Online Graph Learning Via Proximal Newton Method From Streaming Data

Zu-Yu Wu [1], Carrson Fung [2], Jun-Yi Chang [1], Hsin Chuang [1], and Yi-Chen Lin [1]

[1]Affiliation not available
[2]National Yang Ming Chiao Tung University

October 31, 2023

Abstract

Learning graph topology online with dynamic dependencies is a challenging problem. Most existing techniques usually assume the generative model to be a diffusion process instigated by a graph shift operator (GSO) and that a first-order method, such as proximal gradient or least-mean-square (LMS), are used to track the graph topology. However, they are often susceptible to noisy observations and does not perform well against second-order methods. This work proposed two forward-backward splitting algorithms called the proximal Newton-iterated extended Kalman filter (PN-IEKF) and PN-IEKF-vector autoregressive (PN-IEKF-VAR) algorithms to track non-causal and causal graph topology with dynamic dependencies, respectively. The proposed methods directly maximize the posterior probability distribution of the observable graph signal and graph matrix, which make our PN-IEKF framework to be more robust toward additive white Gaussian noise. The two methods can directly handle streaming data which process them as they become available. Effectiveness of the proposed methods can be further improved by including a T-squared detector in the tracking procedure, which helps to inject proper perturbation to the latent dynamic model such that the time-varying nonstationary graph can be reacquired faster amid abrupt changes in the underlying system. Results on relative error and normalized mean square error using synthetic data on Erd\fH{o}s-R\'enyi graph establish the efficacy of the proposed approach. Simulation results using data from the Dataset for Emotion Analysis Using EEG, Physiological and Video Signals (DEAP) and National Oceanic and Atmospheric Administration (NOAA) are encouraging. Computational and time complexity analysis of the proposed algorithm are given and compared with other algorithms.

Online Graph Learning Via Proximal Newton Method From Streaming Data

Zu-Yu Wu, Carrson C. Fung, *Member, IEEE*, Jun-Yi Chang, Hsin Chuang and Yi-Chen Lin*
Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan 300
*National Taiwan University
email: *j6yj3m4@gmail.com, c.fung@ieee.org, developing90@gmail.com, hsin.chuang.ee10@nycu.edu.tw, ryrylin@gmail.com*

***Abstract*—Learning graph topology online with dynamic dependencies is a challenging problem. Most existing techniques usually assume the generative model to be a diffusion process instigated by a graph shift operator (GSO) and that a first-order method, such as proximal gradient or least-mean-square (LMS), are used to track the graph topology. However, they are often susceptible to noisy observations and does not perform well against second-order methods. This work proposed two forward-backward splitting algorithms called the proximal Newton-iterated extended Kalman filter (PN-IEKF) and PN-IEKF-vector autoregressive (PN-IEKF-VAR) algorithms to track non-causal and causal graph topology with dynamic dependencies, respectively. The proposed methods directly maximize the posterior probability distribution of the observable graph signal and graph matrix, which make our PN-IEKF framework to be more robust toward additive white Gaussian noise. The two methods can directly handle streaming data which process them as they become available. Effectiveness of the proposed methods can be further improved by including a T-squared detector in the tracking procedure, which helps to inject proper perturbation to the latent dynamic model such that the time-varying nonstationary graph can be reacquired faster amid abrupt changes in the underlying system. Results on relative error and normalized mean square error using synthetic data on Erdős-Rényi graph establish the efficacy of the proposed approach. Simulation results using data from the Dataset for Emotion Analysis Using EEG, Physiological and Video Signals (DEAP) and National Oceanic and Atmospheric Administration (NOAA) are encouraging. Computational and time complexity analysis of the proposed algorithm are given and compared with other algorithms.**

***Index Terms*—Online graph learning, graph signal processing, forward-backward (FB) splitting algorithm, proximal Newton method, iterated extended Kalman filter**

I. Introduction

Graph is a powerful form of representation that can model pairwise relationship between non-Euclidean data residing on irregular structures. Graph signal processing (GSP) extends traditional signal processing operations such as delay, convolution and Fourier transform to graph domain, such that analysis and processing of data on graph can be performed as easily as in the time or spatial domain. Unlike these traditional domains, where the data are the only entities that carry information, graph is a natural data structure to capture interactions amid signals. In real world applications, the latent graph topology is not provided in advance, which motivates the development of graph learning techniques that acquire the graph topology from observed signals. Approaches stem from statistics, physics and GSP perspective have been investigated to learn the underlying graph [1], [2]. To illustrate, local area wireless networks can be modeled as undirected graphs, where the edge weights are inversely proportional to the physical distance between the transmitter and receivers, and the the graph signals could be signal strength collected at different measurement points in the area [3]. The interpretability of estimated graph depends on model selection and applications. For instance, different choices of parcellation of brain regions or different modalities of brain signals will result in different brain graphs [4], [5]. In most cases, learning the hidden structure is an ill-posed problem and requires prior knowledge before the problem can be solved.

GSP based graph learning methods has garnered growing attention in the signal processing community. By introducing the graph shift operator (GSO) that defines traditional shifting operation in the graph domain, graph signals priors can be easily included in the formulation of graph learning problem [6], [7]. Many graph learning problems assume the GSO is involved in the signal model. [8] introduced the causal graph process which is based on the matrix polynomial of a GSO. Other approaches, such as [9], attempts to recover the GSO that contains information about the relationship between graph signals from observations generated by the diffusion process instigated by the GSO. Other studies have focused on solving the graph learning problem by considering the graph Laplacian matrix, which is one of the GSOs widely used to define the graph spectral domain, from which the notion of signal smoothness on the graph is established [10]–[12]. [10] directly dealt with the graph adjacency matrix and applied a logarithmic barrier to node degree to prevent isolated nodes. [11], [12] revealed the equivalence between imposing the signal smoothness assumption in graph learning problem and treating the Laplacian matrix as a class of precision matrices in a Gaussian-Markov random field (GMRF) [13], which allows the problem to be addressed in a statistical viewpoint. The precision matrix of a GMRF has drawn con-

All authors are affiliated with the Institute of Electronics at the National Yang Ming Chiao Tung University. Yi-Chen Lin is now with the National Taiwan University.
This work has been partially supported by the Ministry of Science and Technology Grant 110-2221-E-A49-051 and the Google exploreCSR grant 111Q90004C.

siderable attention from researchers in recent decades [14]–[16] as its nonzero off-diagonal entries encode the pairwise conditional dependency between two nodes in the GMRF. [14], [15] formulated the graphical LASSO problem to obtain a sparse representation of the GMRF. In [16], specialized learning algorithms for three types of Laplacian matrices, generalized graph Laplacians (GGLs), diagonally dominant generalized graph Laplacians (DDGLs), and combinatorial graph Laplacians (CGLs), which characterize different classes of GMRF models, were developed.

Using batch learning approaches require data at all time to be prepared before performing the learning process [17], [18], which leads to issues such as potential latency, large amount of computations and storage that can be impractical when nodal dependencies are dynamic. To circumvent these problems, it is desirable to develop online graph learning methods whose performance is comparable to the batch learning approaches but is more adaptive to a dynamically evolving graph network. The least-mean-square (LMS) algorithm was used in [19] to deal with the time-varying graph learning problem, which utilized the heat diffusion model in [20] to model the evolution of graph signals. The authors in [21] assumed the input signal to the generative model to be white and that the output signal is the result of a diffusion process in the GSO so that it is stationary. Then a two-step process is used to first perform a rank-one update on the covariance matrix of the output after collecting a number of streaming observations followed by an update of its eigenvalues using the Newton method [22] and eigenvectors, which are equal to those of the GSO. Finally the GSO is updated using the Alternating Direction Method of Multipliers (ADMM) method, which allows the algorithm to incorporate a number of *a priori* knowledge about the GSO. Even though [23] extended the work in [21] for non-stationary observations, both works are only applicable to symmetric GSO. In addition, the requirement of computing the eigendecomposition of the covariance matrix implies the technique requires samples to be buffered, thus it cannot handle pure streaming data. Instead of tracking the eigenvectors shared by the GSO and the observation covariance matrix, [24] exploited the commutative property of the GSO and the observation covariance matrix and track the GSO by the online proximal gradient (PG) method. Reproducing kernel dictionary was used to model the nonlinear relationship between vertices in [25], in which the stochastic group zero-attracting LMS (GZA-LMS) [26] was applied for online updating of the dictionary. [27] utilized the smoothness assumption of graph signals in [10] and tackled the graph learning problem in time-varying scenario by using the online proximal gradient method, and further performed signal classification according to the smoothness property on different graphs by first learning multiple graphs using the same online PG method [28]. Similarly, extending the work in [29], the online proximal gradient method was used to optimize a memory-aware loss function inspired by the vector autoregressive (VAR) process in [30]. In [31], [32], the PG method was embedded in a prediction-correction method to learn the graph topology lying inside a Gaussian graph model and the structural equation model separately. Recently, [33] proposed a general framework for online graph learning that subsume other online graph learning models, such as Gaussian graphical model, structural equation model and smoothness-based model, by proposing a algorithmic framework that makes their method to be model independent. Unfortunately, the authors did not benchmark their work with others, such as [14] or [19], and did not contain discussion about tracking causality graph. In addition to gradient based methods, Kalman filter has also been used to simultaneously track the graph signals and learn the graph when the graph signals follow an autoregressive process. Using Kalman filter, [34] and [35] dealt with the signal tracking and graph learning problems in two separate procedures. Specifically, [35] solved the problem in [8] online. Unfortunately, there were unresolved errors in the results and missing parameter values prevented resimulation of the algorithm.

Most of the above mentioned online learning methods are based on the first-order method, which has advantage in terms of low computational complexity compared to second-order methods, such as one proposed herein. However, as will be shown in the simulation results below, the proposed method improves upon accuracy of the estimated graph. A novel proximal Newton-iterated extended Kalman filter (PN-IEKF) is proposed for online learning of undirected graphs that is based on the IEKF proposed in [36], [37]. The PN-IEKF can achieve better estimation accuracy with the same number of streaming time sample compared to the first-order methods. The underlying diffusion model is also replaced by the VAR model to create the PN-IEKF-VAR method that track Granger causlity in directed graphs. The organization of the paper is described as follows.

- The detail of the system model inspired by the heat diffusion and the VAR process, together with the assumptions made in the signal models, are given in Sec. II.
- The proposed PN-IEKF and PN-IEKF-VAR methods are described in Sec. III.
- The simulation setup and simulation results are shown in Sec. IV.
- Sec. V gives a summary of this work.
- Appendix A provides the detail derivation of the IEKF procedures contained in the proposed PN-IEKF and the PN-IEKF-VAR methods.

The main contributions of this work are

1) A novel second-order online graph learning method using proximal Newton method is proposed that outperforms its first-order counterparts for undirected and directed graphs.
2) To the best of our knowledge, this is the first work that introduces the scaled proximal operator into the iterated extended Kalman filter framework.
3) Faster convergence speed is obtained using the second-order method compared to prior works using first-order methods.
4) Accuracy of proposed second-order method is less sensitive to additive noise compared to prior works using first-order approaches.
5) Relative error and normalized mean square error results

are shown to show the efficacy of the proposed approach.
6) Computational complexity analysis and time complexity are also included for the proposed methods.
7) A brief discussion about the convergence of the proposed algorithms is given.

Notations: Uppercase (lowercase) bold face letters indicate matrices (column vectors). Superscript ^{T} denotes transposition. $\mathbf{A}^2$ is defined as $\mathbf{A}\mathbf{A}$. $[\mathbf{A}]_{i,j}$ denotes the $(i,j)^{th}$ element of $\mathbf{A}$. $\mathrm{diag}(\mathbf{A})$ creates a column vector using the diagonal elements of $\mathbf{A}$. $\mathrm{Diag}(\mathbf{a})$ creates an $N \times N$ diagonal matrix using the elements of $\mathbf{a} \in \mathbb{R}^N$. $|a|$ denotes the magnitude of a. $E[\cdot]$ stands for statistical expectation of the entity inside the square bracket. $\mathbb{S}^M$ denotes the set of real-value symmetric matrices. $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ denotes Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$. $\mathbf{A} \succeq \mathbf{0}_{N\times N}$ designates $\mathbf{A}$ as an $N \times N$ symmetric positive semidefinite matrix. $\mathbf{I}_N$ denotes an $N \times N$ identity matrix. $\mathbf{0}_N$, $\mathbf{1}_N$ and $\mathbf{I}_N$ denote a column vector with N zeros, N ones and $N \times N$ identity matrix, respectively. $tr(\cdot)$ denotes the trace of the matrix. $\mathrm{Tri}(\mathbf{A})$ extracts the upper triangular portion of $\mathbf{A}$. $vec(\mathbf{A})$ denotes matrix vectorization where the columns of $\mathbf{A}$ are stacked together into a vector. $\mathbf{A}^\dagger$ denotes the pseudoinverse of $\mathbf{A}$. $\|\mathbf{A}\|_1$ and $\|\cdot\|^2_{\mathbf{A}}$, respectively, represent the ℓ_1 norm of $\mathbf{A}$ and the quadratic norm with respect to $\mathbf{A} \succ \mathbf{0}$. $D\boldsymbol{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$ is the Jacobian of $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ with $\mathbf{x} \in \mathbb{R}^n$. $\nabla f(\mathbf{x}) \in \mathbb{R}^n$ is the gradient vector of f with respect to $\mathbf{x}$. $e^{\mathbf{A}}$ is the matrix exponential of $\mathbf{A}$. $Pr(\cdot)$ is the probability of an event. All the definition of symbols associated with the proposed PN-IEKF algorithm used in Secs. II-A and III-A are given in Table I, while the definition of symbols associated with the proposed PN-IEKF-VAR algorithm used in Secs. II-B and III-B are given in Table II.

Symbol	Definition
$\mathcal{G}$	Weighted undirected connected graph without self-loops.
$\mathcal{V}$	Vertex set of $\mathcal{G}$.
$\mathcal{E}$	Edge set of $\mathcal{G}$.
M	Number of nodes.
$\mathbf{W}$	Adjacency matrix of $\mathcal{G}$.
$\mathbf{D}$	Degree matrix of $\mathcal{G}$.
$\mathbf{L}$	Combinatorial graph Laplacian matrix of $\mathcal{G}$.
$\mathbf{s}(t)$	Graph signals from the diffusion process in continuous time.
$\mathbf{u}(t)$	Process noise of the diffusion process in continuous time.
T	Sampling period.
n	Discrete time index.
$\mathbf{s}[n]$	Graph signals from the diffusion process in discrete time.
$\mathbf{u}[n]$	Process noise of the diffusion process in discrete time.
$\mathbf{A}$	Transition matrix of the AR(1) process.
$\mathbf{V}_L$	Orthonormal eigenvector matrix of $\mathbf{L}$.
$\mathbf{\Lambda}_L$	Diagonal eigenvalue matrix of $\mathbf{L}$.
$\bar{\mathbf{V}}_L$	The matrix containing eigenvectors in $\mathbf{V}_L$ associated with positive eigenvalues.
$\bar{\mathbf{\Lambda}}_L$	The diagonal matrix containing positive eigenvalues of $\mathbf{L}$.
$\bar{\mathbf{\Lambda}}_A$	The diagonal matrix containing eigenvalues of $\mathbf{A}$ that are greater than 1, which are also associated with the eigenvectors of $\mathbf{A}$ in $\bar{\mathbf{V}}_L$.
$\bar{\mathbf{A}}[n]$	Transition matrix of the AR(1) process after the centering procedure.
$\bar{\mathbf{s}}[n]$	Graph signals after removing the center value of $\mathbf{s}[n]$.
$\bar{\mathbf{u}}[n]$	Process noise of the AR(1) process after removing the center value of $\mathbf{u}[n]$.
σ_u^2	Variance of each element in $\mathbf{u}[n]$.
$\mathbf{C}_{\bar{u}}$	Covariance matrix of $\bar{\mathbf{u}}[n]$.
$\mathbf{x}[n]$	Noisy observation of graph signals.
$\mathbf{w}[n]$	Additive noise in $\mathbf{x}[n]$.
$\mathbf{C}_w$	Covariance matrix of $\mathbf{w}[n]$.
$\mathbf{s}_W[n]$	State vector of the nonlinear AR(1) process.
$\boldsymbol{\omega}[n]$	Vector that contains the elements in the upper triangular part of the adjacency matrix $\mathbf{W}[n]$.
$P = \frac{1}{2}M^2 + \frac{M}{2}$	Number of elements in the upper triangular part of $\mathbf{W}[n]$, which is the size of $\boldsymbol{\omega}[n]$.
$\mathbf{u}_\omega[n]$	Process noise of $\boldsymbol{\omega}[n]$.
$\boldsymbol{f}(\cdot)$	Nonlinear function of the dynamic model of the augmented AR(1) process.
$\mathbf{u}_W[n]$	Process noise of $\mathbf{s}_W[n]$.
$\mathbf{C}_{u_W}$	Covariance matrix of $\mathbf{u}_W[n]$.
$\mathbf{H}$	Observation matrix in the observation model.
$\widehat{\mathbf{s}}_W[n]$	Estimate of $\mathbf{s}_W[n]$.
$\mathbf{x}_0^n$	Set containing all the observation from 0 to n.
$\widehat{\mathbf{s}}_W[n\|m]$	Estimate of $\mathbf{s}_W[n]$ given $\mathbf{x}_0^m$.
$\mathbf{M}[n\|n-1]$	$E\left[(\boldsymbol{f}(\widehat{\mathbf{s}}_W[n-1\|n]) - \mathbf{s}_W[n]) (\boldsymbol{f}(\widehat{\mathbf{s}}_W[n-1\|n]) - \mathbf{s}_W[n])^T\right]$
$\mathbf{M}[n-1\|n-1]$	$E\left[(\widehat{\mathbf{s}}_W[n-1\|n-1] - \mathbf{s}_W[n-1]) (\widehat{\mathbf{s}}_W[n-1\|n-1] - \mathbf{s}_W[n-1])^T\right]$.
$\ell_1(\mathbf{s}_W[n])$	Differentiable part of the loss function for estimating $\mathbf{s}_W[n]$ given $\mathbf{x}_0^n$.
β	Weighting coefficient of the ℓ_1 norm sparsity penalty function.
i	Iteration index of the PN-IEKF iterations.

$\mathbf{\Phi}$	Hessian of $\ell_1(\mathbf{s}_W[n])$.

TABLE I: Table of notations for the PN-IEKF algorithm.

Symbol	Definition
K	Order of the VAR model.
M	Number of nodes.
$\mathbf{s}[n]$	Graph signal.
$\mathbf{W}_1[n], \ldots, \mathbf{W}_K[n]$	Transition matrices of the VAR model.
$\mathbf{u}[n]$	Process noise of $\mathbf{s}[n]$.
σ_u^2	Variance of each element in $\mathbf{u}[n]$.
M	Number of nodes.
$\mathbf{w}[n]$	Additive noise to the graph signals.
$\mathbf{x}[n]$	Noisy observation of graph signals.
$\bar{\mathbf{s}}[n]$	Concatenated graph signals with $\mathbf{s}[n], \ldots, \mathbf{s}[n-K+1]$ stacked into one column.
$\mathbf{\Theta}$	Transition matrix for the concatenated graph signals $\bar{\mathbf{s}}[n]$
$\mathbf{u}_\theta[n]$	Augmented process noise for the concatenated graph signals $\bar{\mathbf{s}}[n]$.
$J = M^2$	Number of elements in each $\mathbf{W}_k, k = 1, \ldots, K$.
$\bar{\boldsymbol{\omega}}[n]$	Vector concatenating $vec(\mathbf{W}_1[n]), \ldots, vec(\mathbf{W}_K[n])$.
$\mathbf{s}_W[n]$	State vector of the nonlinear AR(1) process.
$\mathbf{u}_\omega[n]$	Process noise of $\bar{\boldsymbol{\omega}}[n]$.
$\boldsymbol{f}(\cdot)$	Nonlinear function of the dynamic model of the augmented AR(1) process.
$\mathbf{u}_W[n]$	Process noise of $\mathbf{s}_W[n]$.
σ_ω^2	Variance of each element in $\mathbf{u}_\omega[n]$.
$\mathbf{C}_{u_W}$	Covariance matrix of $\mathbf{u}_W[n]$.
$\mathbf{x}[n]$	Noisy observation of graph signals.
$\mathbf{w}[n]$	Additive noise to the graph signals.
$\mathbf{C}_w$	Covariance matrix of $\mathbf{w}[n]$.
$\mathbf{H}$	Observation matrix in the observation model.
β	Weighting coefficient of the group sparsity penalty function.
$\mathbf{w}_{p,r}[n]$	Vector collecting all the (p,r) elements from $\mathbf{W}_1[n], \ldots, \mathbf{W}_K[n]$.
$\mathbf{x}_0^n$	Set containing all the observation from 0 to n.
$\ell_1(\mathbf{s}_W[n])$	Differentiable part of the loss function for estimating $\mathbf{s}_W[n]$ given $\mathbf{x}_0^n$.
$\mathbf{\Phi}$	The Hessian of $\ell_1(\mathbf{s}_W[n])$.
i	Iteration index of the PN-IEKF-VAR iterations.
$\widehat{\mathbf{s}}_W[n\|m]$	Estimate of $\mathbf{s}_W[n]$ given $\mathbf{x}_0^m$.
$\mathbf{M}[n\|n-1]$	$E\left[(\boldsymbol{f}(\widehat{\mathbf{s}}_W[n-1\|n]) - \mathbf{s}_W[n]) (\boldsymbol{f}(\widehat{\mathbf{s}}_W[n-1\|n]) - \mathbf{s}_W[n])^T\right]$
$\mathbf{M}[n-1\|n-1]$	$E\left[(\widehat{\mathbf{s}}_W[n-1\|n-1] - \mathbf{s}_W[n-1]) (\widehat{\mathbf{s}}_W[n-1\|n-1] - \mathbf{s}_W[n-1])^T\right]$.

TABLE II: Table of notations for the PN-IEKF-VAR algorithm.

II. System Models

Two different AR processes are discussed in this section. In Sec. II-A, a weighted undirected connected graph without self-loops $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is considered. The mathematical representation of $\mathcal{G}$ with $|\mathcal{V}| = M$ is a square adjacency matrix $\mathbf{W} \in \mathbb{S}^M$ with nonnegative elements and $[\mathbf{W}]_{ij} > 0$ when $(i,j) \in \mathcal{E}$, which implies $[\mathbf{W}]_{ii} = 0,\ i = 1, ..., M$. A diagonal matrix $\mathbf{D} \in \mathbb{R}^{M\times M}$, whose elements are defined as $[\mathbf{D}]_{ii} \triangleq \sum_{i\neq j}[\mathbf{W}]_{ij}$, is the degree matrix, and $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W} \succeq \mathbf{0}_{M\times M}$ is the graph Laplacian.

In Sec. II-B, the K-step VAR model is used to model the multivariate Granger causality between different nodes and time.

The following assumptions are also made in the signal model of both Sec. II-A and Sec. II-B:

Assumption 1. $\mathbf{u}[n] \sim \mathcal{N}(\mathbf{0}_M, \sigma_u^2\mathbf{I}_M)$, $\bar{\mathbf{u}}[n]$, $\mathbf{w}[n] \sim \mathcal{N}(\mathbf{0}_M, \mathbf{C}_w)$, $\mathbf{u}_\omega[n]$, and $\mathbf{u}_W[n]$ are white Gaussian processes. In Sec. II-A, $\bar{\mathbf{u}}[n] \sim \mathcal{N}(\mathbf{0}_M, \mathbf{C}_{\bar{u}})$, $\mathbf{u}_\omega[n] \sim \mathcal{N}(\mathbf{0}_P, \sigma_\omega^2\mathbf{I}_P)$, and $\mathbf{u}_W[n] \sim \mathcal{N}(\mathbf{0}_{M+P}, \mathbf{C}_{u_W})$. In Sec. II-B, $\mathbf{u}_\omega[n] \sim \mathcal{N}(\mathbf{0}_{KJ}, \sigma_\omega^2\mathbf{I}_{KJ})$ and $\mathbf{u}_W[n] \sim \mathcal{N}(\mathbf{0}_{K(M+J)}, \mathbf{C}_{u_W})$

Assumption 2. $\mathbf{s}_W[-1]$, $\mathbf{u}_W[n]$, and $\mathbf{w}[n]$ are statistically independent for all n.

Assumption 3. $Pr(\mathbf{s}_W[n]|\,\widehat{\mathbf{s}}_W[n-1|n]) \approx \mathcal{N}(\boldsymbol{f}(\widehat{\mathbf{s}}_W[n-1|n]), \mathbf{M}[n|n-1])$, $Pr(\mathbf{s}_W[n-1]|\,\mathbf{x}_0^{n-1}) \approx \mathcal{N}(\widehat{\mathbf{s}}_W[n-1|n-1], \mathbf{M}[n-1|n-1])$.

A. AR Process from Diffusion Model

The heat diffusion process describes the phenomenon in which the influence of hot spots spread over a region. In geometric model like graphs, the graph diffusion operator can be used to model the diffusion process evolving at different nodes. In [20], the heat kernel pagerank is shown as the pagerank of a graph and also satisfies the heat equation,

$$\mathbf{s}'(t) = -\mathbf{L}\mathbf{s}(t) + \mathbf{u}(t),$$

where $\mathbf{s}(t) \in \mathbb{R}^M$ is the observed signals on the graph, $\mathbf{u}(t) \in \mathbb{R}^M$ is the perturbation noise in continuous time. The solution is

$$\mathbf{s}(t) = e^{-t\mathbf{L}}\mathbf{s}(0) + \int_0^t e^{-(t-\lambda)\mathbf{L}}\mathbf{u}(\lambda)d\lambda,$$

where $e^{-t\mathbf{L}}$ is the diffusion operator, also known as the heat kernel. Following [19], let t_0 be some time instant during the process, and let T be the sampling period. Define $t_n \triangleq t_0 + nT$. The recursive equation for adjacent time samples can be written as

$$\mathbf{s}(t_n) = e^{-T\mathbf{L}}\mathbf{s}(t_{n-1}) + \int_{t_n-T}^{t_n} e^{-(t_n-\lambda)\mathbf{L}}\mathbf{u}(\lambda)d\lambda.$$

Also define $\mathbf{s}[n] \triangleq \mathbf{s}(t_0 + nT)$ and $\mathbf{u}[n] \triangleq \int_{t_n-T}^{t_n} e^{-(t_n-\lambda)\mathbf{L}}\mathbf{u}(\lambda)d\lambda$ so that the recursive equation in discrete time becomes

$$\begin{aligned}\mathbf{s}[n] &= e^{-T\mathbf{L}}\mathbf{s}[n-1] + \mathbf{u}[n] \\ &= e^{-T(\mathbf{D}-\mathbf{W})}\mathbf{s}[n-1] + \mathbf{u}[n] \\ &= \mathbf{A}\mathbf{s}[n-1] + \mathbf{u}[n], \end{aligned} \tag{1}$$

where $\mathbf{A} \triangleq e^{-T\mathbf{L}} = e^{-T(\mathbf{D}-\mathbf{W})} \in \mathbb{S}_{++}^M$ and $\mathbf{u}[n]$ is the driving noise.

The system in (1) is critically stable since the maximum magnitude of the eigenvalue of $\mathbf{A}$ equals 1. According to [19], to ensure a strictly stable system in which all the eigenvalues of the transition matrix in the dynamic model have magnitude less than 1, assume $\mathcal{G}$ is a connected graph so that the dimension

of the nullspace of $\mathbf{L}$ equals 1. This implies (1) is critically stable with $\mathbf{A}$ having only one eigenvalue equals to 1 because according to its definition

$$\begin{aligned}\mathbf{A} =&e^{-T\mathbf{L}} = \mathbf{V}_L e^{-T\mathbf{\Lambda}_L}\mathbf{V}_L^T\\ =&\begin{bmatrix} \frac{1}{\sqrt{M}}\mathbf{1}_M & \bar{\mathbf{V}}_L \end{bmatrix}\begin{bmatrix} 1 & \mathbf{0}_{M-1}^T \\ \mathbf{0}_{M-1} & e^{-T\bar{\mathbf{\Lambda}}_L}\end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{M}}\mathbf{1}_M^T \\ \bar{\mathbf{V}}_L^T\end{bmatrix}\\ =&\begin{bmatrix} \frac{1}{\sqrt{M}}\mathbf{1}_M & \bar{\mathbf{V}}_L \end{bmatrix}\begin{bmatrix} 1 & \mathbf{0}_{M-1}^T \\ \mathbf{0}_{M-1} & \bar{\mathbf{\Lambda}}_A\end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{M}}\mathbf{1}_M^T \\ \bar{\mathbf{V}}_L^T\end{bmatrix}\\ =&\frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T + \bar{\mathbf{V}}_L\bar{\mathbf{\Lambda}}_A\bar{\mathbf{V}}_L^T,\end{aligned}$$

where $\mathbf{V}_L\mathbf{\Lambda}_L\mathbf{V}_L^T$ is the eigendecomposition of $\mathbf{L}$, $\bar{\mathbf{V}}_L$ contains orthonormal eigenvectors associated with positive eigenvalues of $\mathbf{L}$ in $\bar{\mathbf{\Lambda}}_L$, and $\mathbf{0}_{(M-1)\times(M-1)} \prec \bar{\mathbf{\Lambda}}_A \prec \mathbf{I}_{M-1}$ contains $M-1$ eigenvalues of $\mathbf{A}$. Let $\bar{\mathbf{A}} \triangleq \mathbf{A} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T \prec \mathbf{I}_M$, $\bar{\mathbf{s}}[n] \triangleq \left(\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T\right)\mathbf{s}[n]$, and $\bar{\mathbf{u}}[n] \triangleq \left(\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T\right)\mathbf{u}[n]$ [19]. After removing the center value of $\mathbf{s}[n]$ and applying the doubly stochastic property of $\mathbf{A}$, the diffusion model becomes

$$\begin{aligned}\bar{\mathbf{s}}[n] &= (\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)\mathbf{s}[n] \\ &= (\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)(\mathbf{A}\mathbf{s}[n-1] + \mathbf{u}[n]) \\ &= \mathbf{A}(\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)^2\mathbf{s}[n-1] + \bar{\mathbf{u}}[n] \\ &= (\mathbf{A} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)\bar{\mathbf{s}}[n-1] + \bar{\mathbf{u}}[n] \\ &= \bar{\mathbf{A}}\bar{\mathbf{s}}[n-1] + \bar{\mathbf{u}}[n],\end{aligned} \tag{2}$$

where $\bar{\mathbf{u}}[n]$ is the process noise. The third equality above is true because $\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$ is idempotent. In the time-varying scenario, where the graph changes over time, the AR process becomes

$$\bar{\mathbf{s}}[n] = \bar{\mathbf{A}}[n]\bar{\mathbf{s}}[n-1] + \bar{\mathbf{u}}[n], \tag{3}$$

and the observed signal is modeled as

$$\mathbf{x}[n] = \bar{\mathbf{s}}[n] + \mathbf{w}[n], \tag{4}$$

where $\mathbf{w}[n]$ denotes the measurement noise that is independent of $\bar{\mathbf{s}}[n]$.

B. VAR Model

Redefine the graph signals to be generated from a K-step AR process

$$\mathbf{s}[n] = \sum_{k=1}^{K}\mathbf{W}_k[n]\mathbf{s}[n-k] + \mathbf{u}[n], \tag{5}$$

where $\mathbf{W}_1[n]$, ..., $\mathbf{W}_K[n] \in \mathbb{R}^{M\times M}$ are the transition matrices, $\mathbf{u}[n] \sim \mathcal{N}(\mathbf{0}_M, \sigma_u^2\mathbf{I}_M)$ is the white Gaussian process noise that is independent of $\mathbf{s}[n-k]$. The observation is

$$\mathbf{x}[n] = \mathbf{s}[n] + \mathbf{w}[n]. \tag{6}$$

In order to adapt the K-taps VAR process into the form of the standard extended Kalman filter, which only has state variables at successive time samples n and $n-1$, further define $\bar{\mathbf{s}}[n-1] \triangleq \left[\mathbf{s}[n-1]^T, \mathbf{s}[n-2]^T, \ldots, \mathbf{s}[n-K]^T\right]^T \in \mathbb{R}^{KM}$, $\mathbf{\Theta} \triangleq \begin{bmatrix} \mathbf{W}_1[n] & \cdots & \mathbf{W}_K[n] \\ \mathbf{I}_{(K-1)M} & & \mathbf{0}_{(K-1)M\times M}\end{bmatrix} \in \mathbb{R}^{KM\times KM}$ so that the AR process is rewritten as

$$\bar{\mathbf{s}}[n] = \mathbf{\Theta}\bar{\mathbf{s}}[n-1] + \mathbf{u}_\theta[n] \in \mathbb{R}^{KM}, \tag{7}$$

where $\mathbf{u}_\theta[n] \triangleq \begin{bmatrix}\mathbf{u}^T[n] & \mathbf{0}_{(K-1)M}^T\end{bmatrix}^T \in \mathbb{R}^{KM}$ is the augmented process noise.

III. Methodology

A. Proposed PN-IEKF Approach

The proposed PN-IEKF approach is designed for the signal model in Sec. II-A and it tracks the augmented state vector

$$\mathbf{s}_W[n] = \begin{bmatrix}\bar{\mathbf{s}}[n] \\ \boldsymbol{\omega}[n]\end{bmatrix} \in \mathbb{R}^{M+P} \tag{8}$$

that contains both the graph signal $\bar{\mathbf{s}}[n]$ and the upper triangular portion of the adjacency matrix $\boldsymbol{\omega}[n] \triangleq vec(\text{Tri}(\mathbf{W}[n])) \in \mathbb{R}^P$. Using (3), the augmented state-space model can be written as

$$\begin{aligned}\mathbf{s}_W[n] &= \begin{bmatrix}\bar{\mathbf{A}}[n]\bar{\mathbf{s}}[n-1] \\ \boldsymbol{\omega}[n-1]\end{bmatrix} + \begin{bmatrix}\bar{\mathbf{u}}[n] \\ \mathbf{u}_\omega[n]\end{bmatrix} \\ &= \boldsymbol{f}(\mathbf{s}_W[n-1]) + \mathbf{u}_W[n],\end{aligned} \tag{9}$$

where $\mathbf{u}_\omega[n]$ and $\mathbf{u}_W[n]$ denote the (input) process noise for $\boldsymbol{\omega}[n]$, which accounts for changes in the graph across time, and the overall process noise for $\mathbf{s}_W[n]$, respectively. The observed signal can be written as

$$\mathbf{x}[n] = \mathbf{H}\mathbf{s}_W[n] + \mathbf{w}[n] = \bar{\mathbf{s}}[n] + \mathbf{w}[n], \tag{10}$$

where $\mathbf{H} = \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_{M\times P}\end{bmatrix}$.

In this work, the goal is to find the optimal $\widehat{\mathbf{s}}_W[n]$ by solving

$$\begin{aligned}&(\widehat{\mathbf{s}}_W[n]^\star, \widehat{\mathbf{s}}_W[n-1]^\star) \\ &= \underset{\mathbf{s}_W[n], \mathbf{s}_W[n-1]}{\arg\max}\ Pr\left(\mathbf{s}_W[n], \mathbf{s}_W[n-1] \middle| \mathbf{x}_0^n\right),\end{aligned} \tag{11}$$

where $\mathbf{x}_0^n \triangleq \{\mathbf{x}[0], \mathbf{x}[1], \ldots, \mathbf{x}[n]\}$ is a sequence of observed signal from time 0 to n, and $\widehat{\mathbf{s}}_W[n]^\star = \begin{bmatrix}\widehat{\bar{\mathbf{s}}}[n]^{\star T} & \widehat{\boldsymbol{\omega}}[n]^{\star T}\end{bmatrix}^T$ is the estimate of $\mathbf{s}_W[n]$. Similar definitions are applied to $\widehat{\bar{\mathbf{s}}}[n]^\star$ and $\widehat{\boldsymbol{\omega}}[n]^\star$. Since

$$\begin{aligned}&Pr\left(\mathbf{s}_W[n], \mathbf{s}_W[n-1] \middle| \mathbf{x}_0^n\right) \\ =&Pr(\mathbf{s}_W[n], \mathbf{s}_W[n-1] | \mathbf{x}_0^{n-1}, \mathbf{x}[n]) \\ =&\frac{Pr(\mathbf{s}_W[n], \mathbf{s}_W[n-1], \mathbf{x}[n], \mathbf{x}_0^{n-1})}{Pr(\mathbf{x}[n]|\mathbf{x}_0^{n-1})Pr(\mathbf{x}_0^{n-1})} \\ =&\frac{Pr(\mathbf{s}_W[n], \mathbf{x}[n] | \mathbf{s}_W[n-1], \mathbf{x}_0^{n-1})Pr(\mathbf{s}_W[n-1], \mathbf{x}_0^{n-1})}{Pr(\mathbf{x}[n]|\mathbf{x}_0^{n-1})Pr(\mathbf{x}_0^{n-1})} \\ =&\frac{Pr(\mathbf{x}[n]|\mathbf{s}_W[n], \mathbf{s}_W[n-1], \mathbf{x}_0^{n-1})Pr(\mathbf{s}_W[n]|\mathbf{s}_W[n-1], \mathbf{x}_0^{n-1})}{Pr(\mathbf{x}[n]|\mathbf{x}_0^{n-1})} \times \\ &\frac{Pr(\mathbf{s}_W[n-1], \mathbf{x}_0^{n-1})}{Pr(\mathbf{x}_0^{n-1})} \\ =&\frac{Pr(\mathbf{x}[n]|\mathbf{s}_W[n])Pr(\mathbf{s}_W[n]|\mathbf{s}_W[n-1])Pr(\mathbf{s}_W[n-1]|\mathbf{x}_0^{n-1})}{Pr(\mathbf{x}[n]|\mathbf{x}_0^{n-1})} \\ \propto&Pr\left(\mathbf{x}[n] \middle| \mathbf{s}_W[n]\right) Pr\left(\mathbf{s}_W[n] \middle| \mathbf{s}_W[n-1]\right) Pr\left(\mathbf{s}_W[n-1] \middle| \mathbf{x}_0^{n-1}\right),\end{aligned}$$

where the last equality is induced by **Assumption 2**, (11) is equivalent to

$$\begin{aligned}&(\widehat{\mathbf{s}}_W[n]^\star, \widehat{\mathbf{s}}_W[n-1]^\star)\\ =&\underset{\mathbf{s}_W[n],\mathbf{s}_W[n-1]}{\arg\max}\ Pr\left(\mathbf{x}[n]|\,\mathbf{s}_W[n]\right) Pr\left(\mathbf{s}_W[n]|\,\mathbf{s}_W[n-1]\right)\times\\ &Pr\left(\mathbf{s}_W[n-1]|\,\mathbf{x}_0^{n-1}\right),\end{aligned} \tag{12}$$

which is nonconvex with respect to the two variables. Therefore, instead of solving problem (12) directly, it is proposed to optimize the variables alternately. Specifically, the PN-IEKF method iteratively solves

$$\begin{aligned}&\widehat{\mathbf{s}}_W[n|n]\\ =&\underset{\mathbf{s}_W[n]}{\arg\max}\ Pr\left(\mathbf{x}[n]|\,\mathbf{s}_W[n]\right) Pr\left(\mathbf{s}_W[n]|\,\widehat{\mathbf{s}}_W[n-1|n]\right)\end{aligned} \tag{13}$$

$$\begin{aligned}&\widehat{\mathbf{s}}_W[n-1|n]\\ =&\underset{\mathbf{s}_W[n-1]}{\arg\max}\ Pr\left(\widehat{\mathbf{s}}_W[n|n]|\,\mathbf{s}_W[n-1]\right) Pr\left(\mathbf{s}_W[n-1]|\,\mathbf{x}_0^{n-1}\right)\end{aligned} \tag{14}$$

alternately. $\widehat{\mathbf{s}}_W[n|m]$ is introduced to denote the estimate of $\mathbf{s}_W[n]$ based on $\mathbf{x}_0^m$. The assumed pdfs in **Assumption 1** and **Assumption 3** allow for the derivation of the IEKF. The logic for assuming the conditional probability distribution in **Assumption 3** to be normal distributed originates from the dynamic state model in (9). It is possible to perform statistical sampling to obtain estimate of the distributions in case the Gaussian assumption is false, which will lead to the particle filtering algorithm. This, however, is outside the scope of this work. (13) and (14) can be written as (15) and (16).

The objective in (15) needs to be adjusted so that the resulting graph maintains a sparse structure with nonnegative edge weights. As a result, an ℓ_1 regularization and an indicator function are included in (15) to reformulate the objective for estimating $\mathbf{s}_W[n]$ so that (15) becomes

$$\min_{\mathbf{s}_W[n]}\ \ell_1\left(\mathbf{s}_W[n]\right) + g(\mathbf{s}_W[n]), \tag{17}$$

where $g(\mathbf{s}_W[n]) = \beta\|\boldsymbol{\omega}[n]\|_1 + i_{\mathbb{R}_+^P}(\boldsymbol{\omega}[n])$ is the nondifferentiable part in (17), and $i_{\mathbb{R}_+^P}(\mathbf{a}) = \begin{cases} 0, & \mathbf{a} \succeq \mathbf{0}_P \\ \infty, & \text{otherwise} \end{cases}$. In general, $g(\mathbf{s}_W[n])$ contains prior knowledge of the graph. For applications requiring the graph signals being smooth on the graph [10], $g(\mathbf{s}_W[n])$ can be modified to account for the global smoothness assumption. As described in the following, the PN-IEKF method finds the minimizer of (16) and (17) through the Gauss-Newton method and the proximal Newton method [38], separately. The latter is a splitting algorithm consisting of iterating between a forward and backward step. Since (17) can be viewed as a sum of a differentiable function and a nonsmooth function, it can be solved using the PN method. In the proposed PN-IEKF method, the differentiable term is solved using the IEKF algorithm, which can be interpreted as a Gauss-Newton algorithm ([39, pp. 349-351], [40]–[42]) and is viewed as the forward step, resulting in $\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n] = \widehat{\mathbf{s}}_W^{(i)}[n|n] - \boldsymbol{\Phi}^{-1}\nabla\ell_1(\widehat{\mathbf{s}}_W^{(i)}[n|n])$, where $\boldsymbol{\Phi}$ is the approximated Hessian matrix of $\ell_1(\mathbf{s}_W[n])$ evaluated at $\widehat{\mathbf{s}}_W^{(i)}[n|n]$, i is the iteration index, and the prime symbol denotes the intermediate estimate before obtaining $\widehat{\mathbf{s}}_W^{(i+1)}[n|n]$. $\ell_1(\mathbf{s}_W[n])$ is minimized by $\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]$ since it is quadratic in $\mathbf{s}_W[n]$. The backward step deals with the nonsmooth term can be expressed using the scaled proximal operator as

$$\begin{aligned}&\widehat{\mathbf{s}}_W^{(i+1)}[n|n]\\ =&prox_g^{\boldsymbol{\Phi}}\left(\widehat{\mathbf{s}}_W^{(i)}[n|n] - \boldsymbol{\Phi}^{-1}\nabla f\left(\widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right)\\ =&\arg\min_{\mathbf{s}_W}\ \frac{1}{2}\left(\mathbf{s}_W - \widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right)^T\boldsymbol{\Phi}\left(\mathbf{s}_W - \widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right)\\ &+ g\left(\mathbf{s}_W\right)\\ =&\arg\min_{\mathbf{s}_W}\ \frac{1}{2}\left\|\mathbf{s}_W - \widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right\|_{\boldsymbol{\Phi}}^2 + g\left(\mathbf{s}_W\right),\end{aligned}$$

where $\boldsymbol{\Phi} \succ \mathbf{0}_{(M+P)\times(M+P)}$ will be determined in the sequel. Different from the proximal gradient method [43] that uses the first-order approximation approach, the proximal Newton method replaces the ℓ_2 norm $\|\cdot\|_2^2$ with the quadratic norm $\|\cdot\|_{\boldsymbol{\Phi}}^2$ to form the second-order approximation of the differentiable function in the objective, as shown in Fig. 1.

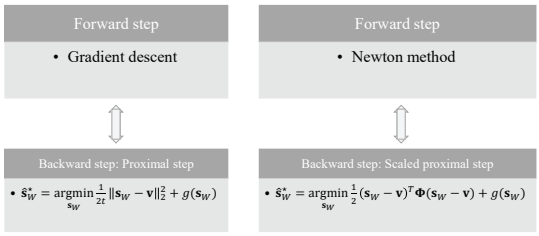


Fig. 1: Comparison between proximal method and proximal Newton method.

Even though (16) is solved inside the PN-IEKF algorithm, the problem is not convex. Furthermore, the Hessian matrices of $\ell_1(\mathbf{s}_W[n])$ and $\ell_2(\mathbf{s}_W[n-1])$ are approximated as $\mathbf{J}_{r_1}^T\mathbf{J}_{r_1} \succeq \mathbf{0}_{(M+P)\times(M+P)}$ and $\mathbf{J}_{r_2}^T\mathbf{J}_{r_2} \succeq \mathbf{0}_{(M+P)\times(M+P)}$ in the Gauss-Newton method, where $\mathbf{J}_{r_1}$ and $\mathbf{J}_{r_2}$ are the Jacobians of $\mathbf{r}_1$ and $\mathbf{r}_2$. As a result, the iterations are not guaranteed to converge because there is no guarantee the minimizer found for (16) will monotonically decrease the objective in (12). Hence, iteration index in the PN-IEKF algorithm i is limited to not exceed a preset number in the simulation. Algorithm 1 shows the pseudocode for the proposed PN-IEKF algorithm, where the derivation is shown in Appendix A. When evaluating $\widehat{\mathbf{s}}_W^{(i)}[n|n-1]$, which is the prediction of $\mathbf{s}_W[n]$ using the estimate of $\mathbf{s}_W[n-1]$, the nonlinear function $\boldsymbol{f}$ is linearized at $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$ and evaluated at $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$. The scaled proximal step is solved in line 10 using the iterative shrinking thresholding algorithm (ISTA) [44]. To reduce the computation complexity of computing the Jacobian of the nonlinear function $\boldsymbol{f}(\mathbf{s}_W[n-1])$ on line 3, the matrix exponential function of $\mathbf{L}$ in $\boldsymbol{f}(\mathbf{s}_W[n-1])$ is replaced by its first-order Taylor series expansion, $\mathbf{I}_M - T\mathbf{L}$. MaxIter (line 13 in Algorithm 1) denotes the maximum number of IEKF iterations being performed.

The process noise of $\boldsymbol{\omega}[n]$ in $\mathbf{s}_W[n]$ is used to model nonstationarity in the graph so that the PN-IEKF approach can adapt more quickly to the scenario in which the graph is chang-

$$\begin{aligned}\widehat{\mathbf{s}}_W[n|n] &= \underset{\mathbf{s}_W[n]}{\arg\min}\ \frac{1}{2}\left(\|\mathbf{x}[n]-\mathbf{H}\mathbf{s}_W[n]\|^2_{\mathbf{C}_w^{-1}} + \|\mathbf{s}_W[n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W[n-1|n]\right)\|^2_{\mathbf{M}^{-1}[n|n-1]}\right)\\ &= \underset{\mathbf{s}_W[n]}{\arg\min}\ \frac{1}{2}\left\| \begin{array}{c} \mathbf{C}_w^{-1/2}\left(\mathbf{x}[n]-\mathbf{H}\mathbf{s}_W[n]\right) \\ \mathbf{M}^{-1/2}[n|n-1]\left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W[n-1|n]\right)-\mathbf{s}_W[n]\right] \end{array}\right\|_2^2 \\ &= \underset{\mathbf{s}_W[n]}{\arg\min}\ \frac{1}{2}\left\|\mathbf{r}_1\left(s_W[n]\right)\right\|_2^2 = \underset{\mathbf{s}_W[n]}{\arg\min}\ \ell_1\left(\mathbf{s}_W[n]\right)\end{aligned} \tag{15}$$

$$\begin{aligned}\widehat{\mathbf{s}}_W[n-1|n] &= \underset{\mathbf{s}_W[n-1]}{\arg\min}\ \frac{1}{2}\left(\|\widehat{\mathbf{s}}_W[n|n]-\boldsymbol{f}\left(\mathbf{s}_W[n-1]\right)\|^2_{\mathbf{C}^{-1}_{u_W}} + \|\mathbf{s}_W[n-1]-\widehat{\mathbf{s}}_W[n-1|n-1]\|^2_{\mathbf{M}^{-1}[n-1|n-1]}\right)\\ &= \underset{\mathbf{s}_W[n-1]}{\arg\min}\ \frac{1}{2}\left\| \begin{array}{c} \mathbf{C}_{u_W}^{-1/2}\left(\widehat{\mathbf{s}}_W[n|n]-\boldsymbol{f}\left(\mathbf{s}_W[n-1]\right)\right) \\ \mathbf{M}^{-1/2}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\mathbf{s}_W[n-1]\right) \end{array}\right\|_2^2 \\ &= \underset{\mathbf{s}_W[n-1]}{\arg\min}\ \frac{1}{2}\left\|\mathbf{r}_2\left(\mathbf{s}_W[n-1]\right)\right\|_2^2 = \underset{\mathbf{s}_W[n-1]}{\arg\min}\ \ell_2\left(\mathbf{s}_W[n-1]\right).\end{aligned} \tag{16}$$

ing (abruptly). However, when the graph remains unchanged, $\mathbf{u}_\omega[n]$ is set to zero, and consequently its covariance matrix as well, in the PN-IEKF algorithm. To determine whether the graph is changing or not, a change detector is embedded in the PN-IEKF to identify the transition in the graph and it is described in greater detail in Sec. III-C.

Algorithm 1: Proposed PN-IEKF based online graph learning algorithm.

Result: $\widehat{\mathbf{s}}_W[n|n]$, $\mathbf{M}[n|n]$
Initialization: $\widehat{\mathbf{s}}_W^{(0)}[n|n]=\widehat{\mathbf{s}}_W[n|n-1]$; $\widehat{\mathbf{s}}_W^{(0)}[n-1|n]=\widehat{\mathbf{s}}_W[n-1|n-1]$; $i=0$;

do
- $previous\_s=\widehat{\mathbf{s}}_W^{(i)}[n|n]$;
- $\mathbf{F}^{(i)}=\left.\frac{\partial \boldsymbol{f}(\mathbf{s}_W[n-1])}{\partial \mathbf{s}_W[n-1]}\right|_{\mathbf{s}_W[n-1]=\widehat{\mathbf{s}}_W^{(i)}[n-1|n]}$;
- $\mathbf{M}^{(i)}[n|n-1]=\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}+\mathbf{C}_{u_W}$;
- $\boldsymbol{\Phi}^{(i)}=\mathbf{H}^T\mathbf{C}_w^{-1}\mathbf{H}+\mathbf{M}^{(i)}[n|n-1]^{-1}$;
- $\mathbf{K}^{(i)}[n]=\mathbf{M}^{(i)}[n|n-1]\mathbf{H}^T\left(\mathbf{H}\mathbf{M}^{(i)}[n|n-1]\mathbf{H}^T+\mathbf{C}_w\right)^{-1}$;
- $\mathbf{S}^{(i)}=\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\mathbf{M}^{(i)}[n|n-1]^{-1}$;
- $\widehat{\mathbf{s}}_W^{(i)}[n|n-1]=\boldsymbol{f}(\widehat{\mathbf{s}}_W^{(i)}[n-1|n])$;
- $\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]=\widehat{\mathbf{s}}_W^{(i)}[n|n-1]+\mathbf{K}^{(i)}[n]\left(\mathbf{x}[n]-\widehat{\mathbf{s}}^{(i)}[n|n-1]\right)$;
- $\widehat{\mathbf{s}}_W^{(i+1)}[n|n]=prox^{\boldsymbol{\Phi}^{(i)}}_{\beta\|\cdot\|_1+i_{R_+^M}(\cdot)}(\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n])$;
- $\widehat{\mathbf{s}}_W^{(i+1)}[n-1|n]=\widehat{\mathbf{s}}_W[n-1|n-1]+\mathbf{S}^{(i)}\left[\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\widehat{\mathbf{s}}_W^{(i)}[n|n-1]-\mathbf{F}^{(i)}\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right]$;
- $i=i+1$;

while ($i<$ MaxIter) *or* $\|\widehat{\mathbf{s}}_W^{(i)}[n|n]-previous\_s\|>\epsilon$;
$\widehat{\mathbf{s}}_W[n|n]=\widehat{\mathbf{s}}_W^{(i)}[n|n]$;
$\mathbf{M}[n|n]=\left(\mathbf{I}-\mathbf{K}^{(i-1)}[n]\mathbf{H}^T\right)\mathbf{M}^{(i-1)}[n|n-1]$;

B. Proposed PN-IEKF-VAR Approach

The PN-IEKF-VAR is designed to identify Granger causality between different nodes in directed graphs by using the model in Sec. II-B. Similar to the procedures described in Sec. III-A, the augmented state vector is

$$\mathbf{s}_W[n]=\left[\begin{array}{c}\bar{\mathbf{s}}[n]\\ \bar{\boldsymbol{\omega}}[n]\end{array}\right]\in\mathbb{R}^{K(M+J)}, \tag{18}$$

where $\bar{\boldsymbol{\omega}}[n]\triangleq(vec(\{\mathbf{W}_1[n]\})^T,\dots vec(\{\mathbf{W}_K[n]\})^T)^T\in\mathbb{R}^{KJ}$. The augmented state-space model immediately follows as

$$\begin{aligned}\mathbf{s}_W[n] &= \left[\begin{array}{c}\boldsymbol{\Theta}[n]\bar{\mathbf{s}}[n-1]\\ \bar{\boldsymbol{\omega}}[n-1]\end{array}\right]+\left[\begin{array}{c}\mathbf{u}_\theta[n]\\ \mathbf{u}_\omega[n]\end{array}\right]\\ &= \boldsymbol{f}(\mathbf{s}_W[n-1])+\mathbf{u}_W[n],\end{aligned} \tag{19}$$

where $\mathbf{u}_\omega[n]$ and $\mathbf{u}_W[n]$ are the process noise for $\bar{\boldsymbol{\omega}}[n]$ and $\mathbf{s}_W[n]$, respectively. The observation equation becomes

$$\mathbf{x}[n]=\mathbf{H}\mathbf{s}_W[n]+\mathbf{w}[n]=\mathbf{s}[n]+\mathbf{w}[n], \tag{20}$$

where $\mathbf{H}=\left[\begin{array}{ccc}\mathbf{I}_M & \mathbf{0}_{M\times(K-1)M} & \mathbf{0}_{M\times KJ}\end{array}\right]$ and $\mathbf{w}[n]\sim\mathcal{N}(\mathbf{0}_M,\mathbf{C}_w)$ is the white Gaussian observation noise that is independent of $\mathbf{s}[n]$.

In the subsequent formulation for estimating $\mathbf{s}_W[n]$ and $\mathbf{s}_W[n-1]$, the necessary changes that need to be made compared to the PN-IEKF is $g\left(\mathbf{s}_W[n]\right)$ in the objective in (17) in order to enforce the same sparsity pattern for $\mathbf{W}_1[n]$ to $\mathbf{W}_K[n]$ in (5) [30]. Specifically, let $g(\mathbf{s}_W[n])=\beta\sum_{p=1}^M\sum_{r=1,r\neq p}^M\|\mathbf{w}_{p,r}[n]\|_2$, where $\mathbf{w}_{p,r}[n]\in\mathbb{R}^K$ collects all the (p,r) elements in $\mathbf{W}_k[n]$, $k=1,...,K$. The reason for modifying $g(\mathbf{s}_W[n])$ is to enforce the group sparsity structure, where the existence of an edge from node r to node p is reflected by the value of $\mathbf{w}_{p,r}[n]$ being all zero or not. To conclude, the PN-IEKF-VAR uses the Gauss-Newton method and the proximal Newton method to solve (16) and (17) with $g(\mathbf{s}_W[n])\triangleq\beta\sum_{p=1}^M\sum_{r=1,r\neq p}^M\|\mathbf{w}_{p,r}[n]\|_2$. As a result, when

solving (17) through the proximal Newton method, the scaled proximal step is written as

$$\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n|n] \\
=&prox_g^{\mathbf{\Phi}}\left(\widehat{\mathbf{s}}_W^{(i)}[n|n]-\mathbf{\Phi}^{-1}\nabla f\left(\widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right) \\
=&\arg\min_{\mathbf{s}_W}\ \frac{1}{2}\left(\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right)^T\mathbf{\Phi}\left(\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right) \\
&+g\left(\mathbf{s}_W\right) \\
=&\arg\min_{\mathbf{s}_W}\ \frac{1}{2}\left\|\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right\|_{\mathbf{\Phi}}^2+\beta\sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\|\mathbf{w}_{p,r}[n]\|_2\,,
\end{aligned} \tag{21}$$

which is solved by the alternating direction method of multipliers (ADMM) method, as described in the following.

To solve (21) using the ADMM method, first a new variable $\bar{\mathbf{z}}$ is introduced to split the problem. Hence (21) is rewritten as

$$\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n|n] \\
&=\arg\min_{\mathbf{s}_W,\bar{\mathbf{z}}}\frac{1}{2}\left\|\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right\|_{\mathbf{\Phi}}^2+\beta\sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\|\mathbf{z}_{p,r}\|_2 \\
&\qquad\text{s.t.}\quad \bar{\boldsymbol{\omega}}-\bar{\mathbf{z}}=\mathbf{0}_{KJ}
\end{aligned} \tag{22}$$

where $\bar{\boldsymbol{\omega}}\triangleq\left[vec(\mathbf{W}_1)^T,\ \ldots,\ vec(\mathbf{W}_K)^T\right]^T=\mathbf{G}\mathbf{s}_W\in\mathbb{R}^{KJ}$ with $\mathbf{G}\triangleq\left[\begin{array}{cc}\mathbf{0}_{KJ\times M} & \mathbf{I}_{KJ}\end{array}\right]$ used to extract the part of $\mathbf{s}_W$ related to the graph adjacency matrix, and $\bar{\mathbf{z}}\triangleq\left[vec(\mathbf{Z}_1)^T,\ \ldots,\ vec(\mathbf{Z}_K)^T\right]^T\in\mathbb{R}^{KJ}$. Similar to the definition of $\mathbf{w}_{p,r}$, $\mathbf{z}_{p,r}\in\mathbb{R}^K$ collects all the (p,r) elements in $\mathbf{Z}_k,\ k=1,...,K$. The augmented Lagrangian function for (22) can then be written as

$$\begin{aligned}
\mathcal{L}_\rho\left(\mathbf{s}_W,\bar{\mathbf{z}},\mathbf{y}\right)=&\frac{1}{2}\left\|\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right\|_{\mathbf{\Phi}}^2+\frac{\rho}{2}\left\|\mathbf{G}\mathbf{s}_W-\bar{\mathbf{z}}\right\|_2^2 \\
&+\mathbf{y}^T\left(\mathbf{G}\mathbf{s}_W-\bar{\mathbf{z}}\right)+\beta\sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\left(\|\mathbf{z}_{p,r}\|_2\right),
\end{aligned} \tag{23}$$

where $\mathbf{y}$ is the Lagrangian multiplier. Defining the scaled dual variable $\mathbf{y}_s\triangleq\frac{1}{\rho}\mathbf{y}$, then (23) can be rewritten in scaled form ADMM as

$$\begin{aligned}
&\mathcal{L}_\rho\left(\mathbf{s}_W,\bar{\mathbf{z}},\mathbf{y}_s\right) \\
=&\frac{1}{2}\left\|\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]\right\|_{\mathbf{\Phi}}^2+\frac{\rho}{2}\left\|\left(\mathbf{G}\mathbf{s}_W-\bar{\mathbf{z}}\right)+\mathbf{y}_s\right\|_2^2 \\
&+\beta\sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\|\mathbf{z}_{p,r}\|_2-\frac{\rho}{2}\|\mathbf{y}_s\|_2^2
\end{aligned} \tag{24}$$

To find the saddle point of $\mathcal{L}_\rho$, the ADMM iterates to update $\mathbf{s}_W$, $\bar{\mathbf{z}}$, $\mathbf{y}_s$ until convergence. The update equations are derived as

$$\begin{aligned}
\mathbf{s}_W^{(k+1)}=&\arg\min_{\mathbf{s}_W}\ \frac{1}{2}\left\|\mathbf{s}_W-\widehat{\mathbf{s}}_W^{(i+1)\prime}[n]\right\|_{\mathbf{\Phi}}^2 \\
&+\frac{\rho}{2}\|\mathbf{G}\mathbf{s}_W-\bar{\mathbf{z}}^{(k)}+\mathbf{y}_s^{(k)}\|_2^2
\end{aligned} \tag{25}$$

$$\begin{aligned}
\bar{\mathbf{z}}^{(k+1)}=&\arg\min_{\bar{\mathbf{z}}}\ \beta\sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\|\mathbf{z}_{p,r}\|_2 \\
&+\frac{\rho}{2}\left\|\bar{\mathbf{z}}-\bar{\boldsymbol{\omega}}^{(k+1)}+\mathbf{y}_s^{(k)}\right\|_2^2
\end{aligned} \tag{26}$$

$$\mathbf{y}_s^{(k+1)}=\mathbf{y}_s^{(k)}+\bar{\boldsymbol{\omega}}^{(k+1)}-\bar{\mathbf{z}}^{(k+1)} \tag{27}$$

The update of $\bar{\mathbf{z}}^{(k+1)}$ can be split and updated as $\mathbf{z}_{p,r}^{(k+1)}$ in parallel, for $p\neq r$. The update equation for $\mathbf{z}_{p,r}$ becomes

$$\begin{aligned}
\mathbf{z}_{p,r}^{k+1}=&\arg\min_{\mathbf{z}_{p,r}}\ \frac{\rho}{2}\|\mathbf{z}_{p,r}-\boldsymbol{\nu}_{p,r}\|_2^2+\beta\|\mathbf{z}_{p,r}\|_2, \\
&\qquad\qquad\forall p,r=1,\ldots,M,r\neq p \\
=&S_{\frac{\beta}{\rho}}\left(\boldsymbol{\nu}_{p,r}\right),\forall p,r=1,\ldots,M,r\neq p
\end{aligned} \tag{28}$$

where $S_{\frac{\beta}{\rho}}(\cdot)$ is the block soft-thresholding operator which is defined as [43, pp. 187, 189]

$$S_{\frac{\beta}{\rho}}\left(\boldsymbol{\nu}_{p,r}\right)\triangleq\left\{\begin{array}{ll}\left(1-(\frac{\beta}{\rho})\frac{1}{\|\boldsymbol{\nu}_{p,r}\|_2}\right)\boldsymbol{\nu}_{p,r} & \|\boldsymbol{\nu}_{p,r}\|_2\geq\frac{\beta}{\rho} \\ 0 & \|\boldsymbol{\nu}_{p,r}\|_2<\frac{\beta}{\rho}\end{array}\right. .$$

Algorithm 2 illustrates the procedure for the PN-IEKF-VAR algorithm, which calls Algorithm 3 to perform the backward step to ensure group sparsity is enforced.

Algorithm 2: Proposed PN-IEKF-VAR online graph learning algorithm.

Result: $\widehat{\mathbf{s}}_W[n|n]$, $\mathbf{M}[n|n]$
Initialization: $\widehat{\mathbf{s}}_W^{(0)}[n|n]=\widehat{\mathbf{s}}_W[n|n-1]$; $\widehat{\mathbf{s}}_W^{(0)}[n-1|n]=\widehat{\mathbf{s}}_W[n-1|n-1]$; $i=0$;

do
- $previous\_s=\widehat{\mathbf{s}}_W^{(i)}[n|n]$;
- $\mathbf{F}^{(i)}=\frac{\partial\boldsymbol{f}(\mathbf{s}_W[n-1])}{\partial\mathbf{s}_W[n-1]}\Big|_{\mathbf{s}_W[n-1]=\widehat{\mathbf{s}}_W^{(i)}[n-1|n]}$;
- $\mathbf{M}^{(i)}[n|n-1]=\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}+\mathbf{C}_{uW}$;
- $\mathbf{\Phi}^{(i)}=\mathbf{H}^T\mathbf{C}_w^{-1}\mathbf{H}+\mathbf{M}^{(i)}[n|n-1]^{-1}$;
- $\mathbf{K}^{(i)}[n]=\mathbf{M}^{(i)}[n|n-1]\mathbf{H}^T\left(\mathbf{H}\mathbf{M}^{(i)}[n|n-1]\mathbf{H}^T+\mathbf{C}_w\right)^{-1}$;
- $\mathbf{S}^{(i)}=\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\mathbf{M}^{(i)}[n|n-1]^{-1}$;
- $\widehat{\mathbf{s}}_W^{(i)}[n|n-1]=\boldsymbol{f}(\widehat{\mathbf{s}}_W^{(i)}[n-1|n])$;
- $\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n]=\widehat{\mathbf{s}}_W^{(i)}[n|n-1]+\mathbf{K}^{(i)}[n]\left(\mathbf{x}[n]-\widehat{\mathbf{s}}^{(i)}[n|n-1]\right)$;
- $\widehat{\mathbf{s}}_W^{(i+1)}[n|n]\longleftarrow$ solve (21) with Algorithm 3;
- $\widehat{\mathbf{s}}_W^{(i+1)}[n-1|n]=\widehat{\mathbf{s}}_W[n-1|n-1]+\mathbf{S}^{(i)}\left[\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\widehat{\mathbf{s}}_W^{(i)}[n|n-1]-\mathbf{F}^{(i)}\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right]$;
- $i=i+1$;

while ($i<$ MaxIter) *or* $\|\widehat{\mathbf{s}}_W^{(i)}[n|n]-previous\_s\|>\epsilon$;
$\widehat{\mathbf{s}}_W[n|n]=\widehat{\mathbf{s}}_W^{(i)}[n|n]$;
$\mathbf{M}[n|n]=\left(\mathbf{I}-\mathbf{K}^{(i-1)}[n]\mathbf{H}^T\right)\mathbf{M}^{(i-1)}[n|n-1]$;

Algorithm 3: ADMM for solving (21)

Input : $\widehat{\mathbf{s}}_W^{(i+1)\prime}[n|n], \mathbf{\Phi}^{(i)}$
Initialization: $k=0$
Output: $\widehat{\mathbf{s}}_W^{(i+1)}[n|n]$
do
  $\mathbf{s}_W^{(k+1)} = \arg\min_{\mathbf{s}_W} \frac{1}{2}\left\|\mathbf{s}_W - \widehat{\mathbf{s}}_W^{(i+1)\prime}[n]\right\|^2_{\mathbf{\Phi}^{(i)}}$
  $+\frac{\rho}{2}\|\mathbf{G}\mathbf{s}_W - \bar{\mathbf{z}}^{(k)} + \mathbf{y}_s^{(k)}\|_2^2$;
  $\bar{\mathbf{z}}^{(k+1)} = \arg\min_{\bar{\mathbf{z}}} \beta \sum_{p=1}^{M}\sum_{r=1,r\neq p}^{M}\|\mathbf{z}_{p,r}\|_2$
  $+\frac{\rho}{2}\left\|\bar{\mathbf{z}} - \bar{\boldsymbol{\omega}}^{(k+1)} + \mathbf{y}_s^{(k)}\right\|_2^2$;
  $\mathbf{y}_s^{(k+1)} = \mathbf{y}_s^{(k)} + \bar{\boldsymbol{\omega}}^{(k+1)} - \bar{\mathbf{z}}^{(k+1)}$;
  $k = k+1$;
while *convergence criteria not satisfied*;
$\widehat{\mathbf{s}}_W^{(i+1)}[n|n] = \mathbf{s}_W^{(k)}$;

C. Change Detection

Under normal steady state of the Kalman filter that tracks a state variable modeled in a linear process, the normalized innovation sequence is defined as

$$\boldsymbol{\eta}[n] \triangleq \left(\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T + \mathbf{C}_\omega\right)^{-1/2}\widetilde{\mathbf{x}}[n], \tag{29}$$

which is produced by the Kalman filter and follows a standard normal distribution as indicated in [45], where $\mathbf{M}[n|n-1]$ is the prediction MSE matrix, $\widetilde{\mathbf{x}}[n] \triangleq \mathbf{x}[n] - \widehat{\mathbf{s}}[n|n-1]$ is the (unnormalized) innovation sequence, and $\widehat{\mathbf{s}}[n|n-1]$ is the estimate of $\mathbf{s}[n]$ based on $\mathbf{x}_0^{n-1}$. When faults or changes occur in the linear system, the pdf of the innovation sequence will deviate from the standard normal distribution. The authors in [45] proposed three different strategies to examine whether the Kalman filter is operating at a normal condition based on the distribution of the normalized innovation. Here the T^2 hypothesis test that is in the category of zero-mean test is introduced.

The null and alternative hypotheses are defined to be

$$\begin{aligned}\mathcal{H}_0 &: \boldsymbol{\eta}[n] \sim \mathcal{N}(\mathbf{0}_M, \mathbf{I}_M)\\ \mathcal{H}_1 &: \boldsymbol{\eta}[n] \nsim \mathcal{N}(\mathbf{0}_M, \mathbf{I}_M)\end{aligned} \tag{30}$$

to determine the decision rule of detecting a change or a fault in the system. Denote the sample mean of $\boldsymbol{\eta}[n]$ as $\widehat{\boldsymbol{\mu}}_\eta[n]$ so that $\widehat{\boldsymbol{\mu}}_\eta[n] \sim \mathcal{N}(\mathbf{0}_M, \frac{1}{N}\mathbf{I}_M)$, where N is the number of samples. During the simulation, N number of past time samples are used for computing the statistic at time n. According to [46], the T^2-test that utilizes the T^2-statistic defined as $T^2 \triangleq N\widehat{\boldsymbol{\mu}}_\eta^T\widehat{\mathbf{C}}_\eta^{-1}\widehat{\boldsymbol{\mu}}_\eta$ is the uniformly most powerful test within the zero-mean tests to detect changes in the system, where $\widehat{\mathbf{C}}_\eta$ is the sample covariance matrix of $\boldsymbol{\eta}[n]$. The hypotheses become

$$\begin{aligned}\mathcal{H}_0 &: \frac{N-M}{M(N-1)}T^2[n] \sim F_{M,N-M}\\ \mathcal{H}_1 &: \frac{N-M}{M(N-1)}T^2[n] \nsim F_{M,N-M},\end{aligned} \tag{31}$$

where $F_{M,N-M}$ denotes the F-distribution with degree of freedom M and $N-M$. To determine the decision rule, it should be noted that under $\mathcal{H}_1$, $\frac{N-M}{M(N-1)}T^2[n]$ will deviate from the F-distribution. Therefore, $\mathcal{H}_1$ will be decided when

$$\frac{N-M}{M(N-1)}T^2[n] > \gamma,$$

where γ is the threshold to be determined by fixing the false alarm rate to α so that

$$\alpha = Pr\left(\frac{N-M}{M(N-1)}T^2[n] > \gamma; \mathcal{H}_0\right),$$

which is the probability of $\frac{N-M}{M(N-1)}T^2[n] > \gamma$ under $\mathcal{H}_0$.

When change in the graph occurs, the graph signals that are evolving based on the new graph structure will deviate from its estimate that is predicted based on the previous obtained graph structure. The transition in the graph perturbs the PN-IEKF from the steady state, which causes the normalized innovation sequence to deviate from the white Gaussian distribution. As a result, the T^2 test is suitable for detecting the graph transition. After detection is made by the detector, a nonzero driving noise, $\mathbf{u}_\omega[n]$, is injected into the system to make the adaptation of the PN-IEKF faster to the changes. Thus, the covariance matrix of $\mathbf{u}_\omega[n]$ is set to nonzero. When the PN-IEKF is still at the transient state, the false alarm of the detector will be high, keeping σ_ω^2 nonzero, and the Kalman gain corresponding to $\boldsymbol{\omega}[n]$ will maintain a high value. To prevent the PN-IEKF from diverging as a consequence of the high Kalman gain, a cool down time needs to be set for the detector so that after detecting a change, the next detection cannot be made until a period of the cool down time is met. During this cool time period, the driving noise $\mathbf{u}_\omega[n]$ associated with the graph will equal $\mathbf{0}_P$ or $\mathbf{0}_{KJ}$ for the PN-IEKF or PN-IEKF-VAR, respectively, until the next detected event/transition. The design is effective in the scenario where the graph topology transition will not occur frequently.

IV. Simulation Results

In this section, the simulation results are divided into two parts according to the two signal models and graph topologies described in Sec. II-A and Sec. II-B.

A. Simulation Results of the PN-IEKF Using the Diffusion Model

To test the PN-IEKF algorithm under the diffusion model, synthetic graph generated using the Erdős-Rényi (ER) graph model and data generated based on the graph are used to evaluate the proposed method. To construct the data, a time-varying graph is first generated, and it is substituted into equation (1) for producing the time-varying graph signals.

In every run of the simulations, two time-varying ER (TV-ER) graphs are generated independently using the GSP toolbox [47], each is a connected graph with edge probability parameter $p = 0.4$. When the graph undergoes transition at $n_0 = 14000$ sample, a graph transition will occur. The edge weight is randomly generated based on a uniform distribution $\mathcal{U}[0,1]$. The graph signal $\mathbf{s}[n]$ is generated according to the diffusion process in (1) and is centered according to (2). All the simulation parameters are summarized in Table III.

The parameters are tuned to reach the best RE performance. Specifically, under the scenario where the graph does not change, MaxIter and β are alternately tuned iteratively to find the lowest RE among all the trials. The value of σ_ω^2 is found such that when it is set to a nonzero value at the time at which the graph changes, the convergence of the PN-IEKF is the fastest. The false alarm probability α is chosen after σ_ω^2 is fixed. α is tuned by evaluating the percentage of actual false alarm and the detection among 520 experiments. The false alarms occurring before the graph changes is around 30% among 520 experiments. Due to the definition of $\mathbf{s}_W[n]$ in (8) and (18), the error covariance matrix $\mathbf{M}[-1|-1]$ can be divided into four portions: the upper left portion, denoted as $\mathbf{M}_{\text{upper}}[-1|-1]$, which corresponds to $\bar{\mathbf{s}}[n]$, the upper right portion, the lower left portion, and the lower right portion, denoted as $\mathbf{M}_{\text{lower}}[-1|-1]$, which corresponds to $\boldsymbol{\omega}[n]$ or $\bar{\boldsymbol{\omega}}[n]$ for the PN-IEKF algorithm or the PN-IEKF-VAR algorithm, respectively. Based on different sizes of the state vector in the two algorithms, $\mathbf{M}_{\text{upper}}[-1|-1]$ is initialized to be either $a\mathbf{I}_M$ (PN-IEKF) or $a\mathbf{I}_{KM}$ (PN-IEKF-VAR). Similarly, $\mathbf{M}_{\text{lower}}[-1|-1]$ is initialized to be either $b\mathbf{I}_P$ or $b\mathbf{I}_{KJ}$.

Simulation Parameters			
Number of vertices (M)	10	10	30
SNR	5 dB	20 dB	20 dB
Variance of process noise $\bar{\mathbf{u}}[n]$ (σ_u^2)	1	1	1
Variance of process noise $\mathbf{u}_\omega[n]$ (σ_ω^2)	0.045	0.002	0.001
Sampling period (T)	1	1	1
Number of Monte Carlo experiments	100	100	100
Number of time samples for warm start	90	90	90
Max. number of IEKF iterations (MaxIter)	3	4	6
Max. number of ISTA iterations	15000	15000	15000
Transition time (n_0)	14000	14000	14000
Number of time samples used in T-squared test (N)	200	200	200
Sparsity regularization factor (β)	0.18	0.029	0.01
Edge probability parameter (p)	0.4	0.4	0.4
False alarm rate (α)	10^{-4}	10^{-5}	10^{-5}
Cool down time samples	6000	6000	6000
Initial MSE of estimated graph signals (a)	100	100	100
Initial MSE of estimated graph (b)	1	1	1

TABLE III: Simulation parameters for PN-IEKF.

The accuracy of the estimate of the graphs is measured using the relative error RE $\triangleq \frac{\|\mathbf{W}[n]-\widehat{\mathbf{W}}[n]\|_F}{\|\mathbf{W}[n]\|_F}$ and the normalized mean square deviation NMSD $\triangleq \frac{E\left[\left\|\bar{\mathbf{A}}[n]-\widehat{\bar{\mathbf{A}}}[n]\right\|_F^2\right]}{E\left[\left\|\bar{\mathbf{A}}[n]\right\|_F^2\right]}$. Figs. 2, 3, 4, and 5 show the results for comparing the performance of the PN-IEKF and the LMS approaches in [19] at 20 dB SNR with $M = 10$ and $M = 30$ respectively; Figs. 6 and 7 show the comparison at 5 dB SNR with $M = 10$. "LMS (no proj.)," "LMS (proj. 1)" and "LMS (proj. 2)" correspond to the vanilla LMS approach and the LMS approach aided with projection steps detailed in (32) and (33), respectively. Specifically, "LMS (no proj.)" only performs LMS update on $\widehat{\bar{\mathbf{A}}}$; "LMS (proj. 1)" takes the estimate from "LMS (no proj.)" and sequentially performs the projection steps in (32a), (32b) and (32c). "LMS (proj. 2)" takes the result from "LMS (proj. 1)" and carries out the projection in (33) by thresholding the eigenvalue matrix, $\mathbf{\Lambda}_{\bar{A}}$, of the input matrix $\bar{\mathbf{A}}$ in (33a), where the threshold is specified in (33b).

$$\left[\text{Proj}_{\mathcal{C}_{ele}}(\bar{\mathbf{A}})\right]_{i,j} = \begin{cases} \bar{\mathbf{A}}_{i,j} & \text{if } \bar{\mathbf{A}}_{i,j} \geq -\frac{1}{M} \\ -\frac{1}{M} & \text{else} \end{cases} \tag{32a}$$

$$\text{Proj}_{\mathcal{C}_{null}}(\bar{\mathbf{A}}) = \bar{\mathbf{A}} - \frac{1}{M}\bar{\mathbf{A}}\mathbf{1}_M\mathbf{1}_M^T \tag{32b}$$

$$\text{Proj}_{\mathcal{C}_{sym}}(\bar{\mathbf{A}}) = \frac{1}{2}(\bar{\mathbf{A}} + \bar{\mathbf{A}}^T) \tag{32c}$$

$$\text{Proj}_{\mathcal{C}_{spec}}(\bar{\mathbf{A}}) = \mathbf{V}_{\bar{A}}\mathbf{\Lambda}_t\mathbf{V}_{\bar{A}}^T, \tag{33a}$$

$$[\mathbf{\Lambda}_t]_{ii} = \begin{cases} 0 & \text{if } [\mathbf{\Lambda}_{\bar{A}}]_{ii} < 0 \\ [\mathbf{\Lambda}_{\bar{A}}]_{ii} & \text{if } 0 \leq [\mathbf{\Lambda}_{\bar{A}}]_{ii} \leq 1 \\ 1 & \text{else} \end{cases}. \tag{33b}$$

The full projection steps in "LMS (proj. 2)" are carried out q times at the same time sample in "LMS (proj. 2 with q iter)." "PN-IEKF (no detector)" removes the mechanism of detecting the graph transition. "PN-IEKF (oracle)" is the case where the variance of $\mathbf{u}_\omega[n]$ is set from zero to the value in Table III at n_0. "PN-IEKF" is the proposed PN-IEKF method using the T^2 detector, which is turned on at $n = 6000$ sample. σ_ω^2 will subsequently change from zero to the value in Table III once a graph transition event is detected. The detector will then be turned off, with σ_ω^2 returning to zero, until the cool down time (6000 samples) is reached. The detector will then be turned back on and the detection process will repeat. When the driving noise $\mathbf{u}_\omega[n]$ is injected to $\boldsymbol{\omega}[n]$, the value of σ_ω^2 is nonzero only for one time sample and will be set back to zero in the next sample for both "PN-IEKF (oracle)" and "PN-IEKF." Since the PN-IEKF approach is designed to directly estimate the adjacency matrix $\mathbf{W}$, while the LMS approaches directly find the transition matrix $\bar{\mathbf{A}}[n]$ instead, the comparison of the two algorithms can be done by translating $\widehat{\mathbf{W}}[n]$ to $\widehat{\bar{\mathbf{A}}}[n]$ through $\widehat{\bar{\mathbf{A}}}[n] = e^{-T\widehat{\mathbf{L}}[n]} - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$ and translating $\widehat{\bar{\mathbf{A}}}[n]$ to $\widehat{\mathbf{W}}[n]$ through $\widehat{\mathbf{L}}[n] = \frac{-1}{T}\ln(\widehat{\bar{\mathbf{A}}}[n] + \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)$ followed by extracting the off-diagonal elements from $\widehat{\mathbf{L}}[n]$, where the $\ln(\cdot)$ denotes matrix logarithm. The step size of the LMS methods is chosen such that the convergent NMSD of the "LMS (proj. 2)" is the same as that of the "PN-IEKF (oracle)." The NMSD and the RE results in Figs. 2 and 3 show that the convergence of the proposed PN-IEKF approach is faster than the LMS approaches before the graph transition. After the transition, the accuracy of the PN-IEKF method would degenerate due to the imperfect detection rate, while it can still outperform the LMS method with full projection steps in terms of RE.

The reason is the proposed PN-IEKF method is designed to directly estimate the graph adjacency matrix $\mathbf{W}[n]$, which the RE is based on, and not the transition matrix $\bar{\mathbf{A}}[n]$. Also, the mapping between $\widehat{\mathbf{W}}[n]$ and $\widehat{\bar{\mathbf{A}}}[n]$ is not isomorphic, i.e. the ranking of performance in the plot of RE of $\mathbf{W}[n]$ is not necessary consistent with that in the NMSD plot of $\bar{\mathbf{A}}[n]$. Furthermore, since the LMS approach does not guarantee the produced $\widehat{\mathbf{A}}[n]$ matrices satisfy the requirement of the matrix logarithm [48], [49], which needs to be positive definite, which explains the fluctuations in the "LMS (proj 2)" case in Fig 5 when M is increased from 10 to 30 and in Fig. 7 when the SNR is decreased from 20 dB to 5 dB while $M = 10$. Interestingly, the NMSD results for both of these cases are shown in Figs. 4 and 6 where the LMS approaches use a much higher step size than the one in Figs. 2 and 3, which resulted in a faster convergence. However, the resulting $\widehat{\mathbf{W}}[n]$ would be inaccurate, which explains the fluctuations for all LMS cases in Figs. 5 and 7.

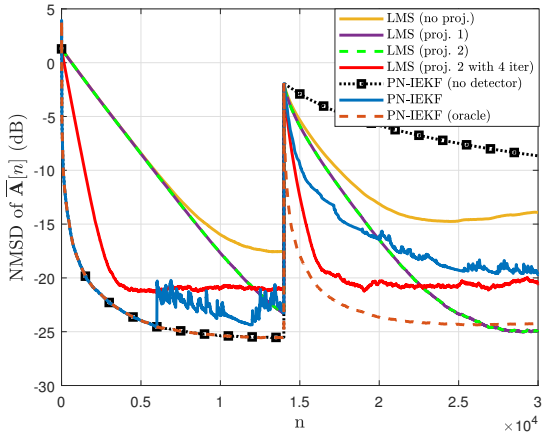


Fig. 2: Normalized mean square deviation vs. time sample n comparison between the PN-IEKF oracle, the PN-IEKF method, three LMS approaches proposed in [19]. $M = 10$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

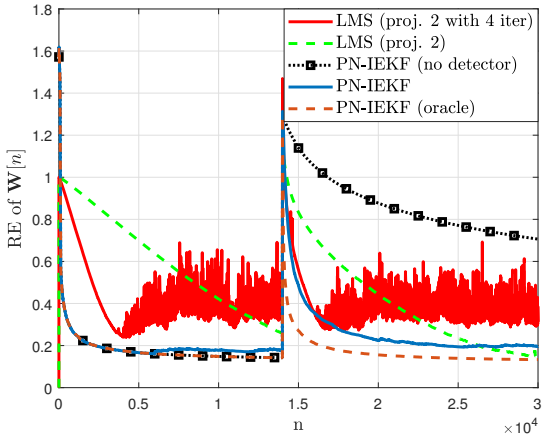


Fig. 3: Relative error vs. time sample n comparison between the PN-IEKF oracle, proposed PN-IEKF method, and the LMS method of the best convergence speed in [19]. $M = 10$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

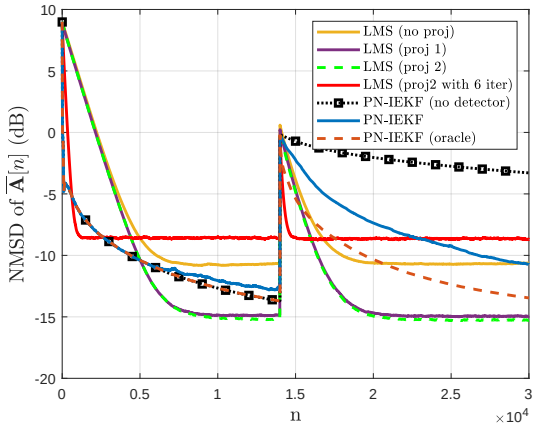


Fig. 4: Normalized mean square deviation vs. time sample n comparison between the PN-IEKF oracle, the PN-IEKF method, three LMS approaches proposed in [19]. $M = 30$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

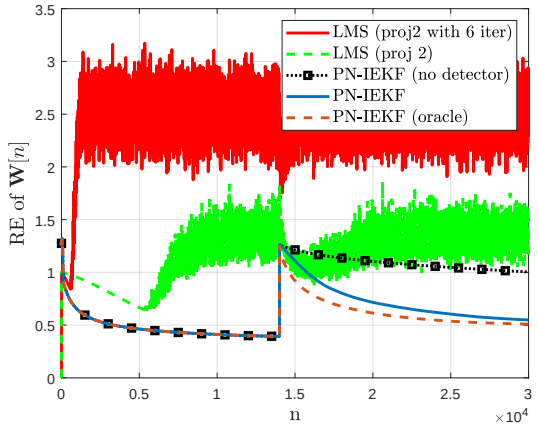


Fig. 5: Relative error vs. time sample n comparison between the PN-IEKF oracle, proposed PN-IEKF method, and the LMS method of the best convergence speed in [19]. $M = 30$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

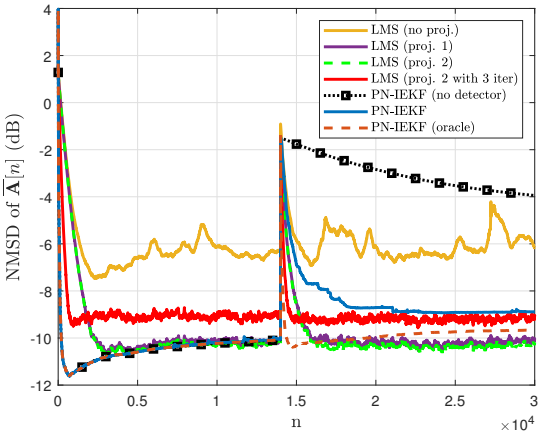


Fig. 6: Normalized mean square deviation vs. time sample n comparison between the PN-IEKF oracle, the PN-IEKF method, three LMS approaches proposed in [19]. $M = 10$, SNR $= 5$ dB. Transition occurs at $n_0 = 14000$ sample.

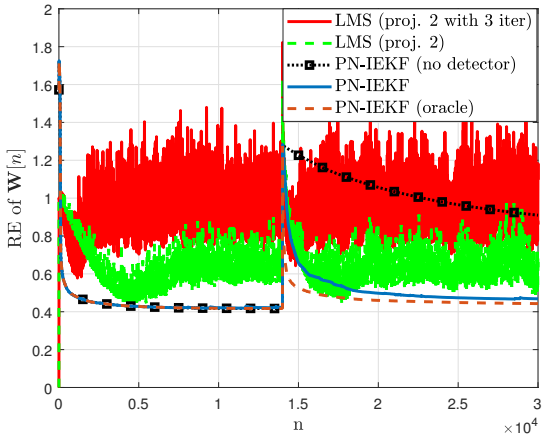
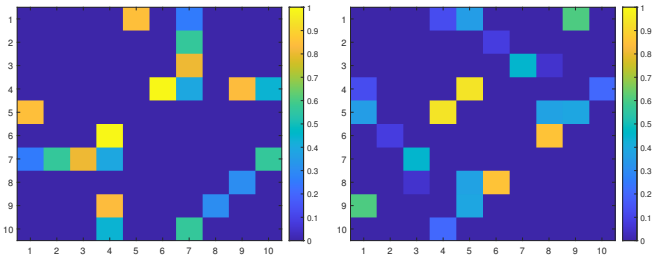


Fig. 7: Relative error vs. time sample n comparison between the PN-IEKF oracle, proposed PN-IEKF method, and the LMS method of the best convergence speed in [19]. $M = 10$, SNR $= 5$ dB. Transition occurs at $n_0 = 14000$ sample.

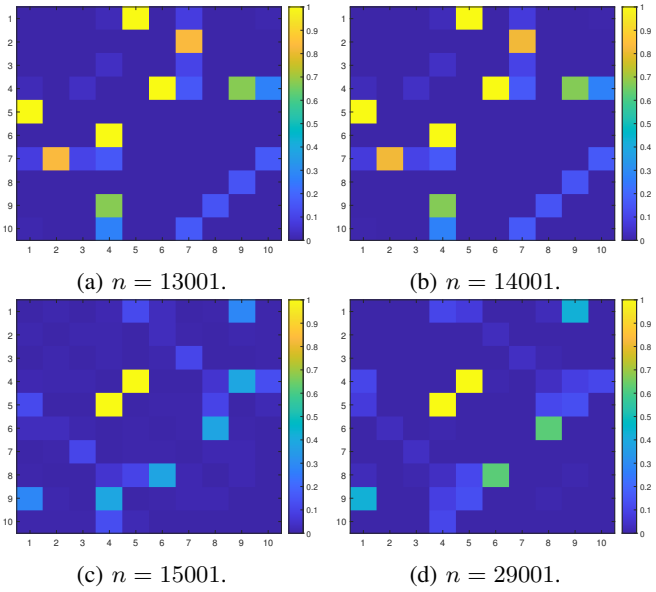
The graph tracking results before and after the graph transition can be visualized using color maps. Specifically, the estimated graph is obtained at $n = 13001$ before the change occurs, right after the change occurs while the change has not been detected at $n = 14001$, after the change has been detected and the new estimate has been reacquired at $n = 15001$, and at $n = 29001$ when the PN-IEKF algorithm has longer time to acquire a more accurate estimate of the new graph. Figs. 8, 9, and 10 show color maps of the ground truth graphs and estimated graphs using the proposed PN-IEKF algorithm at different times and at different SNRs of one of the realizations in the experiments. As seen from these figures, the PN-IEKF algorithm is capable of tracking time varying graphs and obtaining sparse estimate of the graphs.



(a) $n < 14000$. (b) $n \geq 14000$.

Fig. 8: Color map of ground truth graphs using the proposed PN-IEKF algorithm before and after the transition time $n_0 = 14000$.

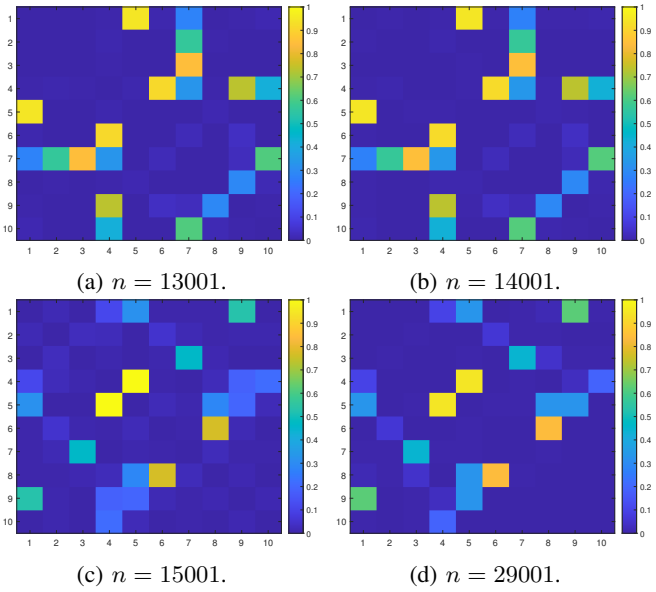
Next, the performance of the PN-IEKF algorithm is evaluated when model mismatch exists. Specifically, the mismatch lies between the assumed variance and the true variance of the process noise of graph signal, σ_u^2. Figs. 11 and 12 show the NMSD and RE results, respectively, of increasing the assumed value of σ_u^2 while keeping the true value of $\sigma_u^2 = 1$. Since the values of the hyperparameters are tuned to achieve the best performance in terms of RE, it can be found from the RE curves in Fig. 12 that when the gap between the true value



(a) $n = 13001$. (b) $n = 14001$.

(c) $n = 15001$. (d) $n = 29001$.

Fig. 9: Color map of learned graphs using the proposed PN-IEKF algorithm at different time. SNR $= 5$ dB.



(a) $n = 13001$. (b) $n = 14001$.

(c) $n = 15001$. (d) $n = 29001$.

Fig. 10: Color map of learned graphs using the proposed PN-IEKF algorithm at different time. SNR $= 20$ dB.

and the assumed value of σ_u^2 becomes larger, the accuracy of the PN-IEKF will decline, and the RE value will plateau at a higher level.

B. Simulation Results of the PN-IEKF-VAR Using the VAR Model

The simulation conditions for the PN-IEKF-VAR is similar to those of PN-IEKF except the graph signals are generated using the VAR process. Simulation parameters are summarized in Table IV. Since the order of the VAR process equals to $K = 2$, two adjacency matrices are generated, hence the NMSD $\triangleq$

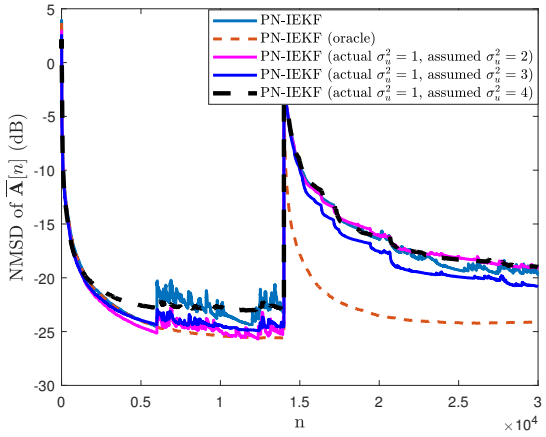


Fig. 11: NMSD vs. time sample n comparison between the PN-IEKF oracle, the PN-IEKF method without mismatch, the PN-IEKF method with mismatch between the assumed value of σ_u^2 and the true value of σ_u^2. SNR $=$ 20 dB. Transition occurs at $n_0 = 14000$ sample.

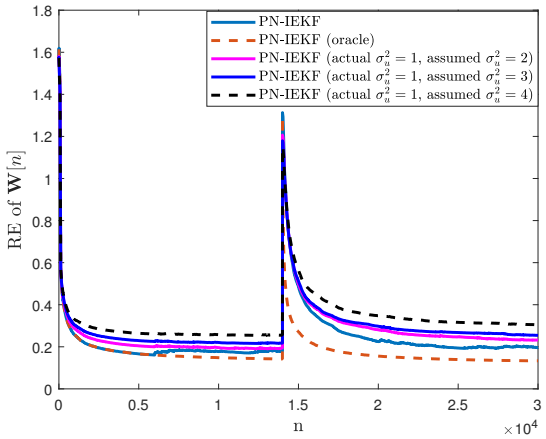


Fig. 12: RE vs. time sample n comparison between the PN-IEKF oracle, the PN-IEKF method without mismatch, the PN-IEKF method with mismatch between the assumed value of σ_u^2 and the true value of σ_u^2. SNR $=$ 20 dB. Transition occurs at $n_0 = 14000$ sample.

$\frac{E\left[\left\|\widehat{\overline{\mathbf{W}}}[n]-\overline{\mathbf{W}}[n]\right\|_F^2\right]}{E\left[\left\|\overline{\mathbf{W}}[n]\right\|_F^2\right]}$ and RE $\triangleq \frac{\left\|\widehat{\overline{\mathbf{W}}}[n]-\overline{\mathbf{W}}[n]\right\|_F}{\left\|\sum_{k=1}^{K}\overline{\mathbf{W}}[n]\right\|_F}$, where $\overline{\mathbf{W}} \triangleq \begin{bmatrix}\mathbf{W}_1 & \cdots \mathbf{W}_K\end{bmatrix}$. Similar definition is used for $\widehat{\overline{\mathbf{W}}}$. Figs. 13, 14 show the NMSD and RE results with a 10 vertices graph at SNR = 20 dB, respectively. Figs. 15 and 16 show the NMSD and RE results with a 10 vertices graph at SNR = 5 dB, respectively. Figs. 17 and 18 show the NMSD and RE results with a 30 vertices graph at SNR = 20 dB, respectively. The PN-IEKF-VAR method is compared with the TIRSO method in [30] with 6 different step sizes. Both sets are results show that our proposed PN-IEKF-VAR method is able to converge to a lower NMSD and RE value compared to the TIRSO method with the best step size of $\alpha_n = \frac{1}{L}$ or $\frac{1}{L\sqrt{n}}$.

Under the assumption of a VAR model, when order $K = 2$, two adjacency matrices can be generated. Therefore, the tracking performance can also be understood through visualization using color maps. Specifically, the estimated graph is obtained at $n = 13001$ before the change occurs, right after the change occurs while the change has not been detected at $n = 14001$, after the change has been detected and the new estimate has been reacquired at $n = 15001$, and at $n = 29001$ when the PN-IEKF-VAR algorithm has longer time to acquire a more accurate estimate of the new graph. Figs. 19, 20 and 21 show color maps of the ground truth graphs and estimated graphs using the proposed PN-IEKF-VAR algorithm at different times at SNR = 5 dB. Figs. 22 and 23 show the same results at SNR = 20 dB. Similar to the results for PN-IEKF, at 20 dB, the color maps for $\mathbf{W}_1$ and $\mathbf{W}_2$ look similar to those of the ground truth, especially after the transition at $n = 29001$ when the algorithm has sufficient time to converge. The performance for 20 dB, as expected, is better than that of 5 dB.

Simulation Parameters			
Number of vertices (M)	10	10	30
SNR	20 dB	5 dB	20 dB
Variance of process noise $\bar{\mathbf{u}}[n]$ (σ_u^2)	0.01	0.01	0.01
Variance of process noise $\mathbf{u}_\omega[n]$ (σ_ω^2)	5×10^{-2}	8.75×10^{-2}	8.5×10^{-2}
Sampling period (T)	1	1	1
Number of Monte Carlo experiments	100	100	100
Number of time samples for warm start	90	90	90
Order of the VAR process (K)	2	2	2
Max. number of PN-IEKF-VAR iterations (MaxIter)	2	2	3
Max. number of ADMM iterations	300	300	300
Transition time (n_0)	14000	14000	14000
Number of time samples used in T-squared test (N)	500	500	500
Sparsity regularization factor (β)	7.5×10^{-3}	8.65×10^{-3}	1.35×10^{-2}
Edge probability parameter (p)	0.4	0.4	0.4
False alarm rate (α)	.01	.01	.001
Cool down time samples	6000	6000	6000
Initial MSE of estimated graph signals (a)	100	100	100
Initial MSE of estimated graph (b)	1	1	1

TABLE IV: Simulation parameters for PN-IEKF-VAR.

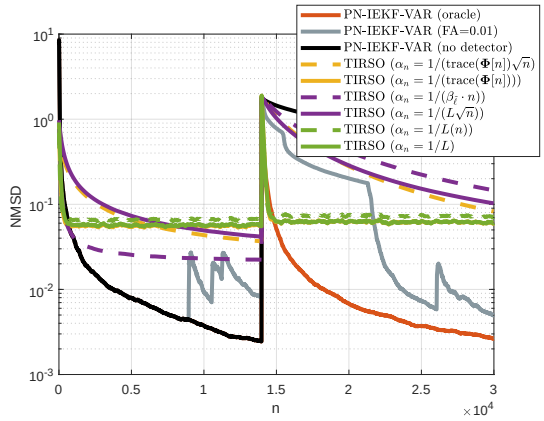


Fig. 13: NMSD vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 10$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

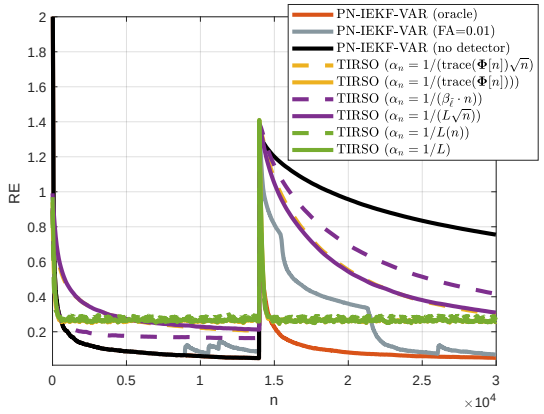


Fig. 14: RE vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 10$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

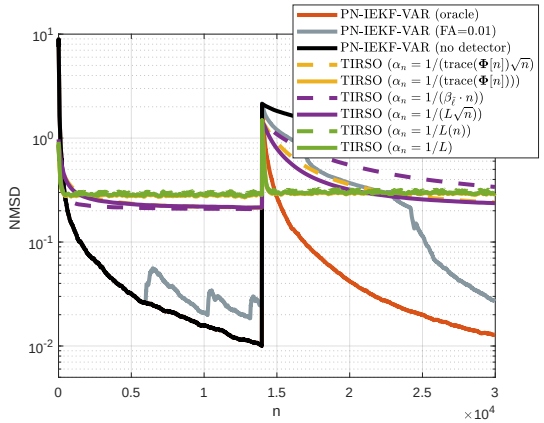


Fig. 15: NMSD vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 10$, SNR $= 5$ dB. Transition occurs at $n_0 = 14000$ sample.

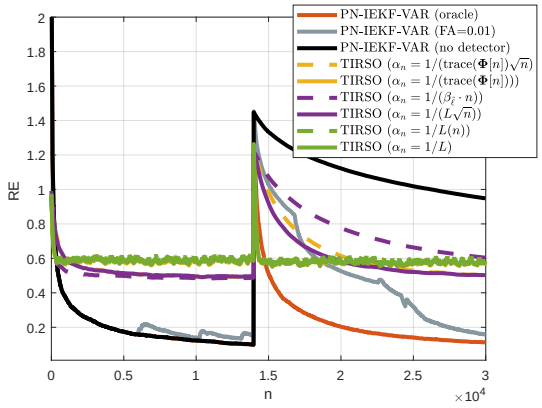


Fig. 16: RE vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 10$, SNR $= 5$ dB. Transition occurs at $n_0 = 14000$ sample.

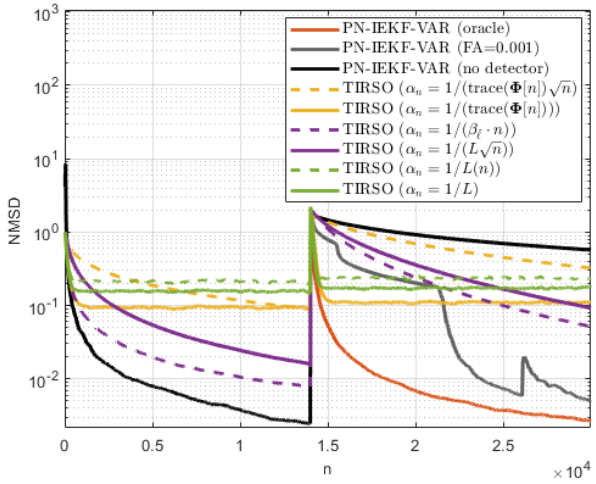


Fig. 17: NMSD vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 30$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.

C. Numerical Results of Actual Dataset Using PN-IEKF

The proposed PN-IEKF method has also been applied to learn actual brain graph from EEG signals. The DEAP dataset [50], which contains a set of EEG time series data where $M = 32$. Note that the dataset contains many subjects, with each one possibly undergoing changes in his/her emotional state due to constant external stimuli. Thus, it is believed that the underlying brain graph should also change its structure as well. Moreover, it is impossible to determine the exact time when or how many times the subject undergoes this shift in emotion in the entire run as the original DEAP experiment did not ask subjects to report any transient emotional changes. Therefore, it is currently beyond our capabilities to accurately gauge the efficacy of the proposed detector on the DEAP dataset.

In the absence of a ground truth graph, the goal is to show the learned brain graph obtained from EEG signals using the PN-IEKF algorithm fits a pattern generated using

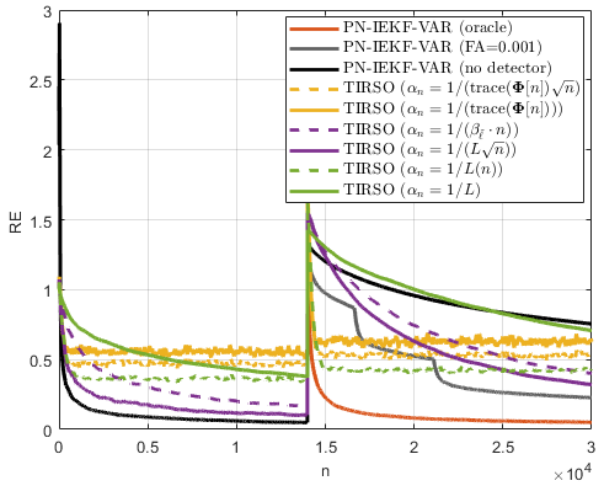
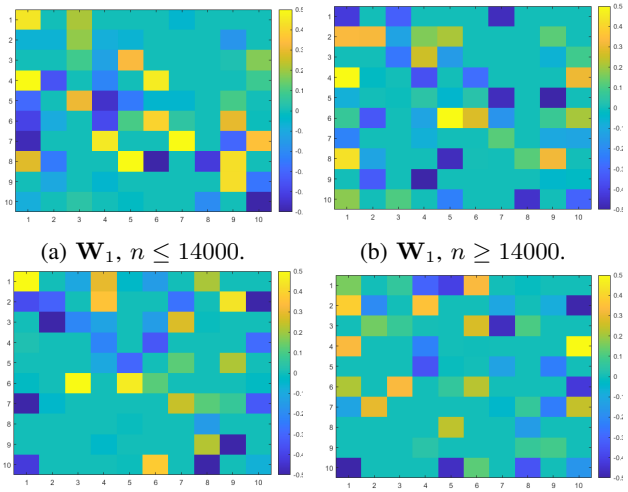


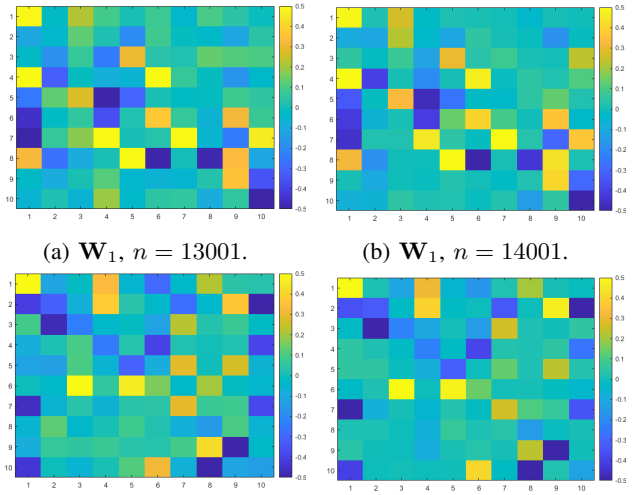
Fig. 18: RE vs. time sample n comparison between the oracle, the PN-IEKF-VAR method, the TIRSO approach with 6 different step sizes proposed in [30]. $M = 30$, SNR $= 20$ dB. Transition occurs at $n_0 = 14000$ sample.



(a) $\mathbf{W}_1$, $n \leq 14000$. (b) $\mathbf{W}_1$, $n \geq 14000$.

(c) $\mathbf{W}_2$, $n \leq 14000$. (d) $\mathbf{W}_2$, $n \geq 14000$.

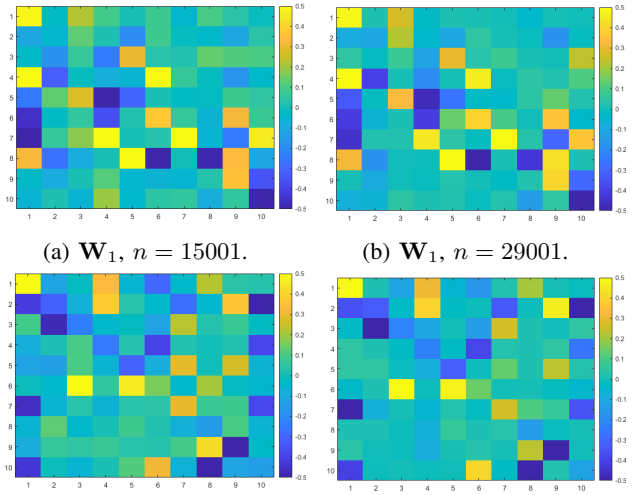
Fig. 19: Color map of ground truth graphs using the proposed PN-IEKF-VAR algorithm before and after the transition time $n_0 = 14000$.



(a) $\mathbf{W}_1$, $n = 13001$. (b) $\mathbf{W}_1$, $n = 14001$.

(c) $\mathbf{W}_2$, $n = 13001$. (d) $\mathbf{W}_2$, $n = 14001$.

Fig. 20: Color map of learned graphs using the proposed PN-IEKF-VAR algorithm at $n = 13001$ and 14001. SNR $= 5$ dB.



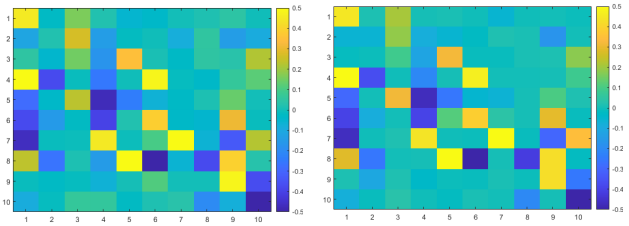
(a) $\mathbf{W}_1$, $n = 15001$. (b) $\mathbf{W}_1$, $n = 29001$.

(c) $\mathbf{W}_2$, $n = 15001$. (d) $\mathbf{W}_2$, $n = 29001$.

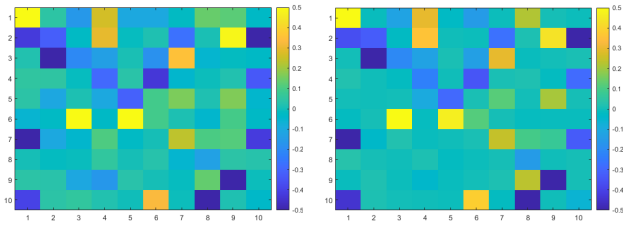
Fig. 21: Color map of learned graphs using the proposed PN-IEKF-VAR algorithm at $n = 15001$ and 29001. SNR $= 5$ dB.

an existing connectivity measure. In this case, the Pearson correlation method [51] was chosen. Furthermore, all negative connectivities are zeroed out, which gives the "ground truth" adjacency matrix. In our configuration, the sparsity regularization factor is set to $\beta = 0.0005$, and the sampling period $T = \frac{1}{128}$ seconds, and the run is assumed to be noise-free, i.e. $\mathbf{w}[n] = \mathbf{0}_M$. The rest of the parameters are the same as those used in the 30 node configuration above for synthetic data. In addition, since the dynamic range of the color map values from the PN-IEKF algorithm is much larger than that of the ground truth, a polynomial regression model is used to 1) reduce the dynamic range of the edge weights, and 2) refine the results of the PN-IEKF method. In actual scenario, such scaling method will not be feasible, and modification to the objective function in (17) may be required, which will be considered in future works. However, such a method allows the PN-IEKF to showcase its potential in real-life problem where the structure of the estimated adjacency matrix is similar to that of the ground truth.

Figs. 24 and 25 show the adjacency matrix in color map format of the ground truth, PN-IEKF without using the polynomial regression model, and using different orders of polynomial regression model, at time instants $n = 8064$ and 6000, respectively. Quantitative results are shown in Tables V and VI, which consist of RE values for $\beta = 0.001$ and $\beta = 0.0005$. Notice that the RE does not decrease indefinitely as the model order increases. This may be because the brain graph changes constantly and the PN-IEKF method is unable to fully converge to the new adjacency matrix due to the insufficient number of samples. This offers a possible explanation as to why later time points (Table V) fail to have

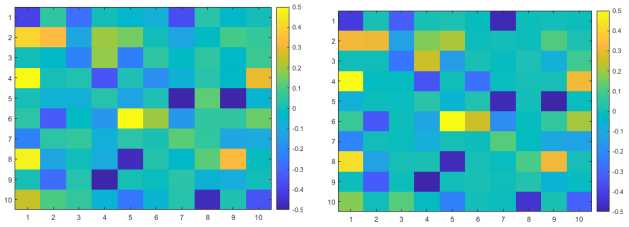


(a) $\mathbf{W}_1$, $n = 13001$. (b) $\mathbf{W}_1$, $n = 14001$.

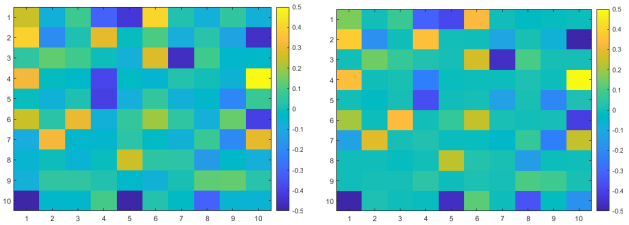


(c) $\mathbf{W}_2$, $n = 13001$. (d) $\mathbf{W}_2$, $n = 14001$.

Fig. 22: Color map of learned graphs using the proposed PN-IEKF-VAR algorithm at $n = 13001$ and 14001. SNR $= 20$ dB.



(a) $\mathbf{W}_1$, $n = 15001$. (b) $\mathbf{W}_1$, $n = 29001$.

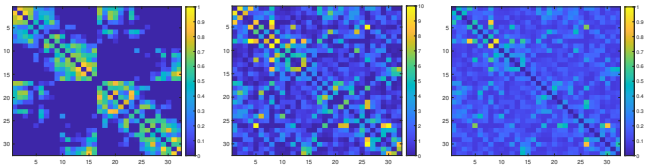


(c) $\mathbf{W}_2$, $n = 15001$. (d) $\mathbf{W}_2$, $n = 29001$.

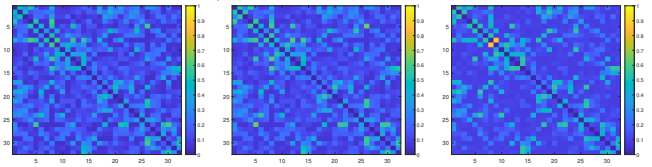
Fig. 23: Color map of learned graphs using the proposed PN-IEKF-VAR algorithm at $n = 15001$ and 29001. SNR $= 20$ dB.

a lower relative error compared to earlier time points (Table VI).

In addition to brain graphs, the PN-IEKF algorithm was also tested using daily average temperature data from [52]. Data from 2012 to 2013 were collected from 10 stations whose locations and names are shown in Fig. 26. Stations from north to south are labeled in order from node 1 to node 10, which implies $M = 10$. Simulation parameters are identical to those in the synthetic data case with 10 nodes, SNR = 20 dB, except the value of β needs to vary in Fig. 27 as its x-axis refers to the percentage of nonzero (p_{nnz}) elements in the adjacency matrix. Upon further testing, β is set to be $\{150, 70, 50, 30, 14, 10, 8, 0.05, 0.9\}$, in this order, that corresponds to the lowest to highest p_{nnz} value. Notice that β does not decrease monotonically and yet, using these values, the p_{nnz} increases. Besides changes in β, the number



(a) Ground Truth (b) PN-IEKF Adjacency matrix, unscaled (c) 1st order poly. scaled ($\beta = 0.0005$)



(d) 5th order poly. scaled (e) 10th order poly. scaled (f) 13th order poly. scaled

Fig. 24: Color maps of adjacency matrices from time point $n = 8064$, scaled using different order polynomial regression models. The adjacency matrix shown in Fig. 24b is scaled using a polynomial regression model to fit that of Fig. 24a, and the following subfigures show its results.

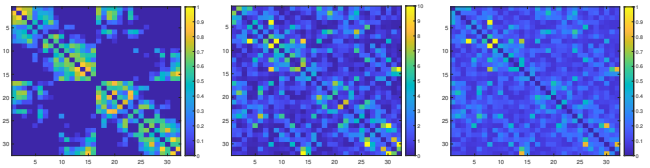
	$\beta = 0.001$	$\beta = 0.0005$
Unscaled	8.1921	8.7886
1st Order	0.7256	0.6817
2nd Order	0.7185	0.6707
3rd Order	0.7169	0.6671
4th Order	0.7013	0.6475
5th Order	0.7001	0.6446
6th Order	0.6995	0.6446
7th Order	0.6990	0.6442
8th Order	0.6973	0.6435
9th Order	0.6949	0.6426
10th Order	0.6940	0.6411
11th Order	0.6943	0.6430
12th Order	0.7098	0.6542
13th Order	0.7400	0.6885

TABLE V: Relative error of adjacency matrix at time $n = 8064$ using different order polynomials.

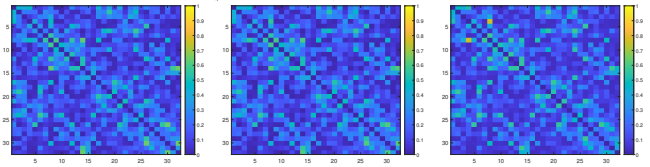
	$\beta = 0.001$	$\beta = 0.0005$
Unscaled	8.1921	8.7886
1st Order	0.6991	0.6496
2nd Order	0.6782	0.6290
3rd Order	0.6779	0.6280
4th Order	0.6729	0.6187
5th Order	0.6651	0.6150
6th Order	0.6628	0.6146
7th Order	0.6626	0.6146
8th Order	0.6624	0.6127
9th Order	0.6574	0.6088
10th Order	0.6573	0.6037
11th Order	0.6581	0.6050
12th Order	0.6603	0.6098
13th Order	0.7706	0.6266

TABLE VI: Relative error of adjacency matrix at time $n = 6000$ using different order polynomials.

of time samples for warm start is changed to 30 from 90 due to the shorter duration of the total number of samples (see



(a) Ground Truth (b) PN-IEKF Adjacency matrix, unscaled (c) 1st order poly. scaled ($\beta = 0.0005$)



(d) 5th order poly. scaled (e) 10th order poly. scaled (f) 13th order poly. scaled

Fig. 25: Color maps of adjacency matrices from time point $n = 6000$, scaled using different order polynomial regression models. The method used is the same as that of $n = 8064$.

below) compared to the synthetic data case, and $\mathbf{w}[n] = \mathbf{0}$. In addition, the detector is turned off since the changes in the underlying graph is more gradual.

Fig. 26: Map view of selected stations from [52]. Stations are labeled as node 1 to node 10 from north to south.

Figs. 27 and 28 show the MSE vs. p_{nnz} (percentage of nonzero entries in the graph) and the squared error SE of the estimated observed signal vs. time n. Both figures contain results from the proposed PN-IEKF (no detector) and different variants of the LMS algorithm in [19]. The mean-squared error in Fig. 27 is defined as $MSE \triangleq \frac{1}{M \times n_{est}} \sum_{i=n_{total}-n_{est}+1}^{n_{total}} \|\mathbf{x}[i] - \widehat{\mathbf{x}}[i]\|^2$, where $n_{total} = 731$ is the total number of time samples and n_{est} is the number of samples to be estimated. For PN-IEKF, $n_{est} = 701$ which exclude the warm start samples and $\widehat{\mathbf{x}}[i]$ correspond to $\widehat{\mathbf{s}}[n|n]$, the estimation of the graph signal part of the state vector $\mathbf{s}_W[n]$. For LMS methods, $n_{est} = 730$ and $\widehat{\mathbf{x}}[i]$ is obtained by $\widehat{\mathbf{x}}[i] = \widehat{\mathbf{A}}[i-1]\mathbf{x}[i-1]$. The reason why only graph signals were evaluated is because the ground truth adjacency matrices are not available. MSE is evaluated as a function of nonzero proportion of the graph $p_{nnz} \triangleq \frac{M_{nnz}}{M(M-1)} = \frac{M_{nnz}}{90}$, where M_{nnz} is the number of nonzero edges in $\widehat{\mathbf{W}}$ [8]. Hence, p_{nnz} is used to represent the sparsity of the graph. The LMS based methods can achieve different p_{nnz} by thresholding the edges. For Fig. 28, the squared error $SE \triangleq \|\mathbf{x}[i] - \widehat{\mathbf{x}}[i]\|^2$ is calculated at each time sample. β is set to be 0.05 for PN-IEKF and no thresholding has been done for LMS methods.

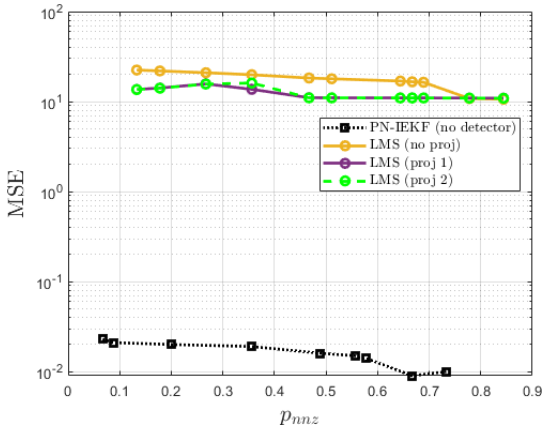


Fig. 27: MSE vs. p_{nnz} comparison between PN-IEKF with no detector and the three variants of LMS method in [19]. $MSE \triangleq \frac{1}{M \times n_{est}} \sum_{i=n_{total}-n_{est}+1}^{n_{total}} \|\mathbf{x}[i] - \widehat{\mathbf{x}}[i]\|^2$ calculates the mean-squared error of the graph signals. $n_{total} = 731$, $n_{est} = 701$ for PN-IEKF and $n_{est} = 730$ for LMS methods. $p_{nnz} \triangleq \frac{M_{nnz}}{M(M-1)} = \frac{M_{nnz}}{90}$ represents the sparsity of the graph [8]. Experiments are done by using two years of data (2012 to 2013) from 10 stations.

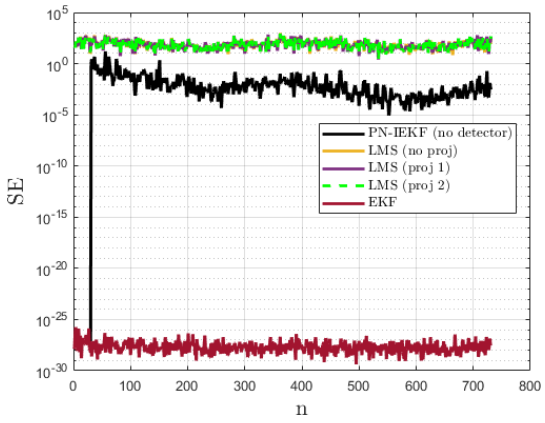


Fig. 28: SE vs. time sample n comparison between PN-IEKF with no detector, three LMS approaches and EKF. $SE \triangleq \|\mathbf{x}[i] - \widehat{\mathbf{x}}[i]\|^2$ calculates the squared-error of the graph signals. $\beta = 0.05$ for PN-IEKF (no detector). Experiments are done by using two years of data (2012 to 2013) from 10 stations.

It is clear that the PN-IEKF performs better than the LMS methods in terms of both MSE and SE. In fact, the MSE performance of the PN-IEKF is better than that of the LMS methods by almost three orders magnitude. The color maps of the resulting adjacency matrices $\widehat{\mathbf{W}}$ at time $n = 180$,

360, 545 and 725 for PN-IEKF, the three LMS variants and EKF are shown in Fig. 29. Similar to Fig. 28, β is again equal to 0.05 for PN-IEKF and no thresholding has to been done for LMS methods. According to the color maps, higher edge weight values can be observed around the main diagonal using the PN-IEKF compare to the LMS methods. Since it is more reasonable for stations that are in the same vicinity geographically to measure similar temperature, this suggests the PN-IEKF performs better than the three LMS variants. Recall in Fig. 28 that the EKF outperforms both the PK-IEKF and LMS based methods in terms of SE. However, the SE only evaluate the performance of estimated graph signals $\mathbf{x}[i]$, and not the estimated adjacency matrix $\widehat{\mathbf{W}}$, hence, in terms of graph tracking, the PN-IEKF is still the best by observing the color maps results in Fig. 30, which directly compares the PN-IEKF and the EKF. In fact, without proper regularization, the EKF can render negative values in the estimated adjacency matrix, therefore, its nodal relationships cannot be clearly attained compare to those of the PN-IEKF.

D. Convergence and Computational Complexity Analysis of PN-IEKF and PN-IEKF-VAR

The convergence of the IEKF method depends on the initial point of the iterations. On one hand, since the method can be viewed as the Gauss-Newton method, whose convergence is highly related to the convexity of the objective. Specifically, the problem for solving $\mathbf{s}_W[n-1|n]$ in (16) is nonconvex, $\widehat{\mathbf{s}}_W[n-1|n]$ found by the PN-IEKF method is not necessary the global minimizer of the problem. As a result, there is no guarantee that the IEKF method will always converge. On the other hand, it is found from our experiments that the warm start using the EKF method often finds a proper starting point for the PN-IEKF method, which reduces the chance for the algorithm to diverge and improves the tracking performance of the state vector given more number of time samples. During the experiment, few cases of divergence occurred.

The bottleneck lies in the relatively high computational complexity of the proposed PN-IEKF method can be attributed to matrix inversion and matrix-matrix multiplications. On lines 4 and 11 of Algorithms 1, the number of multiplications is in the order of $O(P^3)$. When solving the scaled proximal step using ISTA iterations on line 10 of Algorithm 1, each iteration incurs $O(P^2)$ number of operations due to evaluation of the gradient and the exact line search. When $M = 10$, the amount of operations required for the PN-IEKF in each iteration is approximately $1,431,375$ flops, while "LMS (no proj.)," "LMS (proj. 1)" and "LMS (proj. 2)" require 410, 1910, and 4110 flops, respectively.

The computational complexity of PN-IEKF-VAR is higher compared to that of the PN-IEKF. This can be attributed to the increased complexity of the signal model as it uses K number of adjacency matrices to model the graph signal. In Algorithms 2 and 3, the number of multiplication operations from line 4th to the 11th is on the order of $\mathcal{O}(K^3J^3)$. When using the ADMM iterations to solve the 10th line of Algorithm 2, each iteration update results in approximately $\mathcal{O}(K^2J^2)$ operations. When $M = 10$, $K = 2$, PN-IEKF-VAR requires approximately 54,460,000 flops per iteration while the TIRSO algorithm requires approximately 848,000 (for any step sizes) flops per iteration.

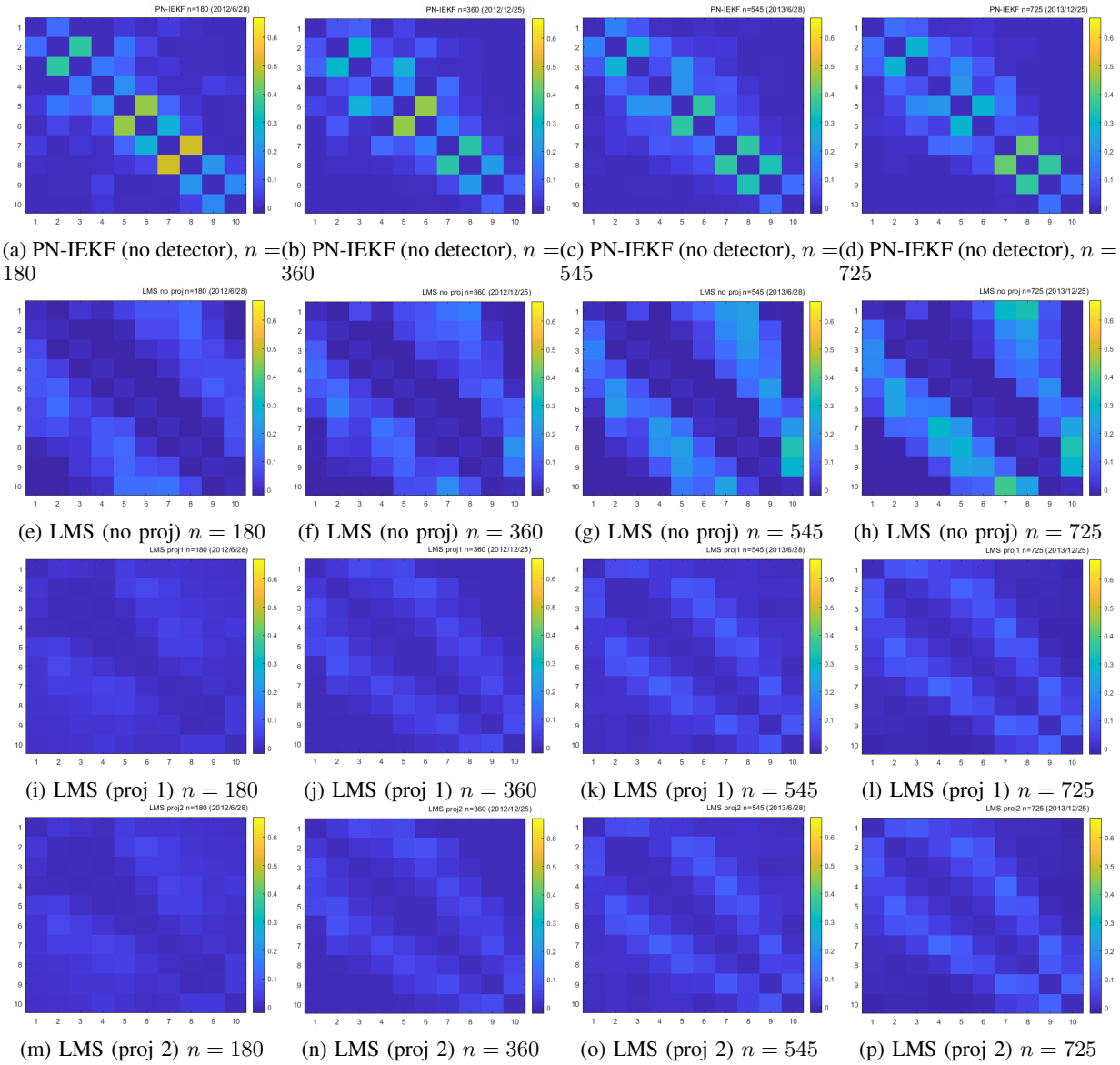
Fig. 31 shows the comparison of time complexity in seconds between the PN-IEKF method and the LMS (proj 2) method, and between the PN-IEFK-VAR and the TIRSO. The time cost of each IEKF/LMS iteration is the average result of 200 Monte Carlo experiments. Due to its high time complexity, the multiplications involved in predicted MSE, corrected MSE, Kalman gain, correction step, and the smoothing step in the PN-IEKF and PN-IEKF-VAR have been executed in parallel by dividing the matrix operands into smaller block matrices of the same size so each block matrix multiplication can be executed in parallel. The time complexity of the parallelized PN-IEKF and parallelized PN-IEKF-VAR are also included in Fig. 31 and as expected, the time complexity of the parallelized methods is much lower than that of their unparallelized counterparts. The parallelization is carried out using 5 PCs, 4 of which has Intel® Core™ i9-11900K, one of them Intel® Core™ i7-9700K. A closer look at the results also reveals that the time complexity of the PN-IEKF is lower than that of the PN-IEKF-VAR, which supports the computational complexity analysis above. In addition, notice in all the graphs that as M increases, the rate of increases in time complexity is higher for PN-IEKF than parallelized PN-IEKF. This behavior also holds true between PN-IEKF-VAR and parallelized PN-IEKF-VAR. This implies it is indeed advantageous to use parallel computing as the number of nodes grows.

V. Conclusion

A novel online graph learning algorithm using proximal Newton-IEKF is proposed for tracking time-varying nonstationary undirected and directed graphs. Results based on NMSD and RE are presented for synthetic and actual data, which prove the efficacy of the proposed technique. This is achieved by incorporating the prior knowledge of the state-space model and the probability distribution of the additive noise into the algorithm. When the second-order statistic of the additive noise is not given, it can be obtained through estimation techniques before carrying out the proposed algorithm. Although the computational and time complexity are relatively higher than those of first-order methods, the proposed methods perform better in terms of RE, which directly relates to the accuracy of the estimated graph adjacency matrix. In addition, a change detector is incorporated into the algorithm to enable the PN-IEKF and PN-IEKF-VAR methods to recover new estimate of the graph faster after abrupt graph transition.

Appendix A: Derivation of the IEKF

The following proof shows the IEKF procedure in Algorithms 1 and 2 can be regarded as the Gauss-Newton iterates [39, pp. 349-351], [40]–[42]. To solve (15) using the Gauss-Newton method, let i denote the previous iteration index and $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$ is the estimate of $\mathbf{s}_W[n-1]$ at ith iteration given



(a) PN-IEKF (no detector), $n = 180$ (b) PN-IEKF (no detector), $n = 360$ (c) PN-IEKF (no detector), $n = 545$ (d) PN-IEKF (no detector), $n = 725$

(e) LMS (no proj) $n = 180$ (f) LMS (no proj) $n = 360$ (g) LMS (no proj) $n = 545$ (h) LMS (no proj) $n = 725$

(i) LMS (proj 1) $n = 180$ (j) LMS (proj 1) $n = 360$ (k) LMS (proj 1) $n = 545$ (l) LMS (proj 1) $n = 725$

(m) LMS (proj 2) $n = 180$ (n) LMS (proj 2) $n = 360$ (o) LMS (proj 2) $n = 545$ (p) LMS (proj 2) $n = 725$

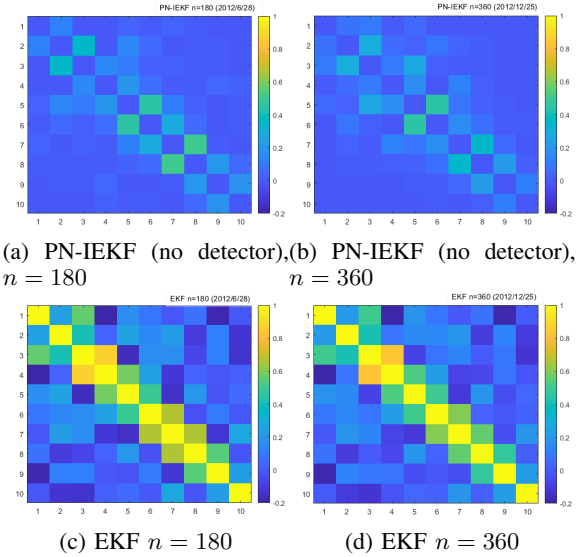
Fig. 29: Color maps of resulting adjacency matrices of PN-IEKF (no detector) and LMS methods at time point $n = 180$, 360, 545 and 725. $\beta = 0.05$ for PN-IEKF (no detector). Experiments are done by using two years of data (2012 to 2013) from 10 stations.

n time samples. This replaces $\widehat{\mathbf{s}}_W[n-1|n]$ in (15). Thus, the objective is written as

$$\widehat{\mathbf{s}}_W[n|n] = \underset{\mathbf{s}_W[n]}{\arg\min}\ \frac{1}{2}\left\| \begin{array}{c} \mathbf{C}_w^{-1/2}\left(\mathbf{x}[n] - \mathbf{H}\mathbf{s}_W[n]\right) \\ \mathbf{M}^{-1/2}[n|n-1]\left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \mathbf{s}_W[n]\right] \end{array} \right\|_2^2 . \tag{34}$$

The Gauss-Newton method finds the solution of (34) using the search direction, $\mathbf{p}_1 = -\nabla^2\ell_1(\mathbf{s}_W[n])^{-1}\nabla\ell_1(\mathbf{s}_W[n])$, where $\nabla^2\ell_1(\mathbf{s}_W[n]) \approx D\mathbf{r}_1(\mathbf{s}_W[n])^T D\mathbf{r}_1(\mathbf{s}_W[n]) = \mathbf{J}_{r_1}^T\mathbf{J}_{r_1}$ and $\nabla\ell_1(\mathbf{s}_W[n]) = D\mathbf{r}_1(\mathbf{s}_W[n])^T\nabla\ell_1(\mathbf{r}_1) = \mathbf{J}_{r_1}^T\mathbf{r}_1$, where $\mathbf{J}_{r_1} \triangleq D\mathbf{r}_1(\mathbf{s}_W[n])$. Hence,

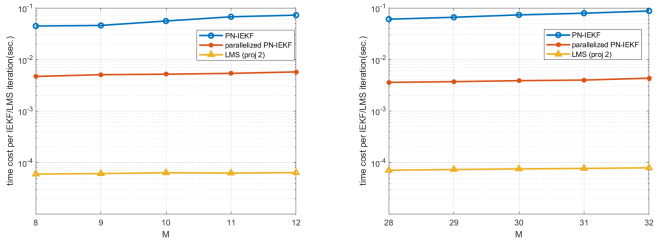
$$\begin{aligned} &\mathbf{J}_{r_1} \\ &= -\begin{bmatrix} \mathbf{C}_w^{-1/2}\mathbf{H} \\ \mathbf{M}^{-1/2}[n|n-1] \end{bmatrix} \text{ and} \\ &\nabla\ell_1(\mathbf{s}_W[n]) \\ &= -\begin{bmatrix} \mathbf{H}^T\mathbf{C}_w^{-1/2} & \mathbf{M}^{-1/2}[n|n-1] \end{bmatrix} \times \\ &\quad \begin{bmatrix} \mathbf{C}_w^{-1}\left(\mathbf{x}[n] - \mathbf{H}\mathbf{s}_W[n]\right) \\ \mathbf{M}^{-1/2}[n|n-1]\left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \mathbf{s}_W[n]\right] \end{bmatrix} \\ &= -\mathbf{H}^T\mathbf{C}_w^{-1}\left(\mathbf{x}[n] - \mathbf{H}\mathbf{s}_W[n]\right) - \\ &\quad \mathbf{M}^{-1}[n|n-1]\left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \mathbf{s}_W[n]\right] \end{aligned}$$



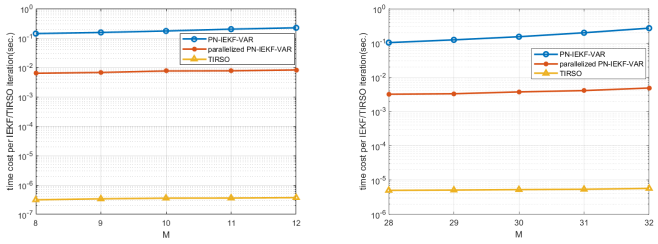
(a) PN-IEKF (no detector), $n = 180$ (b) PN-IEKF (no detector), $n = 360$

(c) EKF $n = 180$ (d) EKF $n = 360$

Fig. 30: Color maps of resulting adjacency matrices of PN-IEKF (no detector) and EKF method at time point $n = 180$ and 360. $\beta = 0.05$ for PN-IEKF (no detector). Experiments are done by using two years of data (2012 to 2013) from 10 stations.



(a) PN-IEKF vs. LMS (proj 2). $M = 8 - 12$. (b) PN-IEKF vs. LMS (proj 2). $M = 28 - 32$.



(c) PN-IEKF-VAR vs. TIRSO. $M = 8 - 12$. (d) PN-IEKF-VAR vs. TIRSO. $M = 28 - 32$.

Fig. 31: (a) Time complexity (sec) vs. number of nodes comparison between the PN-IEKF method, parallelized PN-IEKF and the LMS (proj 2) with $M = 8 - 12$. (b) Similar comparison between the PN-IEKF method, parallelized PN-IEKF and the LMS (proj 2) with $M = 28 - 32$. (c) Similar comparison between PN-IEKF-VAR, parallelized PN-IEKF-VAR, and TIRSO with step size equals $\alpha_n = 1/L$ and $M = 8 - 12$. (d) Similar comparison between PN-IEKF-VAR, parallelized PN-IEKF-VAR, and TIRSO with step size equals $\alpha_n = 1/L$ and $M = 28 - 32$. SNR = 20 dB.

The search direction $\mathbf{p}_1$ is evaluated at the estimate of $\mathbf{s}_W[n]$ at previous iteration, denoted as $\widehat{\mathbf{s}}_W^{(i)}[n|n]$. Hence,

$$\begin{aligned}
&\mathbf{p}_1 \\
&= -\left(\mathbf{J}_{r_1}^T \mathbf{J}_{r_1}\right)^{-1} \mathbf{J}_{r_1}^T \mathbf{r}_1 \\
&= \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \times \\
&\quad \left(\mathbf{H}^T \mathbf{C}_w^{-1} \left(\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]\right) + \right. \\
&\quad \left. \mathbf{M}^{-1}[n|n-1] \left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right]\right).
\end{aligned}$$

The update for $\widehat{\mathbf{s}}_W^{(i)}[n|n]$ using the Gauss-Newton method is derived as

$$\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n|n] \\
=&\widehat{\mathbf{s}}_W^{(i)}[n|n] + \mathbf{p}_1 \\
=&\widehat{\mathbf{s}}_W^{(i)}[n|n] + \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \times \\
&\quad \left(\mathbf{H}^T \mathbf{C}_w^{-1} \left(\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]\right) + \right. \\
&\quad \left. \mathbf{M}^{-1}[n|n-1] \left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right]\right) \\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right) + \\
&\quad \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \times \\
&\quad \left(\mathbf{H}^T \mathbf{C}_w^{-1} \left(\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]\right) + \right. \\
&\quad \left. \mathbf{M}^{-1}[n|n-1] \left[\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right]\right) \\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) + \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \times \\
&\quad \left[\mathbf{H}^T \mathbf{C}_w^{-1} \left(\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]\right) - \right. \\
&\quad \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right) \left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right) \\
&\quad \left. + \mathbf{M}^{-1}[n|n-1] \left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right] \\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) + \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \times \\
&\quad \left[\mathbf{H}^T \mathbf{C}_w^{-1} \left(\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]\right) - \right. \\
&\quad \left. \mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} \left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right] \\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) + \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \mathbf{H}^T \mathbf{C}_w^{-1} \times \\
&\quad \left[\mathbf{x}[n] - \mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n] - \mathbf{H}\left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right) - \widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right].
\end{aligned}$$

Define

$$\mathbf{K}^{(i)}[n] \triangleq \left(\mathbf{H}^T \mathbf{C}_w^{-1} \mathbf{H} + \mathbf{M}^{-1}[n|n-1]\right)^{-1} \mathbf{H}^T \mathbf{C}_w^{-1}.$$

The size of matrix inversion involved in $\mathbf{K}^{(i)}[n]$ can be reduced from $(M+P) \times (M+P)$ to $M \times M$ using the matrix inversion lemma. Thus,

$$\begin{aligned}
&\mathbf{K}^{(i)}[n] \\
=&\left(\mathbf{M}[n|n-1] - \mathbf{M}[n|n-1]\mathbf{H}^T \times \right. \\
&\quad \left. \left(\mathbf{C}_w + \mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)^{-1} \mathbf{H}\mathbf{M}[n|n-1]\right) \mathbf{H}^T \mathbf{C}_w^{-1} \\
=&\mathbf{M}[n|n-1]\mathbf{H}^T \mathbf{C}_w^{-1} - \mathbf{M}[n|n-1]\mathbf{H}^T \times \\
&\quad \left(\mathbf{C}_w + \mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)^{-1} \mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T \mathbf{C}_w^{-1}
\end{aligned}$$

$$
\begin{aligned}
=&\mathbf{M}[n|n-1]\mathbf{H}^T\left(\mathbf{C}_w^{-1}-\right.\\
&\left.\left(\mathbf{C}_w+\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)^{-1}\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\mathbf{C}_w^{-1}\right)\\
=&\mathbf{M}[n|n-1]\mathbf{H}^T\left(\mathbf{C}_w+\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)^{-1}\times\\
&\left[\left(\mathbf{C}_w+\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)\mathbf{C}_w^{-1}-\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\mathbf{C}_w^{-1}\right]\\
=&\mathbf{M}[n|n-1]\mathbf{H}^T\left(\mathbf{C}_w+\mathbf{H}\mathbf{M}[n|n-1]\mathbf{H}^T\right)^{-1}. \qquad (35)
\end{aligned}
$$

Hence, $\widehat{\mathbf{s}}_W^{(i)}[n|n]$ is updated as

$$
\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n|n]\\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)+\\
&\mathbf{K}^{(i)}[n]\left[\mathbf{x}[n]-\mathbf{H}\widehat{\mathbf{s}}_W^{(i)}[n|n]-\mathbf{H}\left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)-\widehat{\mathbf{s}}_W^{(i)}[n|n]\right)\right]\\
=&\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)+\mathbf{K}^{(i)}[n]\left[\mathbf{x}[n]-\mathbf{H}\left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)\right], \qquad (36)
\end{aligned}
$$

which contains both the prediction and the correction steps in the IEKF procedure in Algorithm 1, where $\mathbf{H}\left(\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)$ extracts the signal part in $\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)$, and $\mathbf{K}^{(i)}[n]$ is obtained using (35).

Similarly, (16) can be solved using the Gauss-Newton method. By substituting $\widehat{\mathbf{s}}_W^{(i+1)}[n|n]$ obtained from Eq. (36) for $\widehat{\mathbf{s}}[n|n]$, the objective in (16) is written as

$$
\begin{aligned}
&\widehat{\mathbf{s}}_W[n-1|n]=\\
&\underset{\mathbf{s}_W[n-1]}{\arg\min}\ \frac{1}{2}\left\|\begin{array}{c}\mathbf{C}_{uw}^{-1/2}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}(\mathbf{s}_W[n-1])\right)\\ \mathbf{M}^{-1/2}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\mathbf{s}_W[n-1]\right)\end{array}\right\|_2^2 \qquad (37)
\end{aligned}
$$

The Gauss-Newton method finds the solution of (37) using the search direction, $\mathbf{p}_2=-\nabla^2\ell_2\left(\mathbf{s}_W[n-1]\right)^{-1}\nabla\ell_1\left(\mathbf{s}_W[n-1]\right)$, where $\nabla^2\ell_2\left(\mathbf{s}_W[n-1]\right)\approx D\mathbf{r}_2\left(\mathbf{s}_W[n-1]\right)^T D\mathbf{r}_2\left(\mathbf{s}_W[n-1]\right)=\mathbf{J}_{r_2}^T\mathbf{J}_{r_2}$ and $\nabla\ell_2\left(\mathbf{s}_W[n-1]\right)=D\mathbf{r}_2\left(\mathbf{s}_W[n-1]\right)^T\nabla\ell_2(\mathbf{r}_2)=\mathbf{J}_{r_2}^T\mathbf{r}_2$, where $\mathbf{J}_{r_2}\triangleq D\mathbf{r}_2\left(\mathbf{s}_W[n-1]\right)$. Hence,

$$
\begin{aligned}
&\mathbf{J}_{r_2}\\
=&-\begin{bmatrix}\mathbf{C}_{uw}^{-1/2}\mathbf{F}\\ \mathbf{M}^{-1/2}[n-1|n-1]\end{bmatrix}\ \text{and}\\
&\nabla\ell_2\left(\mathbf{s}_W[n-1]\right)\\
=&-\begin{bmatrix}\mathbf{F}^T\mathbf{C}_{uw}^{-1/2} & \mathbf{M}^{-1/2}[n-1|n-1]\end{bmatrix}\times\\
&\begin{bmatrix}\mathbf{C}_{uw}^{-1/2}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\mathbf{s}_W[n-1]\right)\right)\\ \mathbf{M}^{-1/2}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\mathbf{s}_W[n-1]\right)\end{bmatrix}\\
=&-\mathbf{F}^T\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\mathbf{s}_W[n-1]\right)\right)\\
&-\mathbf{M}^{-1}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\mathbf{s}_W[n-1]\right),
\end{aligned}
$$

where $\mathbf{F}$ denotes the Jacobian matrix of $\boldsymbol{f}(\mathbf{s}_W[n-1])$. The search direction $\mathbf{p}_2$ is evaluated at the estimate of previous iteration, $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$, and $\mathbf{F}^{(i)}$ is the Jacobian of $\boldsymbol{f}(\mathbf{s}_W[n-1])$ evaluated at $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$. Hence,

$$
\begin{aligned}
&\mathbf{p}_2\\
=&-\left(\mathbf{J}_{r_2}^T\mathbf{J}_{r_2}\right)^{-1}\mathbf{J}_{r_2}^T\mathbf{r}_2\\
=&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)+\right.\\
&\left.\mathbf{M}^{-1}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right).
\end{aligned}
$$

Thus, $\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$ is updated as

$$
\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n-1|n]=\widehat{\mathbf{s}}_W^{(i)}[n-1|n]+\mathbf{p}_2\\
=&\widehat{\mathbf{s}}_W^{(i)}[n-1|n]+\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)+\right.\\
&\left.\mathbf{M}^{-1}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)\\
=&\widehat{\mathbf{s}}_W[n-1|n-1]-\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)+\\
&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)+\right.\\
&\left.\mathbf{M}^{-1}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)\\
=&\widehat{\mathbf{s}}_W[n-1|n-1]+\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\left[\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)-\right.\\
&\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)\times\\
&\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\\
&\left.+\mathbf{M}^{-1}[n-1|n-1]\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right]\\
=&\widehat{\mathbf{s}}_W[n-1|n-1]+\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\left[\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left(\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right)-\right.\\
&\left.\left(\mathbf{F}^{(i)}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}\right)\left(\mathbf{s}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right]\\
=&\widehat{\mathbf{s}}_W[n-1|n-1]+\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\times\\
&\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\left[\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)-\right.\\
&\left.\mathbf{F}^{(i)}\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right].
\end{aligned}
$$

Define

$$
\mathbf{S}^{(i)}=\left(\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}\mathbf{F}^{(i)}+\mathbf{M}^{-1}[n-1|n-1]\right)^{-1}\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}.
$$

Using the matrix inversion lemma,

$$
\begin{aligned}
&\mathbf{S}^{(i)}\\
=&\left[\mathbf{M}[n-1|n-1]-\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\times\right.\\
&\left.\left(\mathbf{C}_{uw}+\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\right)^{-1}\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\right]\\
&\times\mathbf{F}^{(i)^T}\mathbf{C}_{uw}^{-1}
\end{aligned}
$$

$$
\begin{aligned}
=&\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\times \\
&\left[\mathbf{C}_{u_W}^{-1}-\right. \\
&\left(\mathbf{C}_{u_W}+\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\right)^{-1}\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1] \\
&\left.\times\mathbf{F}^{(i)^T}\mathbf{C}_{u_W}^{-1}\right] \qquad (38)\\
=&\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\left(\mathbf{C}_{u_W}+\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\right)^{-1}\times \\
&\left[\left(\mathbf{C}_{u_W}+\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\right)\mathbf{C}_{u_W}^{-1}-\right. \\
&\left.\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\mathbf{C}_{u_W}^{-1}\right] \\
=&\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\left(\mathbf{C}_{u_W}+\mathbf{F}^{(i)}\mathbf{M}[n-1|n-1]\mathbf{F}^{(i)^T}\right)^{-1}. \qquad (39)
\end{aligned}
$$

$\widehat{\mathbf{s}}_W^{(i)}[n-1|n]$ is updated as

$$
\begin{aligned}
&\widehat{\mathbf{s}}_W^{(i+1)}[n-1|n] \\
=&\widehat{\mathbf{s}}_W[n-1|n-1]+\mathbf{S}^{(i)}\times \\
&\left[\widehat{\mathbf{s}}_W^{(i+1)}[n|n]-\boldsymbol{f}\left(\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)-\right. \\
&\left.\mathbf{F}^{(i)}\left(\widehat{\mathbf{s}}_W[n-1|n-1]-\widehat{\mathbf{s}}_W^{(i)}[n-1|n]\right)\right], \qquad (40)
\end{aligned}
$$

which is line 11 in Algorithm 1, where $\mathbf{S}^{(i)}$ is evaluated using (39).

REFERENCES

[1] D.I. Shuman *et al.*, "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, vol. 30(3), pp. 83-98, May 2013.

[2] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36(3), pp. 44-63, May 2019.

[3] H.-M. Chiu, C.C. Fung, and A. Ortega, "Graph Learning and Augmentation Based Interpolation of Signal Strength for Location-Aware Communications," *Proc. of the European Signal Processing Conference*, Amsterdam, The Netherlands, Jan. 2021.

[4] J. Richiardi, S. Achard, H. Bunke, and D.V.D. Ville, "Machine learning with brain graphs: Predictive modeling approaches for functional imaging in systems neuroscience," *IEEE Signal Processing Magazine*, vol. 30(3), pp. 58-70, May 2013.

[5] S.M. Plis, M.P. Weisend, E. Damaraju, T. Eichele, A. Mayer, V.P. Clark, T. Lane and V.D. Calhoun, "Effective connectivity analysis of fMRI and MEG data collected under identical paradigms," *Computers in Biology and Medicine*, vol. 41(12), pp. 1156-1165, 2011.

[6] A. Sandryhaila and J.M.F. Moura, "Discrete signal processing on graphs," *IEEE Trans. on Signal Processing*, vol. 61(7), pp. 1644-1656, Apr. 2013.

[7] A. Ortega, P. Frossard, J. Kova?evi?, 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, May 2018, doi: 10.1109/JPROC.2018.2820126.

[8] J. Mei and J.M.F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. on Signal Processing*, vol. 65(8), pp. 2077-2092, 2017.

[9] S. Segarra, A.G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. on Signal and Information Processing over Networks,* vol. 3(3), pp. 467-483, Sep. 2017.

[10] V. Kalofolias, "How to learn a graph from smooth signals," *AISTATS*, 2016.

[11] X. Dong, D. Thanou, P. Frossard and P. Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," *in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3736-3740, 2015, doi: 10.1109/ICASSP.2015.7178669.

[12] X. Dong, D. Thanou, P. Frossard and P. Vandergheynst, "Learning Laplacian Matrix in Smooth Graph Signal Representations," *in IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160-6173, 2016, doi: 10.1109/TSP.2016.2602809.

[13] H. Rue and L. Held, "Gaussian Markov Random Fields: Theory and Applications (1st ed.)." Chapman and Hall/CRC, 2005, https://doi.org/10.1201/9780203492024

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9(3), pp. 432-441, 2008.

[15] O. Banerjee and L. El Ghaoui, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485-516, Mar. 2008.

[16] H.E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE. Journal on Selected Topics on Signal Processing*, vol. 11(6), pp. 825-841, Sep. 2017.

[17] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Processing*, Brighton, UK, May 2019.

[18] V. Kalofolias, A. Loukas, D. thanou, and P. Frossard, "Learning time varying graphs," *Proc. of the IEEE Intl. on Acoustics, Speech and Signal Processing*, pp. 2826-2830, 2017.

[19] S. Vlaski *et al.*, "Online graph learning from sequential data," *Proc. of the Data Science Workshop*, Lausanne, Switzerland, Jun. 2018.

[20] F. Chung, "The heat kernel as the pagerank of a graph," *Proceedings of the National Academy of Sciences*, vol. 104(50), pp. 19735-19740, 2007.

[21] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals," *Proc. of the IEEE Data Science Workshop*, Minneapolis, MN, USA, Jul. 2019.

[22] M. Gu and S.C. Eisenstat, "A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem," *SIAM Journal on Matrix Analysis and Applications*, vol. 15(4), pp. 1266-1276, 1994.

[23] R. Shafipour, S. Segarra, A.G. Marques, G. Mateos, "Identifying the topology of undirected networks from diffused non-stationary graph signals," *IEEE Open Journal of Signal Processing*, vol. 2, Mar. 2021.

[24] R. Shafipour and G. Mateos, "Online proximal gradient for learning graphs from streaming signals," *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 865-869.

[25] M. Moscu, R. Borsoi, and C. Richard, "Online graph topology inference with kernels for brain connectivity estimation," *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, May 2020.

[26] D. Jin, J. Chen, C. Richard and J. Chen, "Adaptive Parameters Adjustment for Group Reweighted Zero-Attracting LMS," *Proc. of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4294-4298, 2018, doi: 10.1109/ICASSP.2018.8462004.

[27] S.S. Saboksayr, G. Mateos, and M. Cetin, "Online graph learning under smoothness priors," *arXiv:2103.03762v1*, Mar. 2021.

[28] S.S. Saboksayr, G. Mateos, and M. Cetin, "Online discriminative graph learning from multi-class smooth signals," *arXiv preprint arXiv:2101.00184*, 2021.

[29] B. Zaman, L.M. Lopez-Ramos, D. Romero, and B. Beferull-Lozano, "Online topology estimation for vector autogressive processes in data networks," *2017 IEEE 7th Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Curacao, Dec. 2017.

[30] B. Zaman, L.M.L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Trans. on Signal Processing*, vol. 69, pp. 210-225, 2021.

[31] A. Natali, M. Coutino, E. Isufi and G. Leus, "Online Time-Varying Topology Identification Via Prediction-Correction Algorithms," *Proc. of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5400-5404, 2021.

[32] A. Natali, E. Isufi, M. Coutino and G. Leus, "Online Graph Learning From Time-Varying Structural Equation Models," *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 1579-1585, 2021.

[33] A. Natali, E. Isufi, M. Coutino and G. Leus, "Learning time-varying graphs from online data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212-228, May. 2022.

[34] M. Ramezani-Mayiami, "Joint graph learning and signal recovery via kalman filter for multivariate auto-regressive process," *Proc. of the 26th European Signal Processing Conference*, Rome, Italy, Sep. 2018.

[35] M. Ramezani-Mayiami, "Joint topology learning and graph signal recovery via kalman filter in causal data processes," *Proc. of the IEEE Intl. Workshop on Machine Learning for Signal Processing*, Aalborg, Denmark, Sep. 2018.

[36] R.P. Wishner, J.A. Tabaczynski and M. Athans, "On the estimation of the state of noisy nonlinear multivariable systems," *Proc. of the IFAC Sym. on Multivariable Control Systems*, Dusseldorf, West Germany, Oct. 1968.

[37] R.P. Wishner, J.A. Tabaczynski and M. Athans, "A comparison of three non-linear filters," *Automatica*, vol. 5, pp. 487-496, 1969.

[38] J.D. Lee, Y. Sun and M.A. Saunders, "Proximal Newton-type methods for minimizing composite functions," *SIAM Journal on Optimization*, vol. 24(3), pp. 1420-1443, 2014.

[39] A.J. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, 1970.

[40] B.M. Bell and F.W. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Trans. on Automatic Control*, vol. 38(2), pp. 294-297, Feb. 1993.

[41] B.M. Bell, "The iterated Kalman smoother as a Gauss-Newton method," *SIAM Journal on Optimization*, vol. 4(3), pp. 626-636, Aug. 1994.

[42] M.A. Skoglund, G. Hendeby and D. Axehill, "Extended kalman filter modifications based on an optimization view point," *18th Intl. Conf. on Information Fusion*, pp. 1856-1861, 2015.

[43] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1(3), pp. 123-231, 2013.

[44] I. Daubechies, M. Defrise and C.D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, pp. 1413-1457, 2004.

[45] R.K. Mehra and J. Perchon, "An innovative approach to fault diagnosis in dynamic system," *Automatica*, vol. 7, pp. 637-640, 1971.

[46] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, John Wiley, New York, 1958

[47] P. Nathanaël, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst and D.K. Hammond, "GSPBOX: A toolbox for signal processing on graphs." *Arxiv e-print arXiv:1408.5781v2*, 2014.

[48] A. H. Al-Mohy and N.J. Higham, "Improved inverse scaling and squaring algorithms for the matrix logarithm," *SIAM Journal on Scientific Computing*, 34(4), pp. C153?C169, 2012

[49] A. H. Al-Mohy, N.J. Higham and S.D. Relton, "Computing the Frechet derivative of the matrix logarithm and estimating the condition number," *SIAM Journal on Scientific Computing*, 35(4), pp. C394?C410, 2013

[50] DEAP, "A dataset for emotion analysis using physiological signals," *https://www.eecs.qmul.ac.uk/mmv/datasets/deap/*.

[51] L.E. Ismail and W. Karwowski, "A graph theory-based modeling of functional brain connectivity based on EEG: A systematic review in the context of neuroergonomics," *IEEE Access*, vol. 8, pp. 155103-155135, Aug. 2020.

[52] *National Centers for Environmental Information*. [Online]. Available: *https://www.ncei.noaa.gov/access/search/data-search/global-summary-of-the-day*