Received July 15, 2021, accepted August 16, 2021, date of publication August 31, 2021, date of current version September 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3109288

Joint Vertex-Time Filtering on Graphs With Random Node-Asynchronous Updates

OGUZHAN TEKE, (Member, IEEE), AND PALGHAT P. VAIDYANATHAN, (Life Fellow, IEEE)
Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA

Corresponding author: Oguzhan Teke (oteke@caltech.edu)

This work was supported in part by the Office of Naval Research under Grant N00014-18-1-2390 and Grant N00014-21-1-2521, and in part by the California Institute of Technology.

ABSTRACT In graph signal processing signals are defined over a graph, and filters are designed to manipulate the variation of signals over the graph. On the other hand, time domain signal processing treats signals as time series, and digital filters are designed to manipulate the variation of signals in time. This study focuses on the notion of vertex-time filters, which manipulates the variation of a time-dependent graph signal both in the time domain and graph domain simultaneously. The key aspects of the proposed filtering operations are due to the random and asynchronous behavior of the nodes, in which they follow a collect-compute-broadcast scheme. For the analysis of the randomized vertex-time filtering operations, this study first considers the random asynchronous variant of linear discrete-time state-space models, in which each state variable gets updated randomly and independently (and asynchronously) in every iteration. Unlike previous studies that analyzed similar models under certain assumptions on the input signal, this study considers the model in the most general setting with arbitrary time-dependent input signals, which lay the foundations for the vertex-time graph filtering operations. This analysis shows that exponentials continue to be eigenfunctions in a statistical sense in spite of the random asynchronous nature of the model. This study also presents the necessary and sufficient condition for the mean-squared stability and shows that stability of the underlying state transition matrix is neither necessary nor sufficient for the mean-squared stability of the randomized asynchronous recursions. Then, the proposed filtering operations are proven to be mean-square stable if and only if the filter, the graph operator and the update probabilities satisfy a certain condition. The results show that some unstable vertex-time graph filters (in the synchronous case) can be implemented in a stable manner in the presence of randomized asynchronicity, which is also demonstrated by numerical examples.

INDEX TERMS Autonomous networks, asynchronous updates, randomized iterations, vertex-time filters.

I. INTRODUCTION

Linear time-invariant discrete-time systems are well studied mathematical models that find applications in wide range of different areas ranging from mathematical finance to implementation of digital filters [1], [2]. Although such models are especially useful for analyzing (and controlling) dynamical systems that evolve in time, the mathematical models are useful in numerical linear algebra problems as well. An example is the ‘‘power method’’ that can extract the dominant eigenvector of the transition matrix, whose notable application is the PageRank algorithm used in search engines for ranking web pages [3].

State-space models are studied also in the field of graph signal processing [4], [5], in which the state transition matrix is assumed to model the underlying graph structure and state variables are interpreted as signals held by the nodes of the graph. In this setting an iteration of the state-space model is equivalent to the nodes communicating with their neighbors. With this formalism, state recursions are utilized for distributed implementation of polynomial (FIR) graph filters [6]–[11] as well as rational (IIR) ones [12]–[15].

Despite their distributed implementation on graphs, the use of standard state-space models requires the nodes to exchange data simultaneously, or wait for each other, before starting

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

the next round of communication. This type of implementation clearly requires a synchronization over the whole network, which becomes an important limitation when the size of the network is large, e.g. distributed large-scale graph processing frameworks [16]–[18], or when the network has autonomous behavior without a centralized control. In order to eliminate the need for such a synchronization, the recent studies [19]–[25] considered randomized asynchronous variants of the state recursions. These studies guaranteed the convergence of the randomized updates on graphs under various assumptions on the underlying state transition matrix (e.g. graph operator), the input signal, and the update probabilities.

The paper has several new contributions, in addition to its tutorial and review value. All derivations are presented in detail. The main novelty, in the context of other recent work, is described below.

- The recent work in [19]–[25] considered only input signals that are constant in time although they depend on the vertex of the graph. This paper extends this to the case of time-dependent signals. Thus, we propose a random asynchronous method that achieves filtering of time-dependent graph signals both in time and over a graph simultaneously.
- The paper shows that even in the presence of randomized asynchronicity, exponentials remain eigenfunctions of linear systems in a statistical sense. Therefore, the notion of frequency response remains valid, and it can be utilized to manipulate time-dependent input signals.
- Although joint graph (vertex)-time filters are well-studied problems in graph signal processing [26]–[31], the main advantage of the filtering method proposed in this study is due to its flexibility to allow the nodes of a graph to behave randomly and asynchronously in a collect-compute-broadcast scheme without requiring any centralized control.
- This study analyzes the mean behavior and the mean-squared stability of the proposed filtering method rigorously from the view-point of Markov-jump switching systems [32] and shows that randomized asynchronicity can be utilized to induce stability into a system.

Technical contributions of this study will be detailed in the next section.

A. OUTLINE AND CONTRIBUTIONS OF THE PAPER

This study consists of two main parts. The first part (Sections II - IV) explores the behavior of random asynchronous linear state-space models with time-dependent inputs. In particular, Section II investigates the notion of "frequency response" by showing that, in spite of the random asynchronous nature of the iterations, exponentials continue to be eigenfunctions of these systems in a *statistical average sense* (Proposition 1). So, a random asynchronous system can be treated as a time-invariant system in an average sense. Section III focuses on the second order statistics of a randomized system and provides the necessary and sufficient condition for the mean-squared stability of the randomized recursions (Corollary 1 to Theorem 1). This result shows also that an unstable system (in the synchronous world) may get stable with randomized asynchronicity. Although similar conclusions were observed in the special cases of zero input [19] and time-independent input [23], Section III of this study analyzes this phenomena in its full generality and precisely characterizes the error in the state variables due to randomization and time variation in the input signal. Section IV shows that stability conditions in the synchronous and asynchronous settings do not imply each other in general (Lemma 4). The main motivation of the first part is to provide a mathematical framework for the analysis of the proposed vertex-time filtering operations in later sections. A preliminary version of these results was presented in [33], and some of these results appeared partially in the thesis [34].

The second part of this study (Sections V-VIII) focuses on vertex-time filters with node-asynchronous updates. In particular, Section V introduces the node-asynchronous joint vertex-time filtering operations and describes how they can be modeled as random asynchronous state recursions. Based on the analysis done in the first part, Section V-A presents the necessary and sufficient condition for the mean-squared stability of the filtering operations. Section VI visits the notion of discrete-time graph Fourier transform, which represents a given time-varying graph signal in terms of eigenvectors of the graph operator, and the complex exponential. Based on this notion Section VII presents the graph-frequency response of a filter in the presence of randomized node asynchronicity. Section VIII presents the simulation results of the proposed filtering operations and verifies the theoretical developments.

B. NOTATION

We will use $\mathbb{P}[\cdot]$ and $\mathbb{E}[\cdot]$ to denote the probability and expectation, respectively. For a matrix X, we will use X^* and X^{H} to denote its conjugate and conjugate transpose, respectively, and $\rho(X)$ to denote its spectral radius (the largest eigenvalue in magnitude). For a matrix $X \in \mathbb{C}^{N\times M}$ we will use $\mathrm{vec}(X) \in \mathbb{C}^{NM}$ to denote a vector obtained by cascading the columns of X, and we will use $\mathrm{vec}^{-1}(\cdot)$ for the inverse vectorization operation such that $\mathrm{vec}^{-1}(\mathrm{vec}(X)) = X$ holds true for any matrix X. We will use $\prec$ and $\preceq$ to denote positive definite and positive semi-definite ordering, respectively. We will use $\odot$ to denote the Hadamard product, and $\otimes$ to denote the Kronecker product.

We will use $\mathcal{T}$ to denote a subset of $\{1, \cdots, N\}$, and $P_{\mathcal{T}} \in \mathbb{R}^{N\times N}$ to denote a diagonal matrix that has value 1 only at the indices specified by the set $\mathcal{T}$. That is,

$$P_{\mathcal{T}} = \sum_{i\in\mathcal{T}} e_i\, e_i^{\mathrm{H}} \quad \text{and} \quad \mathrm{tr}(P_{\mathcal{T}}) = |\mathcal{T}|, \tag{1}$$

where $e_i \in \mathbb{R}^N$ is the i^{th} standard vector that has 1 at the i^{th} index and 0 elsewhere. We will use I_M to denote the identity matrix of size M.

II. RANDOMIZED LINEAR SYSTEMS WITH EXPONENTIAL INPUTS

Consider a discrete time-invariant system with R inputs, P outputs, and N state variables, whose state-space description is given as follows:

$$x[k+1] = A\,x[k] + B\,u[k] + w[k], \tag{2}$$
$$y[k] = C\,x[k] + D\,u[k], \tag{3}$$

where $x[0] \in \mathbb{C}^N$ is the initial state vector (initial condition), and $w[k] \in \mathbb{C}^N$ is the noise term with the following statistics:

$$\mathbb{E}\big[w[k]\big] = \mathbf{0}, \quad \mathbb{E}\big[w[k]\, w^{\mathrm{H}}[k]\big] = \mathbf{\Gamma}, \tag{4}$$

where $\mathbf{\Gamma}$ is allowed to be non-diagonal. We also assume that noise is white, i.e., $\mathbb{E}[w[k]\, w^{\mathrm{H}}[s]] = \mathbf{0}$ when $k \neq s$.

The matrices in the state-space model in (2) have the following dimensions:

$$A \in \mathbb{C}^{N\times N}, \quad B \in \mathbb{C}^{N\times R}, \quad C \in \mathbb{C}^{P\times N}, \quad D \in \mathbb{C}^{P\times R}, \tag{5}$$

where A is referred to as the *state-transition matrix*, and the columns of the matrices B and D will be denoted as follows:

$$B = [B_1 \ \cdots \ B_R], \quad D = [D_1 \ \cdots \ D_R]. \tag{6}$$

We further assume that the input signal $u[k]$ consists of R exponential signals in the following form:

$$u[k] = \begin{bmatrix} \alpha_1^k & \cdots & \alpha_R^k \end{bmatrix}^{\mathrm{T}}, \tag{7}$$

where α_i's are assumed to be distinct without loss of generality. Furthermore, we always assume that

$$|\alpha_i| \leq 1, \quad \forall\, 1 \leq i \leq R, \tag{8}$$

so that $u[k]$ stays bounded throughout the iterations. While exponential inputs may seem restrictive, they form the basis for more general practical signals, making this section useful. Later in Section VI, we will discuss how we can represent an arbitrary time-varying graph signal in terms of eigenvectors of the graph operator and complex exponentials.

In the noise free case, i.e., $\mathbf{\Gamma} = \mathbf{0}$, it is well-known from linear system theory that the output vector $y[k] \in \mathbb{C}^P$ in (3) can be written as follows:

$$y[k] = y^{\mathrm{ss}}[k] + y^{\mathrm{tr}}[k], \tag{9}$$

where $y^{\mathrm{ss}}[k]$ denotes the steady-state component, and $y^{\mathrm{tr}}[k]$ denotes the transient component that are given as follows:

$$y^{\mathrm{ss}}[k] = \sum_{i=1}^{R} H_i(\alpha_i)\, \alpha_i^k, \quad y^{\mathrm{tr}}[k] = C\,A^k \left(x[0] - x^{\mathrm{ss}}[0]\right). \tag{10}$$

where $H_i(z) \in \mathbb{C}^P$ denotes the transfer function that relates the i^{th} input to the output, which is given as follows:

$$H_i(z) = D_i + C\left(zI - A\right)^{-1} B_i. \tag{11}$$

We also note that the term $x^{\mathrm{ss}}[0]$ in (10) is given as follows:

$$x^{\mathrm{ss}}[0] = \sum_{i=1}^{R} \left(\alpha_i I - A\right)^{-1} B_i. \tag{12}$$

It is clear from (10) that when the state-transition matrix A is a stable matrix, i.e., the following holds true:

$$\rho(A) < 1, \tag{13}$$

then the transient component $y^{\mathrm{tr}}[k]$ converges to zero as the iterations progress leaving only the steady-state component $y^{\mathrm{ss}}[k]$ in the output signal. In fact, stability of A is also necessary for the transient part to converge to zero, which is a well-known result from linear system theory [2].

A. THE RANDOM ASYNCHRONOUS MODEL

Distributed processing of data over networks has been studied extensively, and they find applications in different problem formulations ranging from distributed sensor localization [35]–[38] and controlling network of agents [39], [40] to opinion dynamics [41] and PageRank computations [42], [43]. One particular purpose of distributed algorithms is to obtain a consensus among the agents of the underlying network, and the value of the consensus is designed to be the optimal solution of the objective function of interest [44]–[46]. These algorithms generally have robust behavior against the changes in the network topology, communication failures, delays, asynchronous behavior of the agents, and so on [47]. The literature on distributed algorithms is vast, and we refer to [48] for a comprehensive introduction to the topic.

In this study, we will consider the state recursions in (2) in the context of graph signal processing, where the matrix A will represent a graph (connectivity structure) and state variables are associated with the nodes of the graph. As a result, recursions in the form (2) correspond to a synchronous data exchange between the neighboring nodes, and they lay a foundation for polynomial and rational graph filters [6]–[15]. As one goal of this study is to provide a mathematical framework for the node-asynchronous graph filtering operations (see Section V), this section will elaborate on the following *randomized asynchronous* variant of the state-space model:

$$x_i[k+1] = \begin{cases} \big(A\,x[k] + B\,u[k] + w[k]\big)_i, & \text{w.p.} \quad p_i, \\ x_i[k], & \text{w.p.} \quad 1 - p_i, \end{cases} \tag{14}$$
$$y[k] = C\,x[k] + D\,u[k], \tag{15}$$

where p_i denotes *the update probability of the i^{th} state variable*. The model (14) is very similar to the standard state-space model in (3) except the fact that a state variable only a random subset of indices are updated in every iteration, and the remaining indices stay the same. We will use P to denote the diagonal matrix consisting of the index selection probabilities. More precisely,

$$P = \operatorname{diag}\Big(\begin{bmatrix} p_1 & p_2 & \cdots & p_N \end{bmatrix}\Big) \in \mathbb{R}^{N\times N}. \tag{16}$$

It is assumed that P satisfies $\mathbf{0} \prec P \preceq I$, where the positive definiteness follows from the fact that no index should be left out permanently during the updates of (14). See Section IV for further details. Additionally, we note that $\mathrm{tr}(P)$ denotes the number of indices updated per iteration on average.

We note that the model (14) appears as a pull-like algorithm when the state-transition matrix A is interpreted as a graph, in which a node retrieves data from its incoming neighbors. Since the model requires the node to collect data from all of its neighbors, it can be argued that the model is not truly asynchronous. However, in practical implementations we will assume that nodes hold a buffer and they communicate with each other asynchronously in a collect-compute-broadcast scheme. These details will be discussed later in Section V.

B. FREQUENCY RESPONSE IN THE MEAN

Due to the random updates of the state variables it is clear from (14) that the state vector $x[k]$ is a random vector. So, the state vector will not have an exponential behavior exactly even when the input is a simple exponential ($R = 1$ in (7)). Nevertheless, we will show that $x[k]$ still behaves like a sum of exponential signals *in a statistical averaged sense*:

Proposition 1: Assume that the randomized asynchronous state recursions in (14) *are initialized independently and randomly. Then, the expectation of the state vector in* (14) *is as follows*:

$$\mathbb{E}\big[x[k]\big] = x^{\mathrm{ss}}[k] + x^{\mathrm{tr}}[k], \tag{17}$$

where

$$x^{\mathrm{ss}}[k] = \sum_{i=1}^{R} (\alpha_i\, I - \overline{A})^{-1}\, \overline{B}_i\, \alpha_i^k, \tag{18}$$

$$x^{\mathrm{tr}}[k] = \overline{A}^k \left(\mathbb{E}\big[x[0]\big] - x^{\mathrm{ss}}[0]\right), \tag{19}$$

and

$$\overline{A} = I + P\,(A - I), \qquad \overline{B}_i = P\, B_i. \tag{20}$$

Proof: See Appendix B. ■

Here, $\overline{A}$ will be referred to as *the average state-transition matrix*, and $\overline{B}$ *the average input matrix*. We can also represent $x^{\mathrm{ss}}[k]$ in (18) as follows:

$$x^{\mathrm{ss}}[k] = \sum_{i=1}^{R} x_i^{\mathrm{ss}}[0]\, \alpha_i^k \quad \text{where} \quad x_i^{\mathrm{ss}}[0] = (\alpha_i\, I - \overline{A})^{-1}\, \overline{B}_i, \tag{21}$$

which will be useful later in the paper.

As an immediate corollary to Proposition 1, the expectation of the random output $y[k]$ can be written as follows:

$$\mathbb{E}\big[y[k]\big] = y^{\mathrm{ss}}[k] + y^{\mathrm{tr}}[k], \tag{22}$$

where

$$y^{\mathrm{ss}}[k] = \sum_{i=1}^{R} \overline{H}_i(\alpha_i)\, \alpha_i^k, \quad y^{\mathrm{tr}}[k] = C\, \overline{A}^k \left(\mathbb{E}[x[0]] - x^{\mathrm{ss}}[0]\right), \tag{23}$$

and

$$\overline{H}_i(z) = D_i + C\, \left(z\, I - \overline{A}\right)^{-1} \overline{B}_i. \tag{24}$$

Regarding the form in (22) we first note that the terms $y^{\mathrm{ss}}[k]$ and $y^{\mathrm{tr}}[k]$ are deterministic quantities, and the expectation is with respect to the random selection of the indices, the input noise, and the random selection of the initial condition. Furthermore, (22) shows that the random output vector $y[k]$ behaves the same as its deterministic counterpart (9) in expectation. That is, $\mathbb{E}[y[k]]$ can be decomposed into steady-state and transient parts similar to (9). Therefore, the quantity $\overline{H}_i(z)$ given in (24) can be regarded as the "transfer function" from the i^{th} input to the output in the *expectation sense*.

It is clear from (23) that as long as the average state transition matrix $\overline{A}$ is stable, i.e., the following holds true:

$$\rho(\overline{A}) < 1, \tag{25}$$

the component $y^{\mathrm{tr}}[k]$ converges to zero irrespective of the observation matrix C and the statistics of the initial condition. Thus, the condition (25) is both necessary and sufficient for $\mathbb{E}[y[k]]$ to behave like a sum of exponentials, that is,

$$\lim_{k\to\infty} \mathbb{E}\big[y[k] - y^{\mathrm{ss}}[k]\big] = \mathbf{0}. \tag{26}$$

This shows that when (25) is satisfied an exponential input results in an exponential output *in expectation* even with the randomized asynchronous state recursions.

C. A NUMERICAL EXAMPLE

In order to demonstrate the behavior of the random vector $y[k]$, we consider the following state-space model with $N = 4$ state variables, $R = 1$ input, and $P = 1$ output:

$$A = \frac{1}{10}\begin{bmatrix} -4 & -1 & 2 & -6 \\ 4 & -6 & -5 & 3 \\ 2 & -2 & 7 & 2 \\ 5 & 9 & -3 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \end{bmatrix}, \; C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^{\mathrm{T}}, \; D = 0, \tag{27}$$

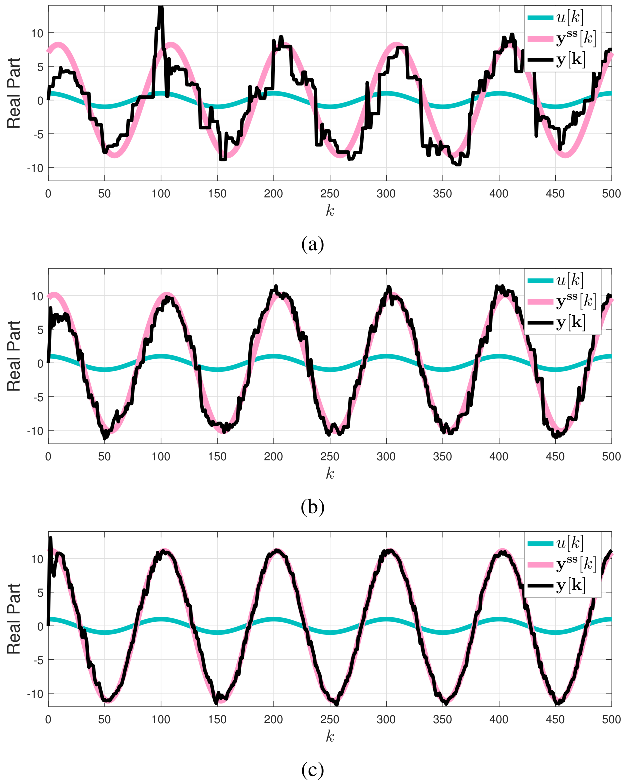
where we point out that the matrix A is *not* stable since $\rho(A) \approx 1.0441$. As we shall discuss later in Section III, stability of the randomized asynchronous state recursions does not require stability of the state-transition matrix in general.

In the following numerical example we assume that $P = p\,I$, i.e., all nodes have the update probability p and assume that $\mathbf{\Gamma} = \mathbf{0}$. Furthermore, we assume that the input signal has the following complex exponential form:

$$\alpha = e^{j2\pi/100} \quad \Longrightarrow \quad u[k] = e^{j2\pi k/100}. \tag{28}$$

In Figure 1 we visualize *a realization* of the output signal $y[k]$ together with the steady-state component $y^{\mathrm{ss}}[k]$ as well as the input signal $u[k]$ for three different update probabilities, namely, $p \in \{0.1,\ 0.3,\ 0.6\}$. The figure shows only the real part of the signals for convenience.

From Figure 1 it is clear that the random vector $y[k]$ is not a complex exponential in a strict sense, yet it "behaves like" one. We also note that $y[k]$ has the same "frequency" (in an average sense) as the input signal irrespective of the update probabilities.



(a)

(b)

(c)

FIGURE 1. A realization of the output signal with the state-space model in (27), the frequency in (28), and the probabilities (a) $p=0.1$, (b) $p=0.3$, (c) $p=0.6$.

Figure 1 shows also that the response of the random asynchronous system depends on the update probabilities, which is also apparent from the expression in (24). In fact, the response of the random asynchronous updates running on a system denoted with (A, B, C, D) can be represented in an average sense as the response of a synchronous system $(\overline{A}, \overline{B}, C, D)$. As a result, different update probabilities result in different "frequency responses" while leaving the output frequency unchanged.

III. SECOND ORDER STABILITY OF THE STATE VARIABLES

Results in Section II-B together with Figure 1 show that the random vector $y[k]$ behaves like $y^{\mathrm{ss}}[k]$ in expectation. However, in order to interpret the response of the randomized asynchronous system meaningfully, the random variable $y[k]$ must be ensured to have a "finite amount of deviation" from $y^{\mathrm{ss}}[k]$. In this regard we consider the following quantities:

$$r[k] = y[k] - y^{\mathrm{ss}}[k], \qquad q[k] = x[k] - x^{\mathrm{ss}}[k], \tag{29}$$

where $r[k]$ will be referred to as the error in the output, and $q[k]$ will be referred to as the error in the state variables. It is readily verified that the error terms in (29) are related with each other through the output matrix C as follows:

$$r[k] = C\, q[k]. \tag{30}$$

In the rest of this section we will focus on the term $q[k]$, i.e., study the internal stability of the random asynchronous system. More precisely, we consider *the error (auto) correlation matrix of the state variables* defined as follows:

$$Q[k] = \mathbb{E}\Big[q[k]\, q^{\mathrm{H}}[k]\Big] \in \mathbb{C}^{N\times N}. \tag{31}$$

At this point it is very important to emphasize that the error correlation matrix $Q[k]$ does not converge to zero in general even when no noise present in the system. More interestingly, $Q[k]$ may not converge to a point at all; rather, it may show an oscillatory behavior. This is an inherent side effect of the randomized asynchronicity, which will be discussed in detail. As a result, we will consider the conditions under which $Q[k]$ stays bounded (or, equivalently $q[k]$ stays bounded in the mean-squared sense). In this paper "second order stability" is synonymous to "boundedness of the matrix $Q[k]$."

We also note that stability of the matrix $\overline{A}$ merely ensures the first order stability of the error term. That is,

$$\rho(\overline{A}) < 1 \qquad \Longleftrightarrow \qquad \lim_{k\to\infty} \mathbb{E}\big[q[k]\big] = \mathbf{0}. \tag{32}$$

On the other hand, stability of $\overline{A}$ is not sufficient to ensure the second order stability of $q[k]$. (See Lemma 3 in Section IV-A).

In what follows, we will first study how the error correlation matrix $Q[k]$ evolves throughout the iterations (Theorem 1). Based on this result, we will consider the necessary and sufficient condition for the boundedness of the error correlation matrix (Corollary 1). Later, we will provide a sufficiency condition for the second order stability (Corollary 2).

We would like to note that the mean and mean-squared behavior of random graph processes have been of interest in recent years. For example, the study [49] considers the case of time varying graphs and considers the mean controllability of graph processes while providing mean-squared error analysis. Similarly, the study [50] studies graph processes from tracking viewpoint and provides mean and mean-squared analysis of Kalman filtering over graphs. Although the randomized model in (14) can be interpreted as a time-varying graph, we would like to note that time-varying graph framework is not directly applicable to the model here. Since the analysis of vertex-time filtering operations (to be described in Section V) will be based on the model in (14), this section will provide a mean-squared analysis tailored for the model.

We would like to also note that convergence (and stability) of product of matrices are well-studied problems in literature [51]–[57], and stability of the randomized model (14) can be analyzed from the viewpoint of these results. However, these frameworks often times either impose additional constraints on the matrix A (such as [51] requiring A to be nonnegative, or [56] requiring A to be positive definite), or they are too restrictive (involving the joint spectral radius) to explain the effect of randomization observed in Figure 1. In this study, we focus on the second order statistics of the randomized model, and the analysis will be based on the Markov jump system viewpoint [32]. We will elaborate further on this in Section III-C.

A. EVOLUTION OF THE ERROR CORRELATION MATRIX

We start by consider the following matrix valued function:

$$\varphi(X) = \overline{A}\,X\,\overline{A}^{\mathrm{H}} + \Big((\overline{A}-I)\,X\,(\overline{A}-I)^{\mathrm{H}}\Big) \odot \big(P^{-1} - I\big), \qquad (33)$$

where $\overline{A}$ is as in (20), and $\odot$ denotes the Hadamard product.

Note that the function $\varphi(\cdot)$ is defined through the average state-transition matrix $\overline{A}$ as well as the update probabilities P. Derivation of the function $\varphi(\cdot)$ will be described in Section III-C.

The importance of the function $\varphi(\cdot)$ follows from the fact that it governs the evolution of the error correlation matrix through the iterations. This is presented precisely as follows:

Theorem 1: The error correlation matrix of the state variables evolves according to the following recursion:

$$\begin{aligned} Q[k+1] = \varphi(Q[k]) + P\,\mathbf{\Gamma}\,P + \mathbf{\Gamma} \odot \big(P - P^2\big) \\ + \Re\Big\{\big(\boldsymbol{\delta}[k] + 2(\overline{A}-I)\,x^{\mathrm{tr}}[k]\big)\,\boldsymbol{\delta}^{\mathrm{H}}[k]\Big\} \odot \big(P^{-1} - I\big), \end{aligned} \qquad (34)$$

where $\Re\{\cdot\}$ denotes the real part of its argument, and the deterministic vector $\boldsymbol{\delta}[k] \in \mathbb{C}^N$ is defined as follows:

$$\boldsymbol{\delta}[k] = x^{\mathrm{ss}}[k+1] - x^{\mathrm{ss}}[k] = \sum_{i=1}^{R} (\alpha_i\,I - \overline{A})^{-1}\,\overline{B}_i\,\alpha_i^k\,(\alpha_i - 1). \qquad (35)$$

Proof: See Appendix C. ■

In order to study the behavior of the recursion in (34), we first represent the linear map $\varphi(\cdot)$ as a matrix-vector product by vectorizing (33). That is,

$$\varphi(X) = \mathrm{vec}^{-1}\Big(S\ \mathrm{vec}(X)\Big), \qquad (36)$$

and the matrix $S \in \mathbb{C}^{N^2 \times N^2}$ is as follows:

$$S = \overline{A}^{*} \otimes \overline{A} + \Big((P^{-1} - I) \otimes I\Big)\,J\,\Big((\overline{A}^{*} - I) \otimes (\overline{A} - I)\Big), \qquad (37)$$

where A^* denotes the element-wise conjugate (not conjugate transpose) of the matrix, and J is a diagonal matrix as follows:

$$J = \sum_{i=1}^{N} (e_i\,e_i^{\mathrm{H}}) \otimes (e_i\,e_i^{\mathrm{H}}) \in \mathbb{R}^{N^2 \times N^2}. \qquad (38)$$

Since the error correlation matrix evolves according to $\varphi(\cdot)$ and S is the matrix representation of the linear map, spectral properties of the matrix S are very important in the behavior of the error correlation matrix $Q[k]$. In this regard, we first note that S has complex eigenvalues in general. Secondly, the matrix S always has a real nonnegative eigenvalue that is equal to its spectral radius, and the corresponding eigenvector is the vectorized version of a positive semi-define matrix. More precisely, it is always possible to find a nonzero $X \succeq \mathbf{0}$ such that the following holds true:

$$\varphi(X) = \rho(S)\,X, \qquad (39)$$

which follows from the extensions of the Perron-Frobenius theorem to positive maps in more general settings. We refer to [58, Theorem 5], or [59, Theorem 2.5] for the precise details.

B. THE NECESSARY AND SUFFICIENT CONDITION

As a corollary to Theorem 1, we present the following result regarding the long term behavior of the error correlation matrix:

Corollary 1: If the following holds true:

$$\rho(S) < 1, \qquad (40)$$

where the matrix S is as in (37)*, then the following holds true regarding the error correlation matrix of the state variables:*

$$\lim_{k \to \infty} \Big(Q[k] - Q^{\mathrm{n}} - Q^{\mathrm{r}}[k]\Big) = \mathbf{0}, \qquad (41)$$

where the matrices Q^{n}, $Q^{\mathrm{r}}[k] \in \mathbb{C}^{N \times N}$ are given as the solution of the following linear matrix equations:

$$Q^{\mathrm{n}} = \varphi(Q^{\mathrm{n}}) + P\,\mathbf{\Gamma}\,P + \mathbf{\Gamma} \odot \big(P - P^2\big). \qquad (42)$$

$$Q^{\mathrm{r}}[k+1] = \varphi\big(Q^{\mathrm{r}}[k]\big) + \Big(\boldsymbol{\delta}[k]\ \boldsymbol{\delta}^{\mathrm{H}}[k]\Big) \odot \big(P^{-1} - I\big). \qquad (43)$$

Conversely, if the condition (40) *is violated, then $Q[k]$ increases unboundedly as k goes to infinity. The matrix $\mathbf{\Gamma}$ in* (42) *refers to the noise correlation matrix as defined in* (4).

Proof: See Appendix D. ■

Regarding the limit in (41) it is important to note that the error correlation matrix $Q[k]$ does not converge to a point in general. So, $\lim_{k\to\infty} Q[k]$ may not exist. However, as long as the stability condition (40) is met, Corollary 1 shows that $Q[k]$ approaches the sum $Q^{\mathrm{n}} + Q^{\mathrm{r}}[k]$, where Q^{n} is the error due to the input noise and $Q^{\mathrm{r}}[k]$ is the error due to the randomized asynchronicity. In what follows we will discuss these terms.

1) ERROR DUE TO THE NOISE

The input noise affects the error correlation matrix through the term Q^{n} defined by the equation (42). By vectorizing both sides of (42), a numerical solution to Q^{n} can be obtained as follows:

$$\mathrm{vec}(Q^{\mathrm{n}}) = \big(I - S\big)^{-1}\,\mathrm{vec}\Big(P\,\mathbf{\Gamma}\,P + \mathbf{\Gamma} \odot \big(P - P^2\big)\Big). \qquad (44)$$

We point out that Q^{n} satisfies $Q^{\mathrm{n}} \succ \mathbf{0}$ as long as $\mathbf{\Gamma} \succ \mathbf{0}$ (see Corollary 2 in Section III-C), and it does not have any dependency on the iteration index k. So, the effect of the input noise remains the same throughout the iterations (which is *not* the case for $Q^{\mathrm{r}}[k]$). Furthermore, it is clear from (44) that Q^{n} depends linearly on the noise covariance matrix $\mathbf{\Gamma}$. However, the error due to the noise is always larger than the noise itself, which is stated more precisely in the following lemma:

Lemma 1: For any given A and P satisfying the stability condition (40)*, the following holds true:*

$$(Q^{\mathrm{n}})_{i,i} \geq (\mathbf{\Gamma})_{i,i} \quad \forall\, 1 \leq i \leq N, \qquad (45)$$

where Q^{n} is the solution of the linear matrix equation in (42).

Proof: See Appendix E. ■

In words, Lemma 1 states that the error variance due to the noise in a state variable is always larger than the variance of the noise at the input term of the state recursion. The relation between Q^{n} and the matrices A and P are quiet intricate. In fact, one can search for a set of probabilities that minimize $\text{tr}(Q^{\text{n}})$ for a given A and $\mathbf{\Gamma}$. However, the optimal choice for P is not known at this time.

2) RANDOMIZATION ERROR

Due to the randomized nature of the updates in (14) there is an inherent error in the state vector that is given precisely by the term $Q^{\text{r}}[k]$ in (43). An important observation is that $Q^{\text{r}}[k]$ *does depend* on the iteration index k in general (unlike the error due to the noise). More precisely, the solution to (43) can be written explicitly as follows:

$$Q^{\text{r}}[k] = \sum_{i=1}^{R} \sum_{j=1}^{R} Q_{i,j} \left(\alpha_i \, \alpha_j^*\right)^k, \tag{46}$$

where α_i's denote the base of the exponential input signals as in (7), and the matrices $Q_{i,j}$'s are as follows:

$$\begin{aligned}\text{vec}(Q_{i,j}) = {} & (1-\alpha_i)\,(1-\alpha_j^*) \\ & \cdot\left(\alpha_i\,\alpha_j^* I - S\right)^{-1} \text{vec}\Big((x_i^{\text{ss}}[0]\;(x_j^{\text{ss}}[0])^{\text{H}}) \odot (P^{\text{-}1} - I)\Big),\end{aligned} \tag{47}$$

and $x_i^{\text{ss}}[0]$ is as in (21).

The solution in (46) shows that decaying components of the input signal, i.e.,$|\alpha_i| < 1$, affect the error correlation matrix initially only. Their effect fade away as the iterations progress. On the other hand, the components with $|\alpha_i| = 1$ (i.e., complex sinusoids) have a sustaining effect on the error correlation.

C. MARKOV JUMP LINEAR SYSTEM VIEWPOINT

We would like to point out that the random asynchronous model in (14) can be viewed as a particular instance of a Markov jump linear system, which has the following model:

$$x[k+1] = A_{i_k}\, x[k] + B_{i_k}\, u[k], \tag{48}$$

where i_k denotes the state of the underlying Markov chain at the iteration k, and the Markov chain has finite number of states. So, the state vector $x[k]$ is updated with a different state transition matrix in every iteration (as determined by the state of the Markov chain). This is a well-studied model, and we refer to [32] for a rigorous treatment of the topic.

It is possible to represent the random asynchronous model (14) as a Markov jump system with the underlying Markov chain having 2^N states since there are 2^N different ways of selecting an update set in every iteration of the model. More precisely, the j^{th} transition matrix can be written as follows:

$$A_j = I + P_{\mathcal{T}_j}(A - I), \qquad B_j = P_{\mathcal{T}_j} B, \tag{49}$$

where $\mathcal{T}_j$ is the j^{th} subset of $\{1, \cdots, N\}$ (assuming all 2^N possible update sets are enumerated). As the model (14) assumes that each state variable is updated independently in every iteration, we get $\mathbb{P}[i_k = j] = \gamma_j$ for all k in the switching network viewpoint of (48). Namely, the state variables are updated with the pair $(A_j,\, B_j)$ with probability γ_j in every iteration, and the probability is given as follows:

$$\gamma_j = \prod_{i \in \mathcal{T}_j} p_i \prod_{i \notin \mathcal{T}_j} (1 - p_i). \tag{50}$$

In fact, the function $\varphi(\cdot)$ defined in (33) is closely related to the set of transition matrices and their corresponding selection probabilities. More precisely (see [34, Section 4.8.6]):

$$\varphi(X) = \sum_{j=1}^{2^N} \gamma_j\, A_j\, X\, A_j^{\text{H}}, \qquad S = \sum_{j=1}^{2^N} \gamma_j\, A_j^* \otimes A_j. \tag{51}$$

Furthermore, a direct application of [32, Corollary 3.26] to the random asynchronous model (14) gives the following result:

Lemma 2 (See [34, Lemma 4.2]): The following statements are equivalent:

- *Random asynchronous model in* (14) *is stable in the mean-squared sense,*
- $\rho(S) < 1$,
- *There exists* $X \succ \mathbf{0}$ *such that* $X \succ \varphi(X)$,
- *For any given* $Y \succ \mathbf{0}$, *there exists a unique* $X \succ \mathbf{0}$ *such that* $X = \varphi(X) + Y$,

where $\varphi(\cdot)$ *is as in* (33), *and the matrix* S *is as in* (37).

Although Lemma 2 endorses the importance of the matrix S in the stability of the random asynchronous model, it is important to note that the viewpoint of switching systems provides a more general framework for randomized linear techniques [34, Chapter V]. In the case of random asynchronous updates, underlying state transition matrices (i.e., states of the Markov chain) are related to the matrix A in a very specific way. So, the analysis presented in this study is tailored for the model in (14), and its mean-squared stability is shown to be determined precisely by the spectral radius of the matrix S.

D. A SUFFICIENT CONDITION

In addition to the necessary and sufficient condition given by Corollary 1, it is also possible to ensure the stability of the recursions with a stronger condition based on a simple linear matrix inequality. In this regard, we present the following result as a corollary to Theorem 1:

Corollary 2: If the state-transition matrix A *and the update probabilities* P *satisfy the following*:

$$A^{\text{H}}\, P\, A \prec P, \tag{52}$$

then, the trace of the error correlation matrix of the state variables can be bounded as follows:

$$\limsup_{k \to \infty}\; \text{tr}\left(Q[k]\right) \;\leq\; \frac{\text{tr}(P\,\mathbf{\Gamma}) + \mathbf{\Delta}^2\, \|P^{-1} - I\|_2}{\lambda_{\min}(P - A^{\text{H}}\, P\, A)}, \tag{53}$$

where $\mathbf{\Delta}$ *is an arbitrary number satisfying the following*:

$$\left\|\delta[k]\right\|_2 \leq \mathbf{\Delta} \quad \forall\, k. \tag{54}$$

Proof: See [34, Section 4.8.7]. ■

A number of remarks regarding Corollary 2 are in order:

1) *Convergence of the error correlation:* When the input signal consists of multiple exponential signal, i.e., $R > 1$, the error correlation matrix shows an oscillatory behavior as described in (46). As a result, $\mathrm{tr}(Q[k])$ does not converge in general. Nevertheless, (53) provides an upper bound on the error term as the iterations progress (k goes to infinity).

2) *Difference in the state variables:* As long as the state variables have a finite steady-state component, it is always possible to select a finite value for $\boldsymbol{\Delta}$. Generally speaking, when the input signal $u[k]$ varies slowly, the vector $\boldsymbol{\delta}[k]$ tends to be smaller. So, the upper bound (as well as the error term itself) gets smaller.

3) *Equal probabilities:* When all the indices are updated with equal probabilities, i.e., $P = p\,I$ for some p, the condition (52) reduces to $\|A\|_2 < 1$. So, the error correlation matrix stays bounded when the state variables are updated with equal probabilities (no matter what the probability is) and A has a bounded spectral norm. However, the probability does affect the actual value of the error correlation matrix.

E. THE CONSTANT INPUT: FIXED-POINT ITERATIONS

When the input signal is time-invariant, i.e., $u[k] = u$ for all k, it suffices to consider the case of $R = 1$ with $\alpha = 1$ in the model (7). In this case, we get

$$x^{\mathrm{ss}}[k] = \left(I - \overline{A}\right)^{-1} \overline{B} = \left(I - A\right)^{-1} B = x^{\star}, \tag{55}$$

which shows that the steady-state component $x^{\mathrm{ss}}[k]$ depends neither on the update probabilities nor on the iteration index k. In fact, $x^{\mathrm{ss}}[k] = x^{\star}$ corresponds to *the fixed-point* of the asynchronous iterations in (7), i.e., $x^{\star} = A\,x^{\star} + B$.

Asynchronous (non-random) fixed-point iterations are well studied problems in the literature. Theoretical analysis of the linear case can be traced back to the studies in [60], [61], which showed that the following condition is both necessary and sufficient for the convergence of the asynchronous updates:

$$\rho(|A|) < 1, \tag{56}$$

where $|A|$ is the matrix obtained by replacing the elements of A by their absolute values.

It was shown in [23, Lemma 1] that the sufficiency condition in (52) is more relaxed than the condition in (56). Although these results appear to be contradictory, the difference is the notion of convergence: the condition (56) is necessary and sufficient for the convergence of any index sequence (as in sure convergence), whereas the condition (40) is necessary and sufficient for the mean-squared convergence. In addition, the result in [32, Corollary 3.46] implies that the condition (40) is *sufficient* for almost sure convergence as well.

As a result, we conclude that the asynchronous case is more restrictive than the synchronous case when the worst case behavior is considered. On the other hand, the asynchronous case may be less restrictive than the synchronous case when the statistical behavior is considered.

IV. SYNCHRONOUS vs. ASYNCHRONOUS STABILITY: CONNECTIONS AND COMPARISONS

In previous sections we studied stability of the randomized state recursions from two different perspectives, namely expected behavior and the second order statistics of the error term. In this section, we will discuss the relations between stability of the state-transition matrix A and the matrices $\overline{A}$ and S. In particular, we will show that stability in the synchronous world and stability in the random asynchronous world do not imply each other in general. This is contrary to non-random asynchronicity, which is indeed more restrictive than the synchronous case [60]. Furthermore, stability in the random asynchronous world depends on the update probabilities P as well as the eigenvector structure of the matrix A.

Recall from Section III that when a randomized asynchronous system is said to be stable, it means that the error correlation matrix $Q[k]$ stays bounded as the randomized iterations progress. In general, the error correlation matrix does not converge to zero even when no noise is present in the system. In fact, the amount of error depends on the amount of variation in the input signal as well as the update probabilities.

Two remarks are in order:

1) THE SYNCHRONOUS CASE

Results given by Proposition 1 and Corollary 1 are consistent with the synchronous case. That is,

$$P = I \quad \Longrightarrow \quad \overline{A} = A, \quad S = A^{*} \otimes A, \tag{57}$$

which shows that stability of $A, \overline{A}$ and S are equivalent to each other in the case of synchronous updates. In this case we also note that the randomization error (43) becomes $Q^{\mathrm{r}}[k] = \mathbf{0}$. So, as long as A is a stable matrix, the error correlation matrix converges to Q^{n} whose value depends only on the noise statistics $\boldsymbol{\Gamma}$ and the matrix A.

2) STRICT POSITIVITY OF THE UPDATE PROBABILITIES

Stability of the matrix S implicitly requires the strict positivity of the update probabilities. More precisely,

$$\rho(S) < 1 \quad \Longrightarrow \quad P \succ \mathbf{0}, \tag{58}$$

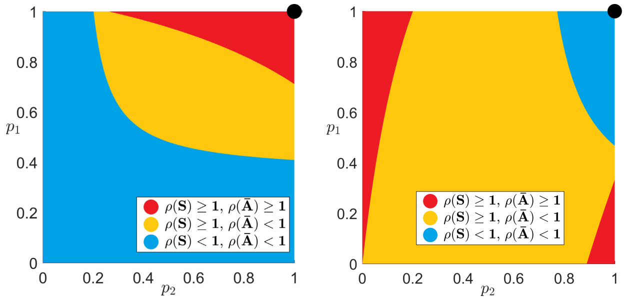
which can be verified by observing that when there exists an index i such that $p_i = 0$ the matrix S has a left eigenvector $e_i \otimes e_i$ with eigenvalue 1. So, the stability condition requires no state variable to be left out *permanently* during the updates.

A. MEAN vs. MEAN-SQUARE ERROR

Since having a finite variance is more restrictive than having a finite mean for a random variable, it is reasonable to expect that stability of the matrix S is more restrictive than stability of the matrix $\overline{A}$. This is, indeed, the case:

Lemma 3: Stability of the matrix S implies stability of the matrix $\overline{A}$, that is,

$$\rho(S) < 1 \quad \Longrightarrow \quad \rho(\overline{A}) < 1. \tag{59}$$



(a)

(b)

FIGURE 2. The set of probabilities that ensures the stability of the randomized asynchronous updates for the matrices (a) A_1, (b) A_2 defined in (61). The top-right corner indicates $P = I$, which corresponds to the synchronous case.

Proof: From the definition of $\varphi(\cdot)$ in (33), we have $\varphi(X) \succeq \overline{A}\, X\, \overline{A}^{\mathrm{H}}$ for any $X \succeq \mathbf{0}$.

From Lemma 2, the condition $\rho(S) < 1$ implies that there exist $X \succ \mathbf{0}$ such that the following holds true:

$$X \succ \varphi(X) \succeq \overline{A}\, X\, \overline{A}^{\mathrm{H}} \quad \Longrightarrow \quad X \succ \overline{A}\, X\, \overline{A}^{\mathrm{H}}, \tag{60}$$

which implies that $\rho(\overline{A}) < 1$ due to the stability properties of the discrete Lyapunov equation. ■

The importance of Lemma 3 follows from the fact that there is no need to consider matrices $\overline{A}$ and S separately. As long as S is stable, it is guaranteed that $\mathbb{E}[q[k]]$ converges to zero and $\mathbb{E}[\|q[k]\|_2^2]$ remains bounded as the iterations progress. Since the converse of (59) does not hold true in general (see Figure 2), we will focus on stability of S in the rest of this section.

B. STABILITY IN SYNCHRONOUS vs. ASYNCHRONOUS CASE

The most important observation regarding stability of the randomized asynchronous state recursions is that a stable synchronous system may get unstable with randomized asynchronicity, and conversely an unstable system (in the synchronous world) may be stabilized simply by the use of randomized asynchronicity. This is a remarkable property of the randomized asynchronous updates, which is observed also in [19], [21] for the case of A being a normal matrix. We state this observation formally with the following lemma:

Lemma 4: In general, stability of A is neither necessary nor sufficient for stability of $\overline{A}$ and stability of S.

Proof: Consider the following examples of size $N = 2$:

$$A_1 = \begin{bmatrix} -0.9 & 0.8 \\ 0.8 & -0.3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.25 & 0.25 \\ -6.25 & -1.25 \end{bmatrix}, \tag{61}$$

which can be verified to satisfy $\rho(A_1) > 1$ and $\rho(A_2) = 0$. Then, we construct the matrices $\overline{A}$ and S as in (20) and (37), respectively for both A_1 and A_2 for all possible values of $P = \mathrm{diag}([p_1 \;\; p_2])$. Figure 2 presents the regions of P for which $\overline{A}$ is stable, or S is stable. This proves the claim. ■

We would like to note that the sufficiency condition given by Corollary 2 does require A to be a stable matrix. So, Corollary 2 fails to explain why unstable synchronous systems may get stable with the randomized asynchronicity. The importance of Corollary 2 follows from the fact that condition (52) is easy to check, or satisfy, in practical applications [23, Theorem 3].

C. THE SET OF PROBABILITIES ENSURING STABILITY

Although Figure 2(a) shows that some unstable systems (in the synchronous world) can get stable with the use of randomized asynchronicity, it should be noted that *not every unstable synchronous system can get stable with randomized asynchronicity.* Therefore, it is important to check whether or not there exists a set of update probabilities for which randomized asynchronous recursions are stable. In this regard, we consider the following definition:

Definition 1 (The stability set): For a given matrix A, we will use $\mathcal{S}(A)$ to denote the set of probabilities such that the matrix S is stable. More precisely,

$$\mathcal{S}(A) = \big\{P \mid \rho(S) < 1, \;\; \mathbf{0} \preceq P \preceq I\big\}, \tag{62}$$

where P is diagonal, and the matrix S is as in (37).

As a numerical example, consider the matrices A_1 and A_2 in (61). The blue regions in Figures 2(a) and 2(b) denote the stability sets $\mathcal{S}(A_1)$ and $\mathcal{S}(A_2)$, respectively.

Some remarks are in order regarding Definition 1:

1) If the set $\mathcal{S}(A)$ is empty for a given matrix A, then the system cannot be stable whether the updates are synchronous, or asynchronous.

2) When the matrix A itself is stable, i.e., $\rho(A) < 1$, the stability set $\mathcal{S}(A)$ is not empty since $I \in \mathcal{S}(A)$.

3) The stability set $\mathcal{S}(A)$ is not convex in general. That is, a random asynchronous system can be stable with probabilities P_1 or P_2, but it may get unstable with $\tau\, P_1 + (1 - \tau)\, P_2$ where $0 < \tau < 1$. See Figure 2(a) as an example, in which the stability set (indicated with color blue) is visibly non-convex.

4) The sufficiency condition given by Corollary 2 describes *a convex subset* of the stability set. More precisely,

$$A^{\mathrm{H}}\, P\, A \prec P \quad \Longrightarrow \quad P \in \mathcal{S}(A), \tag{63}$$

and it is readily verified that the set of probabilities satisfying the condition (52) is convex. However, it should be noted that when A is not a stable matrix, the set of probabilities described by the condition $A^{\mathrm{H}} P A \prec P$ is empty, whereas the stability set $\mathcal{S}(A)$ may or may not be empty.

V. JOINT GRAPH-TIME FILTERING ON GRAPHS

This section considers joint vertex-time filtering on graphs, in which nodes are assumed to communicate with each other randomly and asynchronously. We will show that the proposed communication scheme executes the time domain filtering of classical signal processing and the spatial filtering of graph signal processing *simultaneously*. We would like to note that joint graph (vertex) - time filtering is a well-studied topic in the field of graph signal processing [26]–[31]. The main advantage of the implementation presented in this study is due to its ability to handle random asynchronous behavior of the nodes. We also note that this section does not consider the design of such filters (for which we refer to [26]–[31] and references therein); rather, we will study the behavior of the proposed random asynchronous communication scheme for a filter specified as follows:

$$h(x) = p(x)\,/\,q(x), \tag{64}$$

where $p(x)$ and $q(x)$ are polynomials of degree (at most) L, and they are assumed to be in the following form:

$$p(x) = \sum_{n=0}^{L} p_n\, x^n, \qquad q(x) = 1 + \sum_{n=1}^{L} q_n\, x^n. \tag{65}$$

The coefficients are allowed to be complex in general, i.e., $p_n, q_n \in \mathbb{C}$. In particular, polynomial filters, which corresponds to the case of $q_1 = \cdots = q_L = 0$, are not excluded. In addition, let the quadruple $(F,\ b,\ c,\ d)$ be an L-dimensional state-space realization such that the rational function (64) has the following infinite order polynomial representation:

$$h(x) = d + \sum_{n=1}^{\infty} c\, F^{n-1}\, b\, x^n, \tag{66}$$

where $(F,\ b,\ c,\ d)$ have the following dimensions:

$$F \in \mathbb{C}^{L\times L}, \quad b \in \mathbb{C}^{L}, \quad c^{\mathrm{T}} \in \mathbb{C}^{L}, \quad d \in \mathbb{C}. \tag{67}$$

In the following we will use $G \in \mathbb{C}^{N\times N}$ to denote a graph operator for the graph with N nodes. Here $G_{i,j}$ denotes the weight of the edge from node j to node i. In particular, $G_{i,j} = 0$ when nodes i and j are not neighbors. Examples of such local graph operators include the adjacency matrix, the graph Laplacian, etc. We will use $\mathcal{N}_{\text{in}}(i)$ and $\mathcal{N}_{\text{out}}(i)$ to denote the incoming and outgoing neighbors of the node i. More precisely we have:

$$\mathcal{N}_{\text{in}}(i) = \{j \mid G_{i,j} \neq 0\}, \quad \mathcal{N}_{\text{out}}(i) = \{j \mid G_{j,i} \neq 0\}. \tag{68}$$

In the graph setting considered here nodes are assumed to operate in discrete-time. In particular, let $u_i[k]$ denote the input signal at the i^{th} node at the k^{th} time step, and let $u[k]$ be *the input graph signal* at time k:

$$u[k] = \begin{bmatrix} u_1[k] & u_2[k] & \cdots & u_N[k] \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^N. \tag{69}$$

Similarly, let $y_i[k]$ denote the output signal at the i^{th} node, and let $y[k]$ be *the output graph signal* at time k:

$$y[k] = \begin{bmatrix} y_1[k] & y_2[k] & \cdots & y_N[k] \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^N. \tag{70}$$

In short, subscripts are graph-node indices, and arguments as in $[k]$ are time indices.

We will consider the setting where the output graph signal is related to the input signal via the node-asynchronous filtering operation described in [23]. Namely, the i^{th} node is assumed to have a state vector $x_i \in \mathbb{C}^L$, and a buffer of size $L \cdot |\mathcal{N}_{\text{in}}(i)|$ so that the node stores the state vectors of its *incoming* neighbors. At time k, the node i either stays inactive with probability $1-p_i$, or it gets activated with probability p_i and executes the following filtering steps sequentially:

$$x_i' \leftarrow \sum_{j\in\mathcal{N}_{\text{in}}(i)} G_{i,j}\, x_j, \tag{71}$$

$$y_i[k] \leftarrow c\, x_i' + d\, u_i[k], \tag{72}$$

$$x_i \leftarrow F\, x_i' + b\, u_i[k], \tag{73}$$

where $(F,\ b,\ c,\ d)$ is a state-space representation of the filter as in (66), and G is the corresponding graph operator. We will also use P to denote the diagonal matrix with the i^{th} entry being the update probability of the i^{th} node p_i.

Once the filtering stage is completed, the node i broadcasts its most recent state vector x_i to its *outgoing neighbors*, who can use its value to update themselves in future iterations, in a similar manner. We refer to [23, Section III] for a detailed explanation of the described asynchronous filtering procedure.

The study [23] showed also that the filtering operation described in (71)-(73) can be equivalently represented as a random asynchronous variant of the following NL dimensional *augmented* state-space model:

$$x[k+1] = A\, x[k] + b \otimes u[k], \tag{74}$$

$$y[k] = (c \otimes G)\, x[k] + d \otimes u[k], \tag{75}$$

where the state-transition matrix of the augmented model, A, is given as follows:

$$A = F \otimes G \in \mathbb{C}^{NL\times NL}. \tag{76}$$

In the following, we will use the theoretical developments presented in Sections II and III in order to study the behavior of the node-asynchronous filtering operations.

A. MEAN-SQUARED STABILITY OF THE RANDOMIZED FILTERING

Due to the randomized behavior of the nodes, it is clear that the output graph signal $y[k]$ is a stochastic quantity even when the input signal $u[k]$ is deterministic. So, it is necessary to make sure that $y[k]$ has a finite variance in order to interpret the filtering operations meaningfully.

For the mean-squared analysis of the filtering, we will use random switching system viewpoint summarized in Section III-C. As the graph has N independently behaving nodes, random asynchronous variant of (74) can be considered to be switching between 2^N different transition matrices, where the j^{th} transition matrix has the following form:

$$A_j = I_{NL} + (I_L \otimes P_{\mathcal{T}_j})\,(F \otimes G - I_{NL}) \tag{77}$$

with the selection probability γ_j given in (50). By substituting A_j from (77) into the expression (51), the function that controls the evolution of the error correlation matrix in the graph filtering operations can be written as follows:

$$\varphi(X) = \overline{A}\, X\, \overline{A}^{\mathrm{H}} + \Big((\overline{A} - I_{NL})\, X\, (\overline{A} - I_{NL})^{\mathrm{H}} \Big) \odot \big(\mathbb{1}_L \otimes (P^{-1} - I_N) \big), \tag{78}$$

where $\mathbb{1}_L$ denotes the all-ones matrix of size $L \times L$, and the average state transition matrix of the augmented model, $\overline{A}$, is given as follows:

$$\overline{A} = I_{NL} + (I_L \otimes P)\,(F \otimes G - I_{NL}). \tag{79}$$

Spectral properties of the linear map $\varphi(\cdot)$ in (78) is crucial in determining the mean-squared stability of the filtering operations. In this regard, we first obtain the following matrix representation of $\varphi(\cdot)$ by vectorizing both sides of (78):

$$S = \overline{A}^* \otimes \overline{A} + \Big(I_L \otimes (P^{-1} - I_N) \otimes I_{NL}\Big)\, J\, \Big((\overline{A}^* - I_{NL}) \otimes (\overline{A} - I_{NL})\Big), \tag{80}$$

where J is a diagonal matrix as follows:

$$J = \sum_{i=1}^{N} I_L \otimes (e_i\, e_i^{\mathrm{H}}) \otimes I_L \otimes (e_i\, e_i^{\mathrm{H}}) \in \mathbb{R}^{(NL)^2 \times (NL)^2}. \tag{81}$$

In light of Lemma 2 and Corollary 1, stability of the matrix S in (80) determines the mean-squared stability of the filtering operations. More precisely, if the following holds true

$$\rho(S) < 1, \tag{82}$$

then the output graph signal $y[k]$ can be decomposed as follows (for sufficiently large k):

$$y[k] = y^{\mathrm{ss}}[k] + r[k], \tag{83}$$

where $y^{\mathrm{ss}}[k]$ is a *deterministic* quantity representing the expected steady-state behavior of the output vector, and $r[k]$ is the randomization error with the following statistics:

$$\mathbb{E}\big[r[k]\big] = \mathbf{0}, \quad \mathbb{E}\big[\|r[k]\|_2^2\big] < \infty \quad \forall k. \tag{84}$$

We note that (83) holds true for sufficiently large values of k, as the transient-part of the output converges to zero throughout the iterations. As a result, $y^{\mathrm{ss}}[k]$ can be considered as the expected output of the filtering operations when the input signal is $u[k]$. In the following subsections, we will describe the input-output relation of these operations.

We also note that the following condition was shown to be *sufficient* for the mean-squared stability of the filtering operations in the case of time-invariant input signals [23, Theorem 3]:

$$\|F\|_2^2\; G^{\mathrm{H}}\, P\, G \prec P. \tag{85}$$

As the mean-squared stability of random asynchronous recursions does not depend on the input signal (see Corollaries 1 and 2), we can argue that the condition (85) is also sufficient for the mean-squared stability of the filtering operations in the case of time-varying input signals.

B. RESPONSE TO A SINGLE EXPONENTIAL INPUT

In Section II we showed that complex exponentials remain to be eigenfunctions of the randomized asynchronous state recursions in an expected sense, despite the randomized system having time-varying characteristics. We now show that a similar result holds true for the graph filtering operations, which is presented in the following lemma:

Lemma 5: When the input graph signal is as follows:

$$u[k] = u\; e^{j\omega k} \tag{86}$$

for an arbitrary vector $u \in \mathbb{C}^N$, the steady-state component of the output graph signal is given as follows:

$$y^{\mathrm{ss}}[k] = h\big(G\,Z(e^{j\omega})\big)\; u\; e^{j\omega k}, \tag{87}$$

where $h(\cdot)$ is the filter described in (64), and the diagonal matrix $Z(e^{j\omega})$ is as follows:

$$Z(e^{j\omega}) = P\,\Big(P + (e^{j\omega} - 1)\,I_N\Big)^{-1}. \tag{88}$$

Proof: See Appendix F. ■

We note that (87) describes the *temporal* response of the randomized filtering operation over the network, which depends on the filter $h(\cdot)$, the graph operator G, temporal frequency of the input ω as well as the update probabilities P.

Notice also that when the input signal is time-invariant, i.e., $\omega = 0$, the steady-state output becomes $y^{\mathrm{ss}}[k] = h(G)\,u$, which is the *ordinary graph filtering* studied extensively in the field of graph signal processing [4], [5]. In this case, the study [23] further showed that the stochastic output signal $y[k]$ indeed converges to $h(G)\,u$ in the mean-squared sense under the assumption that (85) is satisfied. The stability of S in (80) is, in fact, both necessary and sufficient for the convergence.

C. VERTEX-TIME EIGENFUNCTIONS

Lemma 5 shows that complex exponentials are *temporal* eigenfunctions of the filtering operations described in (71)-(73). However, an arbitrary exponential input signal may not remain invariant over the graph. In this section we will discuss vertex-time eigenfunctions, which remain invariant under the filtering operations both in time and over the graph..

Inspired by the expression in (87), we now consider *an eigenpair* of the matrix $G\,Z(e^{j\omega})$ as follows:

$$G\,Z(e^{j\omega})\; v(e^{j\omega}) = \xi(e^{j\omega})\; v(e^{j\omega}), \tag{89}$$

where we explicitly indicate the dependency of both the eigenvector $v(e^{j\omega})$ and the eigenvalue $\xi(e^{j\omega})$ on the temporal frequency $e^{j\omega}$. When the input graph signal is as follows:

$$u[k] = v(e^{j\omega})\; e^{j\omega k}, \tag{90}$$

the use of Lemma 5 with the input signal as in (90) gives the following output graph signal.

$$y^{\mathrm{ss}}[k] = h\big(\,\xi(e^{j\omega})\big)\; v(e^{j\omega})\; e^{j\omega k} = h\big(\,\xi(e^{j\omega})\big)\; u[k]. \tag{91}$$

Thus, signals in the form (90) are eigenfunctions of the randomized filtering operations both *in time* and *over the* *graph* in an expected sense. Furthermore, the filter $h(\cdot)$ still determines the response of the overall filtering operation.

Although vertex-time eigenfunctions of the filtering operations are given precisely in (90), they do not provide a practical tool since the eigenvectors $v(e^{j\omega})$ are coupled with the temporal frequency $e^{j\omega}$ in general. This poses a limitation, as the eigenvectors in (89) needs to be computed for each temporal frequency. Interestingly, Eigen-elements in time and over graph get decoupled in the case of all nodes having the same update probability, as we shall elaborate next.

D. GRAPH-TIME EIGENFUNCTIONS FOR UNIFORM PROBABILITIES

In this section we will assume that all the nodes have the same update probability, that is, $P = p\,I$ for some $0 < p \leq 1$. So, the matrix $G\,Z(e^{j\omega})$ appearing in Lemma 5 reduces to the following form:

$$G\,Z(e^{j\omega}) = \frac{p}{p + e^{j\omega} - 1}\,G. \tag{92}$$

Thus, eigenvectors of the graph operator G serve as spatial eigenvectors *irrespective of the temporal frequency*. More precisely, assume that the input signal has the following form:

$$u[k] = v\, e^{j\omega k}, \tag{93}$$

where v is an eigenvector of the graph operator G with the corresponding eigenvalue λ, that is, $Gv = \lambda v$. Then, we have the following:

$$y^{\mathrm{ss}}[k] = h\Big(\frac{p\,\lambda}{p + e^{j\omega} - 1}\Big)\, v\, e^{j\omega k}, \tag{94}$$

where p denotes the update probability of the nodes, and λ is the corresponding eigenvalue of the eigenvector v.

The main convenience of uniform node update probabilities follows from the fact that only the eigenvectors of the graph operator G are needed in order to construct vertex-time elements of the filtering operations. This is also consistent with the fact that graph Fourier basis is defined via eigenvectors of the graph operator [5]. So, vertex-time element in (93) is the product of graph and time elements. This observation will lead to the notion of discrete-time graph Fourier transform, as we shall elaborate in the next section.

VI. DISCRETE-TIME GRAPH FOURIER TRANSFORM

The previous section showed that signals in the form of $v\, e^{j\omega k}$ (where v is an eigenvector of the graph G) are eigenfunctions of the randomized filtering operations described in (71)-(73) in the case of all nodes having the same update probability. In this section, we will show than an arbitrary time-varying graph signal $x[k]$ can be decomposed in terms of these functions. This decomposition will be referred to as discrete-time graph Fourier transform (DTGFT).

We note that the notion of discrete-time graph Fourier transform was introduced first in [26] under the name ‘‘joint time-vertex Fourier transform (JFT),’’ and then studied in detail later in [27] within a vertex-time signal processing framework. This section is intended to provide a brief overview of this concept, as it is crucial for us to quantify the response of the randomized filtering operations in the vertex-time domain when all nodes have the same update probability (see Section VII).

In order to discuss DTGFT, we start by assuming that the graph operator G is diagonalizable, which is a common assumption in the field of graph signal processing, and it is readily satisfied for graphs with undirected edges. Then, we write its eigenvalue decomposition as follows [4], [5]:

$$G = V\,\boldsymbol{\Lambda}\,V^{-1}. \tag{95}$$

We note that for a vector $x \in \mathbb{C}^N$, its graph Fourier transform is defined as $\widehat{x} = V^{-1}x$ so that the signal x can be represented as follows:

$$x = \sum_{i=1}^{N} \widehat{x}_i\, v_i, \tag{96}$$

where v_i is the i^{th} eigenvector of the graph operator G.

On the other hand, a discrete-time signal $x[k] \in \mathbb{C}^N$ can be represented via complex exponentials as follows:

$$x[k] = \frac{1}{2\pi}\int_0^{2\pi} \widehat{x}(e^{j\omega})\, e^{j\omega k}\, \mathrm{d}\omega, \tag{97}$$

where $\widehat{x}(e^{j\omega})$ is known as the discrete-time Fourier transform of the signal $x[k]$, and we assume that $\widehat{x}(e^{j\omega})$ exists.

Combining the representations in (96) (graph domain) and in (97) (time domain), we consider the following vertex-time representation of the signal $x[k]$:

$$x[k] = \frac{1}{2\pi}\sum_{i=1}^{N}\int_0^{2\pi} \widehat{x}(\lambda_i,\, e^{j\omega})\, v_i\, e^{j\omega k}\, \mathrm{d}\omega, \tag{98}$$

where the coefficient $\widehat{x}(\lambda_i,\, e^{j\omega})$ is obtained as follows:

$$\widehat{x}(\lambda_i,\, e^{j\omega}) = e_i^{\mathrm{H}}\, V^{-1} \sum_{k=-\infty}^{\infty} x[k]\, e^{-j\omega k}. \tag{99}$$

The coefficient computed as in (99) will be referred to as *discrete-time graph Fourier transform* of the signal $x[k]$ corresponding to the i^{th} graph Fourier element (in spatial domain) and frequency ω (in time domain). We note that $\widehat{x}(\lambda_i,\, e^{j\omega})$ can also be considered as the graph Fourier transform of the discrete-time Fourier transform of $x[k]$.

VII. GRAPH-FREQUENCY RESPONSE OF A FILTER

In Section V we introduced a random node-asynchronous filtering operation, in which each node holds a state vector and they broadcast their state vectors to their neighbors asynchronously at random (discrete) time instances. The precise details of the filtering is presented in (71)-(73).

In Sections V-C and V-D, we presented vertex-time eigenfunctions of the filtering operations, and in Section VI we showed that a time-varying graph signal can be written as a linear combinations of these vertex-time eigenfunctions. By combining these two results, the following lemma presents the graph-frequency response of a filter over a graph:

Lemma 6: Let $\widehat{u}(\lambda_i, e^{j\omega})$ and $\widehat{y}(\lambda_i, e^{j\omega})$ be the discrete-time graph Fourier transform of the input and the expected steady-state output graph signals of the filtering operations in (71)-(73). *When all the nodes have the same update probability p, the following relation holds true*:

$$\widehat{y}(\lambda_i, e^{j\omega}) = h\left(\frac{p\,\lambda_i}{p + e^{j\omega} - 1}\right)\widehat{u}(\lambda_i, e^{j\omega}), \qquad (100)$$

where λ_i is the i^{th} eigenvalue of the graph operator G.

Proof: We first write the input signal as follows:

$$u[k] = \frac{1}{2\pi}\sum_{i=1}^{N}\int_0^{2\pi} \widehat{u}(\lambda_i, e^{j\omega})\, v_i\, e^{j\omega k}\, \mathrm{d}\omega. \qquad (101)$$

Using linearity and the result in (94), we find that the expected steady-state output signal has the following form:

$$y^{\mathrm{ss}}[k] = \frac{1}{2\pi}\sum_{i=1}^{N}\int_0^{2\pi} \widehat{u}(\lambda_i, e^{j\omega})\, h\left(\frac{p\,\lambda_i}{p + e^{j\omega} - 1}\right) v_i\, e^{j\omega k}\, \mathrm{d}\omega, \qquad (102)$$

which proves the desired result. ■

Lemma VII shows that the response of random node-asynchronous filtering operations is determined by $h(\cdot)$. Unlike ordinary graph filters where the spatial response (over the graph) is defined via the eigenvalue of the graph as $h(\lambda)$ and classical digital filters where the frequency response is defined via the frequency as $h(e^{-j\omega})$, the graph-frequency response depends on the graph eigenvalue and temporal frequency jointly. Furthermore, the update probability p is also crucial in determining the response of the filtering operations, and the response changes when the nodes have a different update probability.

As a final remark, we note that this study does not consider the design of vertex-time filters, as they require a dedicated attention. This follows from the fact that the expression $p\lambda/(p + e^{j\omega} - 1)$ defines a circle on the complex plane centered at $\lambda(1-p)/(2-p)$ with radius $|\lambda|/(2-p)$. Since the underlying graph has (at most) N eigenvalues, graph-frequency response of a filter is determined by the value of $h(\cdot)$ around such N circles. As a result, design of filters with desired graph-frequency characteristics is not trivial, and we leave the design problem as a future research direction.

VIII. NUMERICAL SIMULATIONS

In this section we will simulate the behavior of proposed filtering operations in the presence of random node-asynchronicity and illustrate its effect on the graph-frequency response. For more practical applications of vertex-time filtering on graphs, the interested reader can refer to the studies in [26]–[31] (and references therein).

In what follows, we will study the graph visualized in Figure 3. This is a random geometric graph on $N = 150$ nodes, in which nodes are distributed over the region $[0\ 1] \times [0\ 1]$ uniformly at random. Two nodes are connected to each other if the distance between them is less than 0.15, and the graph is undirected. The graph operator, the matrix $G \in \mathbb{R}^{N\times N}$, is selected as the Laplacian matrix whose eigenvalues can be sorted as follows:

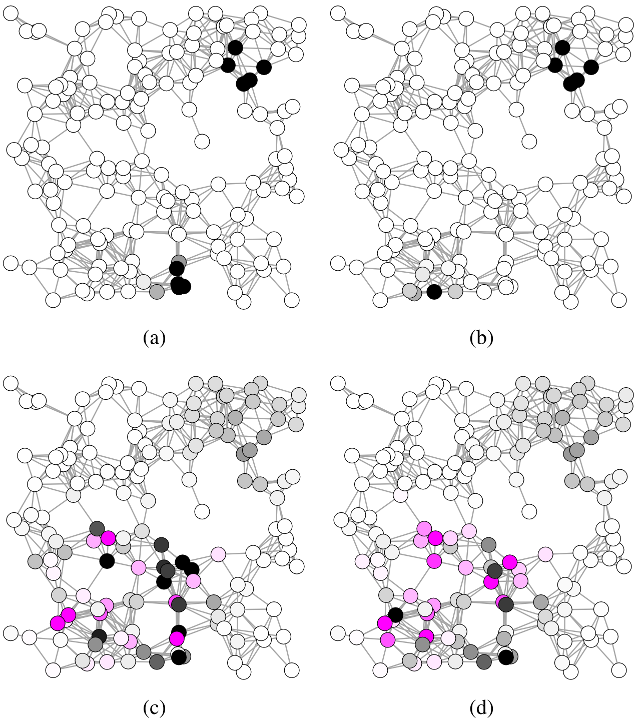


FIGURE 3. Visualization of the signals on the graph. Colors black and pink represent positive and negative values, respectively. Intensity of a color represents the magnitude. The input signal $u[k]$ at (a) $k = 133$, (b) $k = 134$. A realization of the output signal $y[k]$ at (c) $k = 133$, (d) $k = 134$.

$$0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N = \rho(G) = \|G\|_2 = 16.8891, \qquad (103)$$

where the spectral norm of G is computed numerically, and the equality between the spectral radius and the spectral norm follows from the fact that G is a real symmetric matrix.

In order to simulate the filtering operations, we will consider a rational filter $h(x) = p(x)/q(x)$ constructed with the following polynomials of order $L = 3$, and the parameter $\gamma = 0.065$:

$$p(x) = (1 - \gamma\, x)^3, \qquad q(x) = 1 + \sum_{n=1}^{3}\gamma^n x^n. \qquad (104)$$

Regarding the implementation of (104) in the filtering operations described in (71)-(73), we will use the *direct form realization* and select the quadruple (F, b, c, d) as follows:

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\gamma^3 & -\gamma^2 & -\gamma \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad c^{\mathrm{T}} = \begin{bmatrix} -2\gamma^3 \\ 2\gamma^2 \\ -4\gamma \end{bmatrix}, \quad d = 1. \qquad (105)$$

Regarding the filter matrix F in (105) and the graph operator G, we first note that the augmented state-transition matrix

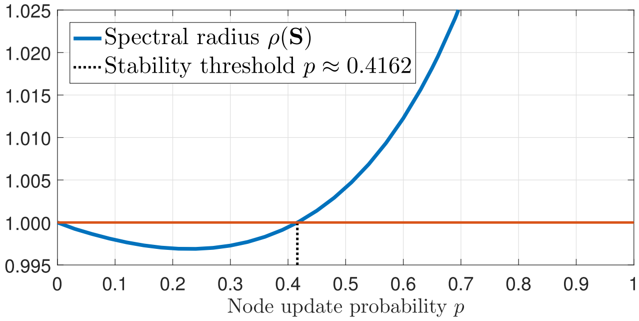


FIGURE 4. Spectral radius of the matrix $S \in \mathbb{C}^{(N^2L^2)\times(N^2L^2)}$ defined in (80) for the filter matrix F in (105) and the graph operator of G.

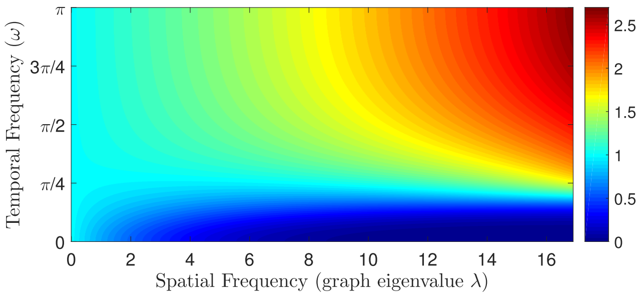
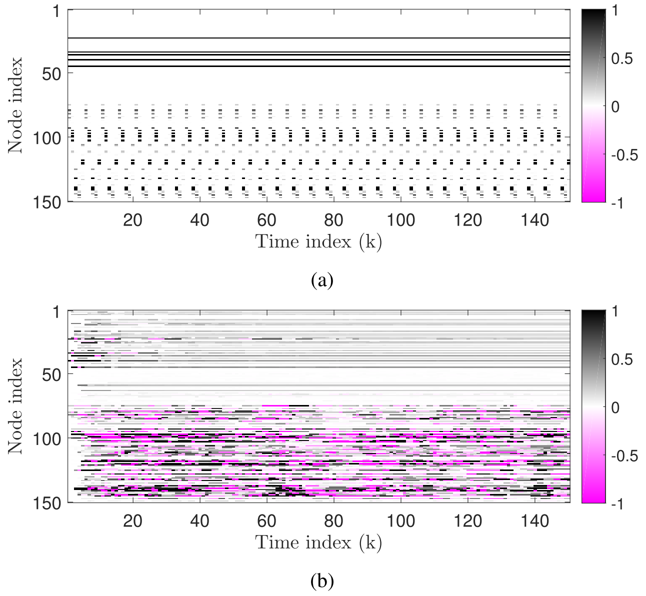


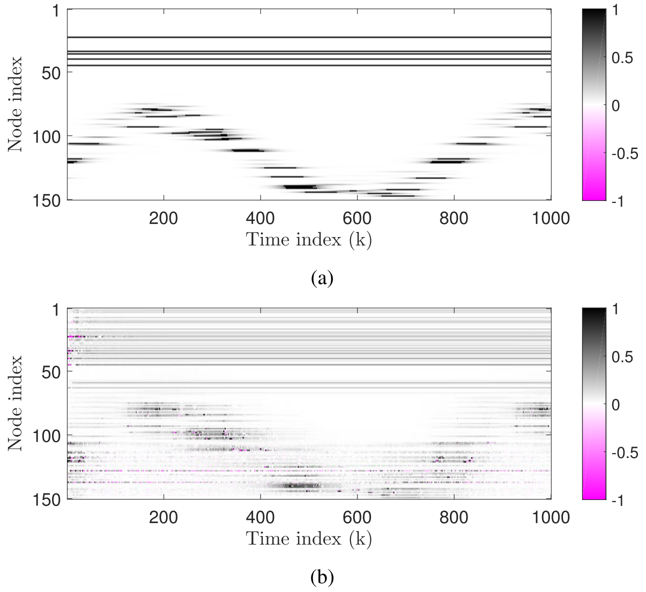
FIGURE 5. Graph-frequency magnitude response of the filter in (104), which is given precisely as $|h(p\,\lambda/(p+e^{j\omega}-1))|$, for the fixed value of $p = 0.4$.



(a)

(b)

FIGURE 6. (a) The input signal $u[k]$ with a rapidly time-varying component. (b) A realization of the random output signal $y[k]$, where the filter is as in (105), and the nodes get updated with probability $p = 0.4$.



(a)

(b)

FIGURE 7. (a) The input signal $u[k]$ with a slowly time-varying component. (b) A realization of the random output signal $y[k]$, where the filter is as in (105), and nodes get updated with probability $p = 0.4$.

in (74) is unstable. More precisely, we have $\rho(F \otimes G) > 1$. Thus, the filtering operations described in (71)-(73) are *not stable in the synchronous case*. Nevertheless, the filtering operations can be made stable (in the mean-squared sense) with the randomized asynchronicity in this particular example. In this regard, we consider the matrix S defined in (80) whose stability determines the mean-squared stability of the randomized updates for a particular set of node update probabilities (see Section V-A). In the rest of this section, we will assume that all the nodes have the same update probability, i.e, $P = pI$. Figure 4 presents the spectral radius of S as a function of the probability p, and the figure shows that the filtering operations remain stable as long as the update probability satisfies $0 < p \leq 0.4162$. So, we use $p = 0.4$ in the rest of the simulations.

The graph-frequency response of the filter in $h(\cdot)$ in (104) is visualized in Figure 5 for the node update probability of $p = 0.4$. Figure 5 shows that $h(\cdot)$ behaves like a low-pass graph filter on the graph at low temporal frequencies, whereas it behaves like a high-pass graph filter at higher temporal frequencies. Thus, filtering characteristics on the graph change with the temporal frequency.

In order to demonstrate the graph-frequency behavior in Figure 5, we will consider the following input graph signal:

$$u[k] = u_1 + u_2[k], \tag{106}$$

where u_1 denotes a time-invariant component that is localized on the graph, and $u_2[k]$ is a rapidly time-varying signal. Figures 3(a) and 3(b) visualize the input signal $u[k]$ on the graph at time indices $k = 133$ and $k = 134$, respectively. We note that u_1 is a 5-sparse signal localized on the top-right portion of the graph, and $u_2[k]$ "moves" around the bottom-left portion of the graph. Figure 6(a) visualizes $u[k]$ for $1 \leq k \leq 150$, in which the nodes of the graph are enumerated according to their vertical positions in Figure 3.

Due to its randomized behavior, the output of the filtering operations is a random quantity, and Figures 3(c) and 3(d) visualize a realization of the output signal $y[k]$ on the graph at time indices $k = 133$ and $k = 134$, respectively. It is clear from the figure that $h(\cdot)$ behaves like a low-pass graph filter for the time-invariant component. Namely, the signal located on the top-right portion of the graph gets smoothed out over the graph. On the contrary, $h(\cdot)$ behaves like a high-pass graph filter for the component $u_2[k]$. The signal located on the bottom-left portion of the graph has a higher amount of variation over the graph. Figure 6(b) visualizes a realization of $y[k]$ for $1 \leq k \leq 150$, which clearly shows the increased variation over the graph for the time-varying component and the decreased variation for the time-invariant component.

We repeat the same experiment with the input signal as in (106), but we now select $u_2[k]$ to be a slowly time-varying signal. This input signal is visualized in Figure 7(a) for $1 \leq k \leq 1000$, and a realization of the corresponding random output signal is presented in Figure 7(b). Due to the temporal low-pass characteristics of the input signal, the filter $h(\cdot)$ behaves a like a low-pass graph filter (see Figure 5 for the graph-frequency response) on the graph. As a result, the output graph signal has a lower variation over the graph.

IX. CONCLUDING REMARKS

In this paper, we proposed novel graph filtering operations that can manipulate the variation of time-varying graph signals both in time and over a graph. In order to analyze the behavior of such randomized operations we first studied a randomized asynchronous variant of the discrete time-invariant state-space models, in which a state variable is updated with some probability independently and asynchronously (with respect to the others) in each iteration. We showed that the randomized model can be treated as a linear time-invariant system (with a frequency response in the expectation sense) despite its randomly time-varying behavior. We presented the necessary and sufficient condition for the mean-squared stability of the randomized state recursions and showed that stability of the underlying state transition matrix is neither necessary nor sufficient for the mean-squared stability. Based on this analysis we provided the necessary and sufficient condition for the mean-square stability of randomized vertex-time filtering operations. We presented eigenfunctions of these graph filtering operations (in the expected sense), based on which we re-visited the notion of discrete-time graph Fourier transform. Based on discrete-time graph Fourier transform of input and output graph signals, we also presented the graph-frequency response of a graph filter in the presence of randomized node-asynchronicity.

For future studies, it is also interesting to study the asynchronous update probabilities that are optimal in terms of minimizing the effect of the randomization error, the input noise, or the rate of convergence. In addition, design of filters with desired graph-frequency characteristics is also left as a future problem.

APPENDIX A
A RESULT ON THE RANDOM INDEX SELECTIONS

Lemma 7: For an arbitrary matrix $X \in \mathbb{C}^{N\times N}$, the following identities hold true:

$$\mathbb{E}\big[P_{\mathcal{T}}\,X\big] = P\,X, \tag{107}$$

$$\mathbb{E}\big[P_{\mathcal{T}}\,X\,P_{\mathcal{T}}\big] = P\,X\,P + X \odot \big(P - P^2\big), \tag{108}$$

where the expectations are taken with respect to the random subset $\mathcal{T}$, to which the i^{th} node belongs with probability p_i.

Proof: The identity in (107) follows directly from the linearity of the expectation and the definition of P in (16).

For the identity (108), we first write the following:

$$\big(P_{\mathcal{T}}\,X\,P_{\mathcal{T}}\big)_{i,j} = \begin{cases} X_{i,j}, & i \in \mathcal{T}, \quad j \in \mathcal{T}, \\ 0, & \text{otherwise}. \end{cases} \tag{109}$$

Thus, we can write the following due to the binary nature of the index selections:

$$\mathbb{E}\Big[\big(P_{\mathcal{T}}\,X\,P_{\mathcal{T}}\big)_{i,j}\Big] = X_{i,j}\ \mathbb{P}[i \in \mathcal{T},\ j \in \mathcal{T}]. \tag{110}$$

Regarding the probabilities in (110), we have the following for the non-diagonal ($i \neq j$) entries:

$$\mathbb{P}[i \in \mathcal{T},\ j \in \mathcal{T}] = \mathbb{P}[i \in \mathcal{T}]\ \mathbb{P}[j \in \mathcal{T}] = p_i\,p_j, \tag{111}$$

which follows from the fact that indices get updated independently from each other. On the other hand, we have the following for the diagonal ($i = j$) entries:

$$\mathbb{P}[i \in \mathcal{T},\ j \in \mathcal{T}] = \mathbb{P}[i \in \mathcal{T}] = p_i, \tag{112}$$

which follows simply from the fact that $i \in \mathcal{T}$ if and only if $j \in \mathcal{T}$ when $i = j$.

Thus, we can write the following:

$$\Big(\mathbb{E}\big[P_{\mathcal{T}}\,X\,P_{\mathcal{T}}\big]\Big)_{i,j} = \begin{cases} p_i\,p_j\,X_{i,j} & i \neq j, \\ p_i\,X_{i,i}, & i = j, \end{cases} \tag{113}$$

which is equivalent to the identity (108). ■

APPENDIX B
PROOF OF PROPOSITION 1

Due to asynchronous updates described in (14), state vector x_k can be written as follows:

$$\begin{aligned} x[k] &= (I - P_{\mathcal{T}_k})\,x[k-1] \\ &\qquad + P_{\mathcal{T}_k}\Big(A\,x[k-1] + \sum_{i=1}^{R} B_i\,\alpha_i^{k-1} + w[k-1]\Big) \\ &= \big(I + P_{\mathcal{T}_k}\,(A - I)\big)x[k-1] \\ &\qquad + P_{\mathcal{T}_k}\left(\sum_{i=1}^{R} B_i\,\alpha_i^{k-1} + w[k-1]\right). \end{aligned} \tag{114}$$

Taking expectation of (114) and using the facts that updated indices are selected independently, the input noise has zero mean, and the noise is uncorrelated with the index selections, we have the following:

$$\begin{aligned} \mathbb{E}[x[k]] &= \overline{A}\,\mathbb{E}[x[k-1]] + \sum_{i=1}^{R} \overline{B}_i\,\alpha_i^{k-1} \\ &= \overline{A}^k\ \mathbb{E}[x[0]] + \sum_{n=0}^{k-1} \overline{A}^n \sum_{i=1}^{R} \overline{B}_i\,\alpha_i^{k-1-n} \end{aligned} \tag{115}$$

$$\begin{aligned}&= \sum_{i=1}^{R} (\alpha_i I - \overline{A})^{-1} \overline{B}_i \, \alpha_i^k \\ &\quad + \overline{A}^k \Big(\mathbb{E}[x[0]] - \sum_{i=1}^{R} (\alpha_i I - \overline{A})^{-1} \overline{B}_i \Big), \\ &= x^{\mathrm{ss}}[k] + \overline{A}^k \Big(\mathbb{E}[x[0]] - x^{\mathrm{ss}}[0] \Big) = x^{\mathrm{ss}}[k] + x^{\mathrm{tr}}[k] \end{aligned} \tag{116}$$

where $\overline{A}$ and $\overline{B}$ are as in (20).

APPENDIX C
PROOF OF THEOREM 1

Using the definition of the error term in (29) and substituting $x[k] = q[k] + x^{\mathrm{ss}}[k]$ into (114), we have the following:

$$\begin{aligned} q[k+1] + x^{\mathrm{ss}}[k+1] &= \big(I + P_{\mathcal{T}_{k+1}} (A - I)\big) \, (q[k] + x^{\mathrm{ss}}[k]) \\ &\quad + P_{\mathcal{T}_{k+1}} \, w[k] + P_{\mathcal{T}_{k+1}} \, P^{-1} \sum_{i=1}^{R} \overline{B}_i \, \alpha_i^k. \end{aligned} \tag{117}$$

which can be written as follows by rearranging the terms:

$$\begin{aligned} q[k+1] = \big(I + P_{\mathcal{T}_{k+1}} (A - I)\big) \, q[k] + P_{\mathcal{T}_{k+1}} \, w[k] \\ + \big(P_{\mathcal{T}_{k+1}} \, P^{-1} - I\big) \, \boldsymbol{\delta}[k]. \end{aligned} \tag{118}$$

where the *deterministic* vector $\boldsymbol{\delta}[k]$ is defined as in (35).

Using (118), we can express the outer product $q[k+1] \, q^{\mathrm{H}}[k+1]$ recursively as follows:

$$\begin{aligned} &q[k+1] \, q^{\mathrm{H}}[k+1] \\ &= P_{\mathcal{T}_{k+1}} \, w[k] \, w^{\mathrm{H}}[k] \, P_{\mathcal{T}_{k+1}} \\ &\quad + \big(I + P_{\mathcal{T}_{k+1}} (A - I)\big) \, q[k] \, q^{\mathrm{H}}[k] \big((A^{\mathrm{H}} - I) P_{\mathcal{T}_{k+1}} + I\big) \\ &\quad + \big(P_{\mathcal{T}_{k+1}} \, P^{-1} - I\big) \, \boldsymbol{\delta}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k] \, \big(P^{-1} \, P_{\mathcal{T}_{k+1}} - I\big) \\ &\quad + \big(I + P_{\mathcal{T}_{k+1}} (A - I)\big) \, q[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k] \big(P^{-1} \, P_{\mathcal{T}_{k+1}} - I\big) \\ &\quad + \big(P_{\mathcal{T}_{k+1}} P^{-1} - I\big) \, \boldsymbol{\delta}[k] \, q^{\mathrm{H}}[k] \, \big(I + (A^{\mathrm{H}} - I) \, P_{\mathcal{T}_{k+1}}\big), \end{aligned} \tag{119}$$

where the cross terms including $w[k]$ are left-out intentionally because these terms will disappear when we take the expectation since $w[k]$ has a zero mean and it is uncorrelated with the index selections.

We now take the expectation of both sides of (119) and use the identities given by Lemma 7, and the independence assumption regarding the index selections, input noise and the initial condition. Then, we obtain the following:

$$\begin{aligned} Q[k+1] = \varphi(Q[k]) & \\ &+ P \, \boldsymbol{\Gamma} \, P + \boldsymbol{\Gamma} \odot \big(P - P^2\big) \\ &+ \big(\boldsymbol{\delta}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k]\big) \odot \big(P^{-1} - I\big) \\ &+ \Big((\overline{A} - I) \, x^{\mathrm{tr}}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k] \Big) \odot \big(P^{-1} - I\big) \\ &+ \Big(\boldsymbol{\delta}[k] (x^{\mathrm{tr}}[k])^{\mathrm{H}} (\overline{A}^{\mathrm{H}} - I) \Big) \odot \big(P^{-1} - I\big), \end{aligned} \tag{120}$$

where the function $\varphi(\cdot)$ is defined in (33). We also note that $\mathbb{E}[q[k]] = x^{\mathrm{tr}}[k]$ as given in (34).

Although $X + X^{\mathrm{H}} \neq 2 \, \Re\{X\}$ in general, we note that the following equality holds true for any $X \in \mathbb{C}^{N \times N}$:

$$\Big(X + X^{\mathrm{H}}\Big) \odot \big(P^{-1} - I\big) = 2 \, \Re\Big\{X\Big\} \odot \big(P^{-1} - I\big), \tag{121}$$

where $\Re\{\cdot\}$ denotes the real part of its argument. So, using the identity (121) in (120) gives the result in (34).

APPENDIX D
PROOF OF COROLLARY 1

We first define the following:

$$Z[k] = Q[k] - Q^{\mathrm{r}}[k] - Q^{\mathrm{n}} \tag{122}$$

where $Q^{\mathrm{r}}[k]$ and Q^{n} are given as the solutions of (43) and (42), respectively. Substituting (122) into the recursion (34), we get:

$$\begin{aligned} &Z[k+1] + Q^{\mathrm{r}}[k+1] + Q^{\mathrm{n}} \\ &= \varphi(Z[k]) + \varphi(Q^{\mathrm{r}}[k]) + \varphi(Q^{\mathrm{n}}) \\ &\quad + P \, \boldsymbol{\Gamma} \, P + \boldsymbol{\Gamma} \odot \big(P - P^2\big) + \Re\Big\{\boldsymbol{\delta}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k]\Big\} \odot \big(P^{-1} - I\big) \\ &\quad + \Re\Big\{2 \, (\overline{A} - I) \, x^{\mathrm{tr}}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k]\Big\} \odot \big(P^{-1} - I\big), \end{aligned} \tag{123}$$

which can be simplified as follows due to the defining equations in (43) and (42):

$$Z[k+1] = \varphi(Z[k]) + \Re\Big\{2 \, (\overline{A} - I) \, x^{\mathrm{tr}}[k] \, \boldsymbol{\delta}^{\mathrm{H}}[k]\Big\} \odot \big(P^{-1} - I\big). \tag{124}$$

Due to the stability assumption (40) and Lemma 3 we have $\rho(\overline{A}) < 1$, so $\lim_{k \to \infty} x^{\mathrm{tr}}[k] = \mathbf{0}$. As a result,

$$\lim_{k \to \infty} Z[k] = \mathbf{0}, \tag{125}$$

which gives the desired result.

Necessity of the condition (40) follows from (39). That is, when $\rho(S) \geq 1$ there exists a nonzero positive semi-definite matrix X that cannot be reduced by the function $\varphi(\cdot)$.

APPENDIX E
PROOF OF LEMMA 1

Assume that the stability condition (40) is met, and solution to (42) exists. Let e_i denote the i^{th} standard basis vector. It is readily verified that the following identity holds true for any $X \in \mathbb{C}^{N \times N}$ and any index $1 \leq i \leq N$:

$$e_i^{\mathrm{H}} \, \varphi(X) \, e_i = (1 - p_i) \, e_i^{\mathrm{H}} \, X \, e_i + p_i \, e_i^{\mathrm{H}} \, A \, X \, A^{\mathrm{H}} \, e_i. \tag{126}$$

Furthermore, we have the following:

$$e_i^{\mathrm{H}} \Big(P \, \boldsymbol{\Gamma} \, P + \boldsymbol{\Gamma} \odot \big(P - P^2\big) \Big) \, e_i = p_i \, e_i^{\mathrm{H}} \, \boldsymbol{\Gamma} \, e_i \tag{127}$$

So, by left and right multiplying (42) with e_i^{H} and e_i respectively, we get the following:

$$e_i^{\mathrm{H}} \, \big(Q^{\mathrm{n}} - \boldsymbol{\Gamma}\big) \, e_i \, p_i = p_i \, e_i^{\mathrm{H}} \, A \, Q^{\mathrm{n}} \, A^{\mathrm{H}} \, e_i \geq 0, \tag{128}$$

where the inequality follows from $Q^{\mathrm{n}} \succeq \mathbf{0}$, and the desired result follows from the fact that $p_i > 0$ for all i.

APPENDIX F
PROOF OF LEMMA 5

Using Proposition 1, the expected steady-state value of the output graph signal can be written as follows:

$$\begin{aligned} y^{\text{ss}}[k] &= d\, u\, e^{j\omega k} \\ &\quad + (c \otimes G)\left(e^{j\omega k}\, I_{NL} - \overline{A}\right)^{-1} (I \otimes P)(b \otimes u)\, e^{j\omega k}, \end{aligned} \tag{129}$$

where $\overline{A}$ is the average augmented state-transition matrix as in (79). Then, we present the following identity, which can be verified via elementary manipulations:

$$\begin{aligned} &\left(e^{j\omega k}\, I_{NL} - \overline{A}\right)^{-1} (I \otimes P) \\ &\quad = \left(I_L \otimes Z^{-1}(e^{j\omega}) - F \otimes G\right)^{-1} \\ &\quad = \left(I_{NL} - F \otimes Z(e^{j\omega})\, G\right)^{-1} \left(I_L \otimes Z(e^{j\omega})\right) \\ &\quad = \sum_{n=0}^{\infty} F^n \otimes \left(Z(e^{j\omega})\, G\right)^n Z(e^{j\omega}), \end{aligned} \tag{130}$$

where $Z(e^{j\omega})$ is as in (88). So, we can write following:

$$\begin{aligned} &(c \otimes G)\left(e^{j\omega k}\, I_{NL} - \overline{A}\right)^{-1} (I \otimes P)(b \otimes u) \\ &\quad = (c \otimes G)\left(\sum_{n=0}^{\infty} F^n \otimes \left(Z(e^{j\omega})\, G\right)^n Z(e^{j\omega})\right)(b \otimes u) \\ &\quad = \sum_{n=0}^{\infty} (c\, F^n\, b)\left(G\, Z(e^{j\omega})\right)^{n+1} u. \end{aligned} \tag{131}$$

When combined with the first term $d\, u\, e^{j\omega k}$, we obtain the following:

$$y^{\text{ss}}[k] = \left(d + \sum_{n=1}^{\infty} (c\, F^{n-1}\, b)\left(G\, Z(e^{j\omega})\right)^n\right) u\, e^{j\omega k} \tag{132}$$

$$= h\left(G\, Z(e^{j\omega})\right) u\, e^{j\omega k}, \tag{133}$$

where the last identity follows from the representation in (66).

REFERENCES

[1] Y. Zeng and S. Wu, *State-Space Models: Applications in Economics and Finance*. New York, NY, USA: Springer-Verlag, 2013.

[2] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., Nov. 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[5] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[6] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 736–751, Dec. 2018.

[7] S. Safavi and U. A. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 54–57, Jan. 2015.

[8] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 1080–1084.

[9] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1406–1413.

[10] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—Part I: Fundamentals," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 409–422, Jan. 2017.

[11] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—Part II: M-channel filter banks," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 423–437, Jan. 2017.

[12] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1113–1117, Aug. 2015.

[13] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[14] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4406–4421, Aug. 2017.

[15] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1931–1935, Nov. 2015.

[16] (2019). *Giraph*. [Online]. Available: https://giraph.apache.org

[17] (2019). *Spark*. [Online]. Available: https://spark.apache.org

[18] (2019). *Dgraph*. [Online]. Available: https://dgraph.io

[19] O. Teke and P. P. Vaidyanathan, "Random node-asynchronous updates on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 11, pp. 2794–2809, Jun. 2019.

[20] O. Teke and P. P. Vaidyanathan, "Node-asynchronous spectral clustering on directed graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5325–5329.

[21] O. Teke and P. P. Vaidyanathan, "The random component-wise power method," *Proc. SPIE*, vol. 11138, pp. 473–488, Sep. 2019.

[22] O. Teke and P. P. Vaidyanathan, "Node-asynchronous implementation of rational filters on graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7530–7534.

[23] O. Teke and P. P. Vaidyanathan, "IIR filtering on graphs with random node-asynchronous updates," *IEEE Trans. Signal Process.*, vol. 68, pp. 3945–3960, 2020.

[24] O. Teke and P. P. Vaidyanathan, "Random node-asynchronous graph computations," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 64–73, Nov. 2020.

[25] M. Coutino and G. Leus, "Asynchronous distributed edge-variant graph filters," in *Proc. IEEE Data Sci. Workshop (DSW)*, Jun. 2019, pp. 115–119.

[26] A. Loukas and D. Foucard, "Frequency analysis of time-varying graph signals," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 346–350.

[27] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 817–829, Feb. 2018.

[28] E. Isufi, G. Leus, and P. Banelli, "2-dimensional finite impulse response graph-temporal filters," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 405–409.

[29] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 200–204.

[30] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4870–4885, Sep. 2019.

[31] A. W. Bohannon, B. M. Sadler, and R. V. Balan, *A Filtering Framework for Time-Varying Graph Signals*. Cham, Switzerland: Springer, 2019, pp. 341–376.

[32] O. L. V. Costa, R. P. Marques, and M. D. Fragoso, *Discrete-Time Markov Jump Linear Systems*. London, U.K.: Springer-Verlag, 2005.

[33] O. Teke and P. P. Vaidyanathan, "Randomized asynchronous recursions with a sinusoidal input," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1491–1495.

[34] O. Teke, "Signals on networks: Random asynchronous and multirate processing, and uncertainty principles," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, Jul. 2020.

[35] S. Safavi and U. A. Khan, "An opportunistic linear–convex algorithm for localization in mobile robot networks," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 875–888, Aug. 2017.

[36] U. A. Khan, S. Kar, and J. M. F. Moura, "Higher dimensional consensus: Learning in large-scale networks," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2836–2849, May 2010.

[37] S. Safavi and U. A. Khan, "Asymptotic stability of LTV systems with applications to distributed dynamic fusion," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5888–5893, Nov. 2017.

[38] T. Sherson, R. Heusdens, and W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 334–347, Oct. 2018.

[39] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[40] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[41] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, "Binary opinion dynamics with stubborn agents," *ACM Trans. Econ. Comput.*, vol. 1, no. 4, pp. 1–30, Dec. 2013.

[42] D. Silvestre, J. Hespanha, and C. Silvestre, "A PageRank algorithm based on asynchronous gauss-seidel iterations," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 484–489.

[43] K. Kambatla, N. Rapolu, S. Jagannathan, and A. Grama, "Asynchronous algorithms in MapReduce," in *Proc. IEEE Int. Conf. Cluster Comput.*, Sep. 2010, pp. 245–254.

[44] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[45] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[46] M. S. Assran and M. G. Rabbat, "Asynchronous gradient push," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 168–183, Jan. 2021.

[47] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks—Part I: Modeling and stability analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–826, Feb. 2015.

[48] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific, 1997.

[49] F. Gama, E. Isufi, A. Ribeiro, and G. Leus, "Controllability of bandlimited graph processes over random time varying graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6440–6454, Dec. 2019.

[50] E. Isufi, P. Banelli, P. D. Lorenzo, and G. Leus, "Observing and tracking bandlimited graph processes from sampled measurements," *Signal Process.*, vol. 177, Dec. 2020, Art. no. 107749.

[51] D. J. Hartfiel, "On infinite products of nonnegative matrices," *SIAM J. Appl. Math.*, vol. 26, no. 2, pp. 297–301, Mar. 1974.

[52] I. Daubechies and J. C. Lagarias, "Sets of matrices all infinite products of which converge," *Linear Algebra Appl.*, vol. 161, pp. 227–263, Jan. 1992.

[53] R. Jungers, *The Joint Spectral Radius: Theory and Applications*. Berlin, Germany: Springer-Verlag, 2009.

[54] L. Elsner, R. Bru, and M. M. Neumann, "Convergence of infinite products of matrices and inner-outer iteration schemes," *Electron. Trans. Numer. Anal.*, vol. 2, pp. 183–193, Dec. 1994.

[55] S.-M. Guu and C.-T. Pang, "On the convergence to zero of infinite products of interval matrices," *SIAM J. Matrix Anal. Appl.*, vol. 25, no. 3, pp. 739–751, Jan. 2003.

[56] H. Avron, A. Druinsky, and A. Gupta, "Revisiting asynchronous linear solvers: Provable convergence rate through randomization," *J. ACM*, vol. 62, no. 6, pp. 51:1–51:27, Dec. 2015.

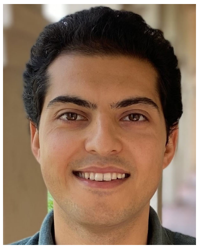
[57] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An algorithmic framework for asynchronous parallel coordinate updates," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A2851–A2879, Jan. 2016.

[58] S. Karlin, "Positive operators," *J. Math. Mech.*, vol. 8, no. 6, pp. 907–937, 1959.

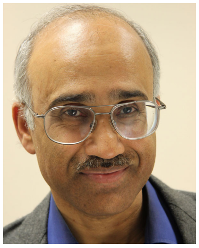
[59] D. E. Evans and R. Høegh-Krohn, "Spectral properties of positive maps on C*-algebras," *J. London Math. Soc.*, vol. 17, no. 2, pp. 345–355, 1978.

[60] D. Chazan and W. Miranker, "Chaotic relaxation," *Linear Algebra Appl.*, vol. 2, no. 2, pp. 199–222, Apr. 1969.

[61] G. M. Baudet, "Asynchronous iterative methods for multiprocessors," *J. ACM*, vol. 25, no. 2, pp. 226–244, Apr. 1978.



OGUZHAN TEKE (Member, IEEE) received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Turkey, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering, with a minor degree in applied and computational mathematics, from California Institute of Technology, USA, in 2020. He was awarded the 2021 Charles and Ellen Wilts Prize for outstanding thesis in electrical engineering at Caltech. His research interests include signal processing, graph signal processing, and randomized algorithms.



PALGHAT P. VAIDYANATHAN (Life Fellow, IEEE) is currently a Kiyo and Eiko Tomiyasu Professor of electrical engineering with California Institute of Technology. His research interests include digital signal processing and machine learning. He is a member of the U.S. National Academy of Engineering. He was a recipient of the IEEE CAS Society Golden Jubilee Medal and the Terman Award of the ASEE. He has received multiple awards for his papers and teaching at Caltech, including the Northrop Grumman Teaching Prize. He was also a recipient of the IEEE Gustav Robert Kirchhoff Award, in 2016, and the IEEE Signal Processing Society's Technical Achievement Award, in 2002, the Education Award, in 2012, and the Society Award, in 2016. He has been selected to receive the EURASIP Athanasios Papoulis Award, in 2021.