Estimated edge disconnections set, \( \hat{C} \).

1. Initialize \( \hat{C}^0 = \emptyset \), \( \hat{E}^0 = \mathcal{E}^{(0)}, \mathbf{L}^0 = \mathbf{L}^{(0)} \), and \( l = 0 \).
2. Find the maximal edge, \( \hat{k} \in \hat{E}^l \), by (42).
3. If \( l(y|\hat{L}^l - E^{(k)}) - \rho(l(y|\hat{L}^l - E^{(k)})) > 0 \) then
   4. Update \( \hat{C}^{l+1} = \hat{C}^l \cup \{k\}, \hat{E}^{l+1} = \hat{E}^l \setminus \{k\} \).
   5. If \( |\hat{C}^l| = r_{max} \) then
      6. Return: \( \hat{C}^l \).
   7. Repeat to step 2.
8. Return: \( \hat{C}^l \).

Algorithm 2: Greedy identification with neighboring strategy

Input:
- Sample covariance matrix, \( \mathbf{S}_y \)
- Initial Laplacian matrix, \( \mathbf{L}^{(0)} \), and its edge set, \( \mathcal{E}^{(0)} \)
- Signal and noise variances, \( \sigma_x^2 \) and \( \sigma_w^2 \)
- Graph filter, \( h(\cdot) \)
- Optional: Sparsity level, \( r_{max} \).

Output: Estimated edge disconnections set, \( \hat{C} \).

1. Initialize \( \hat{C}^0 = \emptyset \), \( \hat{E}^0 = \mathcal{E}^{(0)}, \mathbf{L}^0 = \mathbf{L}^{(0)} \), and \( l = 0 \).
2. Find the maximal edge, \( \hat{k} \in \hat{E}^l \), by (42).
3. If \( l(y|\hat{L}^l - E^{(k)}) - \rho(l(y|\hat{L}^l - E^{(k)}) > 0 \) then
   4. Update \( \hat{C}^{l+1} = \hat{C}^l \cup \{k\}, \hat{E}^{l+1} = \hat{E}^l \setminus \{k\} \).
   5. If \( |\hat{C}^l| = r_{max} \) then
      6. Return: \( \hat{C}^l \).
   7. Repeat to step 2.
8. Return: \( \hat{C}^l \).
Algorithm 2: Greedy identification with a neighboring strategy

Input:
- Sample covariance matrix, \( S_y \)
- Initial Laplacian matrix, \( L^{(0)} \), and its edge set, \( E^{(0)} \)
- Signal and noise variances, \( \sigma_x^2 \) and \( \sigma_w^2 \)
- Graph filter, \( h(\cdot) \)
- Neighboring order, \( \beta \)
- Optional: Sparsity level, \( r_{max} \)

Output: The estimated edge disconnections set \( \hat{C} \).

1. Initialize \( \hat{C}^0 = \emptyset \), \( \hat{E}^0 = E^{(0)} \), \( L^0 = L^{(0)} \) and \( l = 0 \).
2. Find the maximal edge, \( \hat{k} \in \hat{C}^l \), by (44).
3. If \( \Phi_1(y, L, E^{(k)}, N(k, \beta)) - \Phi_2(L, E^{(k)}, N(\hat{k}, \beta)) > 0 \) then
   4. Update \( \hat{C}^{l+1} = \hat{C}^l \cup \{ \hat{k} \} \) and \( L^{l+1} = L^l - E^{(k)} \).
   5. Build the edge set \( \hat{E}^{l+1} \) by (47), and \( l \leftarrow l + 1 \).
6. If \( |\hat{C}^l| = r_{max} \) then
7. Return: \( \hat{C}^l \).
8. Repeat to step 2.
9. Return: \( \hat{C}^l \).

The estimated edge disconnections set \( \hat{C} \) is replaced by using the full-graph measurements. That is, \( \hat{C}^l \) is replaced by \( \hat{C}^l \cup \{ \hat{k} \} \) and \( L^{l+1} = L^l - E^{(k)} \).

VI. SIMULATIONS

In this section, we evaluate the performance of the LRT and the greedy approaches from Sections III and IV and compare them with the performance of state-of-the-art methods. In Subsection VI-A we describe the general experimental setting for the synthetic data and methods for comparison. In Subsections VI-B and VI-C we demonstrate the performance of the detection and identification approaches, respectively. Finally, we present simulations concerned with the identification of outages in power systems in Subsection VI-D.

A. Experimental settings and methods for comparison

In the simulations in Subsections VI-B and VI-C the data was generated according to the measurement model from Subsection II-C where the initial graph, \( G^{(0)} = (V, E^{(0)}, W^{(0)}) \), were generated by using the Watts-Strogatz small-world graph model \( [61] \), with \( N = 50 \) vertices, mean degree of \( K = 2 \), and \( |E^{(0)}| = 100 \). Similar results were obtained for other random graphs, such as graphs generated by the stochastic block model. These results were omitted from this paper due to space limitations. The elements of the adjacency matrix, \( W^{(0)} \), are independent uniform distributed weights in the range \([0, 1, 5]\). The graph after the change is obtained by removing an arbitrarily chosen set of edges \( e \) from \( E^{(0)} \). The output graph signals were generated by implementing (12), where \( \sigma_x^2 = 1 \). The graph filters from Table I were tested with parameters \( \alpha = 0.5 \) and \( \tau = 0.2 \). At least 1,000 Monte-Carlo simulations were conducted to evaluate the performance in each scenario.

The methods for comparison are as follows:

1) Naive smoothness detector: The naive smoothness detector measures the average smoothness of the output graph signal, \( y \), w.r.t. the initial Laplacian, \( L^{(0)} \):

\[
\frac{1}{M} \sum_{m=1}^{M} Q_{L^{(0)}}(y[m]) = \frac{1}{M} \sum_{m=1}^{M} y^T[m]L^{(0)}y[m] \leq_{H_0} H_{\gamma},
\]

where \( \gamma \) is a chosen threshold. The underlying assumption behind this detector is that if this quantity obtains a small value, then, it is more likely that \( y \) has been generated from \( G^{(0)} \); otherwise, \( y \) is associated with a different topology. In contrast with the LRT for the GMRF filter in (39), the naive smoothness detector in (51) does not assume that the topology after the change, represented by \( L^{(k)} \), is known.

2) Matched subspace detectors (MSDs) \([27],[28]\): The MSDs compare the energies of a graph signal \( y \) at high graph frequencies of the graph spectral domain under each hypothesis. We implement here 1) the simple-MSD (SMSD):

\[
\frac{1}{M} \sum_{m=1}^{M} \left( \left\| y_{L^{(0)} \backslash S_B}[m] \right\|_2^2 - \left\| y_{L^{(k)} \backslash S_B}[m] \right\|_2^2 \right) \leq_{H_0} H_{\gamma},
\]

where \( S_B = 1, \ldots, B \), and \( H_0, H_1 \) are the hypotheses in (20); and 2) the blind simple-MSD (BMSD):

\[
\frac{1}{M} \sum_{m=1}^{M} \left\| y_{L^{(0)} \backslash S_B}[m] \right\|_2^2 \leq_{H_0} H_{\gamma}.
\]

We determine the parameter to be \( B = \left\lceil \frac{N}{2} \right\rceil \). The difference between the SMDS and the BMSD is that the SMDS does assume that the topology after the change is known.

TABLE II: The computational complexity of the proposed methods.

| Method | \# Likelihood ratio calculation | Matrices inversion | Searching edge set size |
|--------|---------------------------------|--------------------|------------------------|
| ML decision rule | \( \sum_{r=1}^{r_{max}} (E^{(r)}) \) | \( O(N^3) \) | \( \sum_{r=1}^{r_{max}} (E^{(r)}) \) |
| Greedy approach | \( r_{max} \times |E^{(0)}| \) | \( O(N^3) \) | \( |E^{(0)}| \) |
| Greedy approach with neighboring strategy | \( r_{max} \times |E^{(0)}| \) | \( |N((i, j), \beta)|^3 \) | \( |E^{(0)}| \) |
3) Combinatorial graph Laplacian (CGL) method [34, 36]: The CGL is a Laplacian learning block-coordinate descent method, which is based on the Laplacian structure constraints. The CGL method was shown to be useful in cases of up to 25% mismatch in the connectivity of the graph (see Fig. 6 in [34]). In addition, the model behind the CGL method in [34, 36] results in the same covariance matrix as the covariance matrix of the considered model in [13]. Therefore, the CGL method is appropriate for comparison in the case of problems dealing with identification of edge disconnections in networks based on graph filter outputs. It is based on the connectivity matrix of $G$, $A$, which has the following $(i,j)$ element

$$A_{i,j} \triangleq \begin{cases} 1 \text{ if } W_{i,j} \neq 0 \\ 0 \text{ otherwise} \end{cases},$$

where $W$ is the adjacency matrix of $G$. In the following simulations, edge disconnection identification by the CGL method is implemented by the following four-step approach:

(a) A prefiltering operation is performed, as described in [36], by removing the noise variance and using the inverse graph filter, $h^{-1}()$, on the sample covariance matrix:

$$S_{y}^{\rho} \triangleq h^{-1} \left( \frac{1}{\sigma_{x}^{2}} (S_{y} - \sigma_{w}^{2} I) \right).$$

(b) The CGL method is implemented by using the code in [34] with the inputs: 1) the connectivity matrix of the known topology $G^{(0)}$, and 2) the prefILTER sample covariance matrix, $S_{y}^{\rho}$ from (55), and under the constrained set

$$\mathcal{L}_{c}(A) = \left\{ L | L \succeq 0, L1 = 0, L_{i,j} \leq 0 \text{ for } i \neq j, L_{i,i} = 0 \text{ if } A_{i,i} = 0 \right\}.$$  \hfill (56)

(c) The off-diagonal elements of the estimator of $L$ from Step (b) are thresholded, such that elements that are larger than $-\epsilon$ are set to zero, where we set $\epsilon = 0.1$.

(d) For any edge $(i,j) \in \mathcal{E}^{(0)}$, where the output of Step (c) satisfies $\hat{L}_{i,j} = 0$ but $L_{i,j}^{(0)} \neq 0$, $(i,j)$ is declared as a disconnection.

For sparse graphs, the computational complexity of the CGL method is $O(\Omega(d^3) + N^2)$, where $d = \max_{v \in \mathbb{V}} |N(v,1)|$ is the maximum degree of the graph [36].

4) Constrained CGL (CCGL) method: In order to have a fair comparison with the methods proposed in this paper, we also implement the CGGL method, which is based on the CGL method after adding the information regarding the initial Laplacian matrix, $L^{(0)}$. This information has been integrated into the CGL method by replacing (56) in Step (b) by the constraints set

$$\mathcal{L}_{cc}(A) = \left\{ L | L \succeq 0, L1 = 0, L_{i,j}^{(0)} \leq L_{i,j} \leq 0 \text{ if } A_{i,j} = 1, L_{i,i} = 0 \text{ if } A_{i,i} = 0 \right\}.$$  \hfill (57)

In addition, in Step (c), we change the thresholding by setting an individual thresholding for each entry in $\hat{L}$, $\epsilon_{i,j} = \frac{1}{2}L_{i,j}^{(0)}$.

5) GGM-GLRT [32]: The GGM-GLRT method for deciding if an edge $(i,j) \in \mathcal{E}$ is connected or disconnected is [32]:

$$\mathcal{H}_{1}^{(i,j)}: y \sim \mathcal{N}(\beta L^{(0)} \gamma),$$

where $H_{1}^{(i,j)}$ represents the hypothesis of a disconnection in edge $(i,j)$, and $H_{0}^{(i,j)}$ represents the hypothesis that $(i,j)$ is a connected edge. It can be seen that the GGM-GLRT detector on (58) is only a function of the first-order neighbors of the tested edge. For the identification problem, we implement the GGM-GLRT in (58) on all the edges in the original graph, i.e. $\forall (i,j) \in \mathcal{E}^{(0)}$, where the threshold $\gamma$ is experimentally determined. The computational complexity of the GGM-GLRT as an identification method is $O(|\mathcal{E}^{(0)}|d^3)$, where $d = \max_{v \in \mathbb{V}} |N(v,1)|$ is the maximum degree of the graph. It should be noted that the model with the GMRF filter in (8) is a special case of GGM with the Laplacian as the precision matrix. However, the GGM-GLRT method does not assume a Laplacian-based structure of the precision matrix and was developed for the noiseless case. Thus, it requires an accurate estimation of the covariance matrix by a large number of measurements, in contrast with the proposed methods.

B. Detecting a specific set of edge disconnections

In this subsection, we evaluate the performance of tests that detect a specific edge disconnection set for the graph filters from Table I with $r = 5$ disconnections, $M = 100$ time samples, and noise variance $\sigma_{w}^{2} = 2$. In Fig. 3 we present the Receiver Operating Characteristic (ROC) curves of the detectors: 1) the LRT in (23); 2) the local LRT in (50) for $\beta = 0.1, 3$; the naive smoothness detector in (30); and 4) the SMGMD detector in (52). It can be seen that the LRT and the local LRTs (with $\beta = 0.1$) outperform the other detectors for any probability of false alarm and for any tested smooth graph filter. In addition, the performance of the LRT and local LRT with $\beta = 1$ is almost identical for any graph filter from Table I. Thus, we can conclude that, in the considered cases, by taking into account only the first-order neighbors of the disconnected edges, we obtain the global performance that is based on measurements from the full graph. This property is due to the smoothness of the output graph signal and is a significant advantage when implementing in large networks.

C. Identifying edge disconnections

In the following, we evaluate the performance of the proposed greedy algorithms from Subsection V in terms of detection and identification performance. We assume the worst-case scenario, where any of the edges can be disconnected. The results of the ML decision rule are not shown due to its computational complexity, which makes it impractical.

In Fig. 4 we evaluate the performance of the detection question: is there any disconnection in the topology? That is, the null hypothesis is $H_0$, while the alternative includes the union of all the other options, $H_1, \ldots, H_K$. According to the ML decision rule (38), we evaluate the performance of the detector

$$l(y | L^{(0)} - \hat{E}) - r(L^{(0)} - \hat{E}) \gtrless H_0.$$  \hfill (59)
where $\hat{E}$ is obtained from the estimated edge disconnection, $\hat{C}$, by either the greedy algorithm from Algorithm 1 denoted “full” greedy from Algorithm 2 with $\beta = 0, 1$. In each algorithm, we limit the maximum number of possible edge disconnections in the network to $r_{\text{max}} = 10$. We compare these detectors with the naive smoothness detector from (51) and with the BMDs detector from (53). Figure 4 presents the ROC curves of these methods for the graph filters from Table 1 with $r = 5$ disconnected edges, a noise variance of $\sigma_w^2 = 0.5$, and $M = 100$ time samples. It can be seen that the edge disconnections detectors that were derived from Algorithms 1 and 2 outperform the other detectors for any probability of false alarm and for any tested smooth graph filter. In addition, similar to the results in Fig. 3 the performance of the greedy algorithm with neighboring strategy and $\beta = 1$ is close to that of the “full” greedy algorithm. Thus, the first one should be preferred in practice since it has a lower computational complexity.

For evaluating the classification performance, we use the F-score measure [62]:

$$F_{\text{score}} = \frac{2 \cdot \hat{t}_p \hat{f}_n}{2 \cdot \hat{t}_p \hat{f}_n + \hat{f}_p + \hat{f}_n},$$

where $t_p$, $f_p$, $f_n$ correspond to the true-positive, false-positive, and false-negative of the detection of disconnected edges in the $l$th simulation. We compare the following topology identification methods: 1) the greedy approach in Algorithm 1; 2) the greedy approach with neighboring strategy in Algorithm 2 for $\beta = 0, 1$; 3) the CGL method; 4) the CCGL method; and 5) the GGM-GLRT method. The last three methods are described at the end of Subsection VI-A.

Figure 5 presents the F-score measure of the different methods versus $\frac{1}{N}$ for $M = 1,000$ time samples and $r = 5$. Figure 6 presents the F-score measure of the different methods versus the number of measurements, $M$, for $\sigma_w^2 = 0.1$ and $r = 5$. It can be seen that the F-score measure of all methods increases as the noise variance, $\sigma_w^2$, decreases and/or where the number of time samples, $M$, increases. In addition, it can be seen that the proposed algorithms outperform the other methods, where the neighboring strategy with $\beta = 1$ is preferred in terms of the trade-off between accuracy (it has almost the same performance as the “full” greedy) and computational complexity (see in Subsection V-C1 and in the simulations below). In addition, the proposed modification of the CGL method, the CCGL method, significantly improves the performance, compared with the original CGL method, for large $M$ and small $\sigma_w^2$. Thus, considering the disconnection constraint in (57) and employing appropriate thresholding improves the CGL method from [34], [36] for the problem of edge disconnections identification. It can be seen that the GGM-GLRT method from [32] is more sensitive to noise and to a small number of measurements than the other methods. This is because the model of the GGM-GLRT method does not consider the influence of the noise in the measurements, and is based on covariance matrix estimation, which requires a large number of measurements. Finally, it can be seen that the neighboring strategy, which was developed under the assumption of a local property of the GMRF filter (in Proposition 2), performs well for the regularized Laplacian filter and the heat diffusion filter in the middle and the right subfigures of Figs. 3-6. That is, the neighboring strategy is robust under the other tested smooth graph filters.

In order to demonstrate the empirical complexity of the identification methods for different network sizes, the average computation time was evaluated by running these methods using Matlab on an Intel(R) Core(TM) i7-9700X CPU @ 3.30GHz. Figure 7 shows the run-time of the different methods versus the number of vertices in the graph, $N$, for $\sigma_w^2 = 0.1$, $M = 100,000$, $r = 2$, and $L = 100$ Monte-Carlo simulations. It can be seen that the run-time of all methods increases as $N$ increases. For large values of $N$, the “full” greedy approach from Algorithm 1 has the largest run-time. At the same time, the neighboring strategy efficiently reduces the run-time without a significant performance loss, as shown in the previous figures. Moreover, for large $N$, the neighboring strategy obtains similar complexity to the CGL methods and the GGM-GLRT method. The CCGL method was implemented by ‘quadprog’ solver in Matlab and its run-time can be reduced by implementing more efficient solutions, as proposed for the CGL in [34], [56].

D. Identifying outages in power system dataset

In this subsection, we evaluate the performance of the proposed methods for identifying line outages in electrical networks using phasor angle measurements obtained by phasor...
measurement units (PMUs) [7]–[10]. A power system can be represented as an undirected weighted graph, \( G = (V, E, W) \), where the vertices in \( V \) denote the buses (generators or loads), and the edges in \( E \) denote transmission lines between these buses. The weighted adjacency matrix, \( W \), is determined by the branch susceptances (as described, for example, in [22], [46]). We assume that we have PMUs in the considered system that acquire noisy measurements of the voltage phasors (amplitudes and phases) at all buses. In addition, we assume that the voltage amplitudes equal one in a per unit (p.u.) system, which is a common assumption in power systems [63]. Since the voltage phases are measured over the buses of the electrical network (vertices in the graph representation), they can be considered as graph signals. It has recently been shown in [46]–[48] that the voltages in power systems can be considered as smooth graph signals w.r.t. the associated Laplacian matrix. In order to obtain the voltage dataset, we execute an optimal power flow solution obtained by MATPOWER [64] over the power demand data embedded in Gaussian noise. The simulations were implemented on the IEEE 118-bus test case, which is presented in Fig. 8 as a one-line diagram (left) and as a graphical representation of this grid (right).

We tested four different outages at the transmission lines (disconnections of edges in the graph representation): \{ (23, 24), (29, 31), (52, 53), (80, 97) \}. Then, we uniformly randomized one of the combinations of these outages such that
Fig. 7: Run-time of the methods: greedy approaches from Algorithm 1 and Algorithm 2 for \( \beta = 0, 1 \), CGL method, CCGL method, and the GGM-GLRT method for \( \sigma^2_w = 0.1 \), \( M = 100,000 \), \( r = 2 \), and \( L = 100 \).

\( r \leq 4 \) and \( r_{\text{max}} = 10 \). We implemented the proposed greedy algorithms assuming a GMRF filter, which is a mismatch in the distribution of the signals.

Figure 9 presents the ROC of the detector in (52), where \( E \) is obtained from the “full” greedy algorithm or by the greedy algorithm with the neighboring strategy in Algorithm 2 with \( \beta = 0, 1 \), for a noise variance of \( \sigma^2_w = 0.2 \), and \( M = 1,000 \) time samples; results from the naive smoothness detector from (51) and with the BMDS detector from (53) are also shown. It can be seen that the “full” greedy algorithm outperforms the other detectors for any probability of false alarm. For the greedy algorithm with neighboring strategy, the probability of detection increases as \( \beta \) increases. The naive smoothness detector and the BMDS detector have a higher probability of detection than the \( \beta = 0, 1 \) greedy method for large probabilities of false alarm, which cannot be tolerated in practical systems.

Figure 10 presents the F-score measure of the different methods versus \( \frac{1}{w} \) for \( M = 10,000 \) time samples. The CGL method does not converge on this dataset, and therefore is not presented in this figure. The simulation results show that the proposed methods can be used for this practical scenario, while the GGM-GLRT has poor performance in this case. In addition, the CCGL method performs well only for a large enough number of time samples (for example, for \( M = 1,000 \), the CCGL method does not converge).

VII. CONCLUSION

In this paper we investigate the problem of identifying edge disconnections in networks based on a graph filter output model, where the initial topology of the graph is known. We show that the LRT for detecting the status (connected/disconnected) of a specific set of edges can be interpreted as a local smoothness detector for the noiseless GMRF filter with a Laplacian precision matrix. For the general edge disconnections identification problem, we show that the ML decision rule consists of parallel LRTs of the binary hypothesis testing problems of each edge set. The interpretation of the LRT and the ML decision rule in the graph spectral domain is demonstrated. For nested subsets of edge disconnections and under mild conditions, the ML decision rule imposes a larger penalty on models with a larger number of edge disconnections, which can be interpreted as a way to avoid overfitting. In addition, it is shown that the sufficient statistics of the LRT and the ML decision rule are the graph energy levels under each candidate graph. Thus, there is no need for an accurate estimation of the sample covariance matrix. Since the ML decision rule requires an exhaustive search and is impractical for large networks, we propose two greedy algorithms for its low-complexity implementation. The greedy solutions are iterative methods that are based on gaining a series of single-edge disconnection updates and on local properties of the smooth graph filters, by taking into account only the neighbors of the suspicious edges.

Our simulations demonstrate that the proposed methods outperform existing methods on the tested scenarios in terms of detection and identification performance, as well as computational complexity. Thus, the proposed greedy methods provide a good trade-off between performance and complexity and are practical for large-scale networks. In addition, the simulations show the robustness of the proposed neighboring strategy, which is based on the local property of the GMRF filter, for other smooth graph filter models. We also demonstrate the use of the proposed methods for detecting power system line outages by using PMU phasor angle measurements. Future research directions include extension of the proposed methods for the case of disconnection identification with unknown graph filters, identification of edge disconnections in dynamic models, and detection of new connections in the networks. A deeper analysis of the proposed framework should be conducted, including determining the required number of measurements for sufficient performance and analyzing the robustness of the results to the underlying assumptions.

APPENDIX A: PROOF OF PROPOSITION 1

We begin this appendix by proving that under the conditions of Proposition 1 the eigenvalues of the considered Laplacian matrices satisfy

\[ \lambda_n(L^{(k_1)}) \leq \lambda_n(L^{(k_2)}), \quad n = 1, \ldots, N, \]  

(60)

where \( \lambda_n(L^{(k_1)}) \) and \( \lambda_n(L^{(k_2)}) \) are the \( n \)th eigenvalues of \( L^{(k_1)} \) and \( L^{(k_2)} \), respectively. By substituting the two Laplacian matrices in (18)–(19), one obtains

\[ L^{(k_2)} = L^{(0)} - \sum_{(i,j) \in C^{(k_1)}} E^{(i,j)}, \quad t = 1, 2. \]  

(61)

Under Condition C.2 of Proposition 1 \( C^{(k_2)} \) is a proper subset of \( C^{(k_1)} \). Thus, (61) implies that

\[ L^{(k_2)} = L^{(k_1)} + \sum_{(i,j) \in C^{(k_1)} \setminus C^{(k_2)}} E^{(i,j)}. \]  

(62)

Since \( L^{(k_1)}, L^{(k_2)} \), and \( \sum_{(i,j) \in C^{(k_1)} \setminus C^{(k_2)}} E^{(i,j)} \) are Hermitian matrices, by using the Weyl's inequality (see, e.g. Chapter 4.3 in [65]), one obtains

\[ \lambda_n(L^{(k_1)}) + \tilde{\lambda}_1^E \leq \lambda_n(L^{(k_2)}), \quad 1 \leq n \leq N, \]  

(63)

where \( \tilde{\lambda}_1^E \) is the smallest eigenvalue of \( \sum_{(i,j) \in C^{(k_1)} \setminus C^{(k_2)}} E^{(i,j)} \), which equals zero, since this
C.3 of Proposition 1), the order relation in (60) implies that \( \lambda \) is a monotonic decreasing function for \( \sigma \) since \( L \) is a singular PSD matrix. Therefore, by substituting \( \lambda^2 = 0 \) in (63), we obtain (60). It should be noted that a positive, except for \( \lambda = 1 \).

Since \( L^{(k_1)} \) and \( L^{(k_2)} \) represent connected graphs, the eigenvalues of these Laplacian matrices are positive, except for \( \lambda_1^{(k_2)} = 0 \). Thus, since the graph filter \( h(\lambda) \) is a monotonic decreasing function for \( \lambda \) for any \( \lambda > 0 \) (Condition C.3 of Proposition 1), the order relation in (60) implies that

\[
\sigma^2 h^2(\lambda_n^{(L^{(k_1)})}) \geq \sigma^2 h^2(\lambda_n^{(L^{(k_2)})}), \quad 1 \leq n \leq N,
\]

where we use the fact that \( \sigma^2 > 0 \). By substituting \( k = k_1, k = k_2 \) in (64), the inequality in (65) implies that

\[
\rho(L^{(k)}) = \sum_{n=1}^{N} \log \left( \frac{\sigma^2 h^2(\lambda_n^{(L^{(k)})})}{\sigma^2 h^2(\lambda_n^{(L^{(0)})}) + \sigma_w^2} \right) \\
\leq \sum_{n=1}^{N} \log \left( \frac{\sigma^2 h^2(\lambda_n^{(L^{(k_1)})}) + \sigma_w^2}{\sigma^2 h^2(\lambda_n^{(L^{(0)})}) + \sigma_w^2} \right) = \rho(L^{(k_1)}), \quad (66)
\]

which implies (60) and completes the proof of Proposition 1.

**APPENDIX B: DEVELOPMENT OF \( \rho(L^{(k)}) \) FOR THE NOISELESS GMRF FILTER**

In this appendix, we prove that \( \rho(L^{(k)}) \) from (24) for the noiseless GMRF filter in (8) is only a function of the second-order statistics of the vertices in \( S^{(k)} \). By substituting (8) and \( \sigma_w^2 = 0 \) in (24), we obtain

\[
\rho(L^{(k)}) = \log \left( \frac{\sigma^2 h(\lambda^{(L^{(k)})})}{\sigma^2 h(\lambda^{(L^{(0)})})} \right) = \log \left( \frac{|L^{(0)}|}{|L^{(k)}|} \right), \quad (67)
\]
where the last equality stems from using the definition of the pseudo-determinant of a pseudo-inverse matrix and (18). According to Theorem 4 in [67], the Laplacian matrix of a connected graph satisfies
\[ |L^{(k)}| = \prod_{n=2}^{N} \lambda_n^{L^{(k)}} = |L^{(k)}|, \]  
where
\[ L^{(k)} = L^{(k)} + \frac{1}{N} I 1^T \]  
and \( \lambda_1^{L^{(k)}}, \ldots, \lambda_N^{L^{(k)}} \) are the eigenvalues of \( L^{(k)} \). By substituting (68) in (67), we obtain that in this case
\[ \rho(L^{(k)}) = \log \left( \frac{|L^{(0)}|}{|L^{(k)}|} \right). \]  
In addition, according to Theorem 5 in [67], the inverse of the matrix in (69), satisfies
\[ (L^{(k)})^{-1} = (L^{(k)})^T + \frac{1}{N} 1 1^T = \frac{1}{\sigma_x^2} \Sigma(L^{(k)}) + \frac{1}{N} 1 1^T , \]  
where the last equality is obtained by substituting (16), which holds under the GMRF model.

Without loss of generality, we assume that the non-zero elements in the matrix \( E^{(k)} \) appear in the upper square block of \( E^{(k)} \), i.e. the Laplacian matrix after the disconnections in \( S^{(k)} \) can be written as
\[ L^{(k)} = \begin{bmatrix} L^{(0)}_{S^{(k)}} - E^{(k)}_{S^{(k)}} & L^{(0)}_{S^{(k)}, S^{(k)}} \\ L^{(0)}_{S^{(k)}, S^{(k)}} & L^{(0)}_{S^{(k)}} \end{bmatrix}, \]  
where \( S^{(k)} = V \setminus S^{(k)} \). Similarly, according to (69), we can write
\[ L^{(k)} = \begin{bmatrix} L^{(0)}_{S^{(k)}} - E^{(k)}_{S^{(k)}} & L^{(0)}_{S^{(k)}, S^{(k)}} \\ L^{(0)}_{S^{(k)}, S^{(k)}} & L^{(0)}_{S^{(k)}} \end{bmatrix}, \]  
where \( L^{(0)} = L^{(k)} + \frac{1}{N} 1 1^T \). By applying the block matrix determinant rule [53] on the matrix in (73), one obtains
\[ \begin{vmatrix} L^{(k)} \end{vmatrix} = \begin{vmatrix} L^{(0)}_{S^{(k)}} \end{vmatrix} \times \begin{vmatrix} L^{(0)}_{S^{(k)}} - E^{(k)}_{S^{(k)}} - L^{(0)}_{S^{(k)}, S^{(k)}} (L^{(0)}_{S^{(k)}, S^{(k)}})^{-1} L^{(0)}_{S^{(k)}, S^{(k)}} \end{vmatrix}. \]  
Then, by using the partitioned matrix inversion lemma (see, e.g. Eq. (8) in [53]) on (73), we obtain that the upper block of the inverse matrix of \( L^{(k)} \) satisfies
\[ \begin{vmatrix} (L^{(k)})^{-1} \end{vmatrix}_{S^{(k)}} = \begin{vmatrix} (L^{(0)}_{S^{(k)}} - E^{(k)}_{S^{(k)})} - L^{(0)}_{S^{(k)}, S^{(k)}} (L^{(0)}_{S^{(k)}, S^{(k)})^{-1} L^{(0)}_{S^{(k)}, S^{(k)}}) \end{vmatrix}^{-1}. \]  
By substituting (75) in (74), one obtains
\[ \begin{vmatrix} L^{(k)} \end{vmatrix} = \begin{vmatrix} L^{(0)}_{S^{(k)}} \end{vmatrix} \left( \begin{vmatrix} (L^{(k)})^{-1} \end{vmatrix}_{S^{(k)}} \right)^{-1} = \begin{vmatrix} L^{(0)}_{S^{(k)}} \end{vmatrix} \left( \frac{1}{\sigma_x^2} \Sigma(L^{(k)})_{S^{(k)}} + \frac{1}{N} 1 1^T \right)^{-1}, \]  
where the last equality is obtained by using (71). In a similar manner to the development of (76), it can be shown that the determinant of \( L^{(0)} \) satisfies
\[ \begin{vmatrix} L^{(0)} \end{vmatrix} = \begin{vmatrix} L^{(0)}_{S^{(k)}} \end{vmatrix} \left( \frac{1}{\sigma_x^2} \Sigma(L^{(0)})_{S^{(k)}} + \frac{1}{N} 1 1^T \right)^{-1}. \]  
By substituting (76) and (77) in (70), we obtain
\[ \rho(L^{(k)}) = \log \left( \frac{|L^{(0)}_{S^{(k)}}||\Sigma(L^{(0)})_{S^{(k)}} + 1 1^T |^{-1}}{|\Sigma(L^{(k)})_{S^{(k)}} + 1 1^T |^{-1}} \right). \]  
That is, we obtained that \( \rho(L^{(k)}) \) for the noiseless GMRF filter is only a function of the second-order statistics of the vertices in \( S^{(k)} \), as required.

REFERENCES
[1] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3042–3054, 2014.
[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83–98, 2013.
[3] A. Ortega, P. Frossard, J. Kovacević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” Proceedings of the IEEE, vol. 106, no. 5, pp. 808–828, 2018.
[4] L. Stankovic, M. Dakovic, and E. Sejdic, “Vertex-frequency analysis: A way to localize graph spectral components [lecture notes],” IEEE Signal Process. Mag., vol. 34, no. 4, pp. 176–182, 2017.
[5] T. Rountenberg, “Non-Bayesian estimation framework for signal recovery on graphs,” IEEE Trans. Signal Process., vol. 69, pp. 1169–1184, 2021.
[6] E. Ceci and S. Barbarossa, “Graph signal processing in the presence of topology uncertainties,” IEEE Trans. Signal Process., vol. 68, pp. 1558–1573, 2020.
[7] Y. Zhao, J. Chen, A. Goldsmith, and H. V. Poor, “Identification of outages in power systems with uncertain states and optimal sensor locations,” IEEE Sel. Topics in Signal Process., vol. 8, no. 6, pp. 1140–1153, 2014.
[8] Z. Zhu and G. B. Giannakis, “Sparse overcomplete representations for efficient identification of power line outages,” IEEE Trans. Power Syst., vol. 27, no. 4, pp. 2215–2224, 2012.
[9] F. F. Wu and W.-H. Liu, “Detection of topology errors by state estimation,” IEEE Trans. Power Syst., vol. 4, no. 1, pp. 176–183, 1989.
[10] J. E. Tate and T. J. Overbye, “Line outage detection using phasor angle measurements,” IEEE Trans. Power Systems, vol. 23, no. 4, pp. 1644–1652, 2008.
[11] M.-P. Kwan and D. M. Ransberger, “LiDAR assisted emergency response: Detection of transport network obstructions caused by major disasters,” Computers, Environment and Urban Syst., vol. 34, no. 3, pp. 179–188, 2010.
[12] J. Ghosh, H. Q. Ngo, S. Yoon, and C. Qiao, “On a routing problem within probabilistic graphs and its application to intermittently connected networks,” in Conf. on Computer Communications (INFOCOM), 2007, pp. 1721–1729.
[13] A. P. Dempster, “Covariance selection,” Biometrics, pp. 157–175, 1972.
[14] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” Biostatistics, vol. 9, no. 3, pp. 432–441, 12 2007.
[15] A. Wiesel and A. O. Hero, “Distributed covariance estimation in Gaussian graphical models,” IEEE Trans. Signal Process., vol. 60, no. 1, pp. 211–220, 2012.
[16] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, “Network topology inference from spectral templates,” IEEE Trans. Signal and Inf. Process. over Networks, vol. 3, no. 3, pp. 467–483, 2017.
[17] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” IEEE Trans. Signal and Inf. Process. over Networks, vol. 4, no. 3, pp. 481–496, 2018.
