Kernel-Based Reconstruction of Space-Time Functions on Dynamic Graphs

Daniel Romero, *Member, IEEE*, Vassilis N. Ioannidis, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Graph-based methods pervade the inference toolkits of numerous disciplines including sociology, biology, neuroscience, physics, chemistry, and engineering. A challenging problem encountered in this context pertains to determining the attributes of a set of vertices given those of another subset at possibly different time instants. Leveraging spatiotemporal dynamics can drastically reduce the number of observed vertices, and hence the sampling cost. Alleviating the limited flexibility of the existing approaches, the present paper broadens the kernel-based graph function estimation framework to reconstruct time-evolving functions over possibly time-evolving topologies. This approach inherits the versatility and generality of kernel-based methods, for which no knowledge on distributions or second-order statistics is required. Systematic guidelines are provided to construct two families of space-time kernels with complementary strengths: the first facilitates judicious control of regularization on a space-time frequency plane, whereas the second accommodates time-varying topologies. Batch and online estimators are also put forth. The latter comprise a novel kernel Kalman filter, developed to reconstruct space-time functions at affordable computational cost. Numerical tests with real datasets corroborate the merits of the proposed methods relative to competing alternatives.**

***Index Terms*—Graph signal reconstruction, Kalman filtering, kernel-based learning, ridge regression, time series on graphs.**

I. Introduction

A NUMBER of applications involving social, biological, brain, sensor, transportation, or communication networks call for efficient methods to infer the attributes of some vertices given the attributes of other vertices [1]. For example, in a social network with vertices and edges respectively representing persons and friendships, one may be interested in determining an individual's consumption trends or political orientation based on those of their friends. This task emerges when sampling cost constraints, such as the impossibility to poll one country's entire population, limit the number of vertices with known attributes. Existing approaches typically formulate this problem as the reconstruction of a function or signal on a graph [1]–[6] and rely on its smoothness with respect to the graph, in the sense that neighboring vertices have similar function values. This principle suggests, for instance, estimating one person's age by looking at their friends' age.

A more challenging problem involves reconstructing time-evolving functions on graphs, such as the ones describing the time-dependent activity of regions in a brain network, given their values on a subset of vertices and time instants. Efficiently exploiting spatiotemporal dynamics can markedly impact sampling costs by reducing the number of vertices that need to be observed to attain a target estimation performance. This reduction is of paramount interest in applications such as invasive electrocorticography (ECoG), where observing a vertex requires the implantation of an intracranial electrode [7].

An extensive body of literature has dealt with reconstructing time-invariant graph functions. Machine learning works typically rely on smoothness [2], [3], [6], [8] to reconstruct either binary-valued (see e.g. [6]) or real-valued functions [8]–[11], whereas the community of signal processing on graphs (SPoG) focuses on parametric estimators for real-valued functions adhering to the *bandlimited model*, by which those functions are confined to the span of B eigenvectors of the graph Laplacian or adjacency matrices [12]–[16]. Most of these approaches can be subsumed under the encompassing framework of time-invariant kernel-based learning [17].

Schemes tailored for time-evolving functions on graphs include [18] and [19], which predict the function values at time t given observations up to time $t-1$. However, these schemes assume that the function of interest adheres to a specific vector autoregression and all vertices are observed at previous time instances. Moreover, [18] requires Gaussianity along with an *ad hoc* form of stationarity.

Some methods for reconstructing time-invariant functions can also track functions that change sufficiently slowly over time. This is the case of the dictionary learning approach in [20] and the distributed algorithms in [21] and [22]. Unfortunately, the flexibility of these algorithms to capture spatial information is also limited since [20] is confined to Laplacian

Manuscript received October 31, 2016; revised April 5, 2017 and June 12, 2017; accepted June 13, 2017. Date of publication July 13, 2017; date of current version August 14, 2017. This work was supported by the Army Research Office under Grant W911NF-15-1-0492 and the National Science Foundation under Grants 1343248, 1442686, and 1514056. This paper was presented in part at the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 6–9, 2016. The source code for the simulations is available at the authors' websites. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Antonio Ortega. *(Corresponding author: Georgios B. Giannakis.)*

D. Romero was with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA. He is now with the Department of Information and Communication Technology, University of Agder, 4879 Grimstad, Norway (e-mail: daniel.romero@uia.no).

V. N. Ioannidis and G. B. Giannakis are with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: ioann006@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2017.2726976

regularization whereas [21] and [22] require the function to be bandlimited. Other works investigate special instances of the reconstruction problem with domain-specific requirements and assumptions [23], [24]. Finally, it is worth mentioning that no approach deals with time-evolving topologies.

The contribution of this paper is threefold. First, the existing kernel-based learning framework is naturally extended to subsume time-evolving functions over possibly dynamic graphs through the notion of *graph extension*, by which the time dimension receives the same treatment as the spatial dimension. The versatility of kernel-based methods to leverage spatial information [17] is thereby inherited and broadened to account for temporal dynamics as well. Incidentally, this vantage point also accommodates time-varying sampling sets and topologies. Second, two families of *space-time kernels* are introduced by generalizing Laplacian kernels [3]. The first family enables kernel design in a bidimensional frequency domain, whereas the second caters for time-varying topologies. The third contribution comprises two *function estimators* with complementary strengths based on the popular kernel ridge regression (KRR) criterion; see e.g. [17], [25]. The first can handle more sophisticated forms of spatiotemporal regularization by operating in batch mode. The second estimator, termed kernel Kalman filter (KKF), finds exact online KRR estimates by refining previous ones as new observations become available, therefore offering a more efficient implementation.

The major novelty of this paper is a purely deterministic methodology that obviates the need for assumptions on data distributions, stationarity, or knowledge of second-order statistics. The proposed schemes are therefore of special interest in absence of sufficient historical data, yet the latter can be incorporated if available through covariance kernels [17]. Although more complicated dynamics can be accommodated, one may simply rely on the assumption that the target function is smooth over the graph and over time, which is reasonable whenever the graph is properly constructed and the sampling interval is attuned to the temporal dynamics of the function.

The novel online estimator constitutes the first fully deterministic rigorous application of the Kalman filter (KF) to kernel-based learning. Although [26] already proposed a kernel-based KF, this work heavily relies on heuristics and approximations to explicitly operate in feature space. Moreover, this algorithm needs to solve the challenging preimage problem per time step, which increases inaccuracy and computational cost. Another kernel-based KF was developed in [27] within the framework of kernel-based learning, but its formulation is probabilistic and requires historical data to estimate distributions. Like the schemes in [26], [27], the proposed KKF can be derived from the framework of reproducing kernel Hilbert spaces (RKHSs). However, to simplify the exposition, this paper presents a derivation of the proposed KKF that solely relies on linear algebra.

The rest of the paper is structured as follows. Section II formulates the problem and Section III reviews kernel-based learning for time-invariant functions. Section IV generalizes this framework to reconstructing time-evolving functions and develops a batch estimator together with the KKF. Subsequently, space-time kernels are designed in Section V and numerical tests are reported in Section VI to validate the performance of the proposed algorithms. Finally, Section VII summarizes some closing remarks whereas the Appendix provides the proofs of the main results.

Notation: Scalars are denoted by lowercase letters, vectors by bold lowercase, and matrices by bold uppercase. $(\boldsymbol{A})_{m,n}$ is the (m,n)-th entry of matrix $\boldsymbol{A}$. Superscripts $^\top$ and † respectively denote transpose and pseudo-inverse. If $\boldsymbol{A} := [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$, then $\mathrm{vec}\{\boldsymbol{A}\} := [\boldsymbol{a}_1^\top, \ldots, \boldsymbol{a}_N^\top]^\top := \boldsymbol{a}$ and $\mathrm{unvec}\{\boldsymbol{a}\} := \boldsymbol{A}$. With $N \times N$ matrices $\{\boldsymbol{A}_t\}_{t=1}^T$ and $\{\boldsymbol{B}_t\}_{t=2}^T$ satisfying $\boldsymbol{A}_t = \boldsymbol{A}_t^\top \, \forall t$, $\mathrm{btridiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_T; \boldsymbol{B}_2, \ldots, \boldsymbol{B}_T\}$ represents the symmetric block tridiagonal matrix

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_2^\top & \boldsymbol{0} & \ldots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_2 & \boldsymbol{A}_2 & \boldsymbol{B}_3^\top & \ldots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_3 & \boldsymbol{A}_3 & \ldots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{A}_{T-1} & \boldsymbol{B}_T^\top \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{B}_T & \boldsymbol{A}_T \end{bmatrix}.$$

Similarly, $\mathrm{bdiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\} := \mathrm{btridiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N; \boldsymbol{0}, \ldots, \boldsymbol{0}\}$ is a block diagonal matrix. Although it was assumed that the matrices $\{\boldsymbol{A}_n\}_{n=1}^N$ have the same sizes, the definition of $\mathrm{bdiag}\{\cdot\}$ can be readily extended to accommodate matrices of different sizes. Symbols $\odot$, $\otimes$, and $\oplus$ respectively denote element-wise (Hadamard) matrix product, Kronecker product, and Kronecker sum, the latter being defined for $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{B} \in \mathbb{R}^{N \times N}$ as $\boldsymbol{A} \oplus \boldsymbol{B} := \boldsymbol{A} \otimes \boldsymbol{I}_N + \boldsymbol{I}_M \otimes \boldsymbol{B}$. The n-th column of the identity matrix $\boldsymbol{I}_N$ is represented by $\boldsymbol{i}_{N,n}$. If $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is positive definite and $\boldsymbol{x} \in \mathbb{R}^N$, then $||\boldsymbol{x}||_{\boldsymbol{A}}^2 := \boldsymbol{x}^\top \boldsymbol{A}^{-1} \boldsymbol{x}$ and $||\boldsymbol{x}||_2 := ||\boldsymbol{x}||_{\boldsymbol{I}_N}$. The cone of $N \times N$ positive definite matrices is denoted by $\mathbb{S}_+^N$. Finally, $\delta[\cdot]$ stands for the Kronecker delta and $\mathbb{E}$ for expectation.

II. Problem Formulation

A time-varying graph[1] is a tuple $\mathcal{G} := (\mathcal{V}, \{\boldsymbol{A}_\mathcal{V}[t]\}_{t=1}^T)$, where $\mathcal{V} := \{v_1, \ldots, v_N\}$ is the vertex set and $\boldsymbol{A}_\mathcal{V}[t] \in \mathbb{R}^{N \times N}$ is the adjacency matrix at time t, whose (n,n')-th entry $A_{n,n'}^\mathcal{V}[t]$ assigns a weight to the pair of vertices $(v_n, v_{n'})$ at time t. A time-invariant graph is a special case with $\boldsymbol{A}_\mathcal{V}[t] = \boldsymbol{A}_\mathcal{V}[t'] \, \forall t, t'$. The edge set is defined as $\mathcal{E}[t] := \{(v_n, v_{n'}) \in \mathcal{V} \times \mathcal{V} : A_{n,n'}^\mathcal{V}[t] \neq 0\}$, and two vertices v and v' are said to be *adjacent*, *connected*, or *neighbors* at time t if $(v, v') \in \mathcal{E}[t]$. It is also said that $A_{n,n'}^\mathcal{V}[t]$ is the weight of the edge $(v_n, v_{n'})$ at time t. As usual, see e.g. [1, Ch. 2], [4], [8], this paper assumes that $\mathcal{G}$ (i) has non-negative weights ($A_{n,n'}^\mathcal{V}[t] \geq 0 \, \forall n, n', t$); (ii) no self-edges ($A_{n,n}^\mathcal{V}[t] = 0 \, \forall n, t$); and, (iii) it is undirected ($A_{n,n'}^\mathcal{V}[t] = A_{n',n}^\mathcal{V}[t] \, \forall n, n', t$).

A time-evolving function or signal on a graph,[2] is a map $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}$, where $\mathcal{T} := \{1, \ldots, T\}$ is the set of time indices. The value $f(v_n, t)$ of f at vertex v_n and time t, or its shorthand

[1] See [28] and references therein for alternative representations of time-varying graphs.

[2] The entire framework can naturally be extended to accommodate complex-valued functions f.

version $f_n[t]$, can be thought of as the value of an attribute of $v_n \in \mathcal{V}$ at time t. In a social network, $f_n[t]$ may denote e.g. the annual income of person v_n at year t. The values of f at time t will be collected in $\boldsymbol{f}[t] := [f_1[t], \ldots, f_N[t]]^\top$.

At time t, the vertices with indices in the time-dependent set $\mathcal{S}[t] := \{n_1[t], \ldots, n_{S[t]}[t]\}$, $1 \leq n_1[t] < \cdots < n_{S[t]}[t] \leq N$, are observed. The resulting samples can be expressed as $y_s[t] = f_{n_s[t]}[t] + e_s[t], s = 1, \ldots, S[t]$, where $e_s[t]$ models observation error. In social networks, this encompasses scenarios where a subset of persons have been surveyed about the attribute of interest; e.g. their annual income. By letting $\boldsymbol{y}[t] := [y_1[t], \ldots, y_{S[t]}[t]]^\top$, the observations can be conveniently expressed as

$$\boldsymbol{y}[t] = \boldsymbol{S}[t]\boldsymbol{f}[t] + \boldsymbol{e}[t], \quad t = 1, \ldots, T \tag{1}$$

where $\boldsymbol{e}[t] := [e_1[t], \ldots, e_{S[t]}[t]]^\top$, and the $S[t] \times N$ sampling matrix $\boldsymbol{S}[t]$ contains ones at positions $(s, n_s[t]), s = 1, \ldots, S[t]$ and zeros elsewhere.

The broad goal of this paper is to "reconstruct" f from the observations $\{\boldsymbol{y}[t]\}_{t=1}^T$ in (1). Two formulations will be considered. In the batch reconstruction problem, one aims at finding $\{\boldsymbol{f}[t]\}_{t=1}^T$ given $\mathcal{G}$, the sample locations $\{\boldsymbol{S}[t]\}_{t=1}^T$, and all observations $\{\boldsymbol{y}[t]\}_{t=1}^T$. On the other hand, the online problem is formulated as follows: at every time t, one is given $\mathcal{G}$ together with $\boldsymbol{S}[t]$ and $\boldsymbol{y}[t]$, and the goal is to find $\boldsymbol{f}[t]$. The latter can be obtained possibly based on a previous estimate of $\boldsymbol{f}[t-1]$, but the complexity per time slot t must be bounded, even if $T \to \infty$. To solve these problems, no explicit parametric model for the temporal or spatial evolution of f will be adopted. For instance, one will be able to solely rely on the assumption that f evolves smoothly over both space and time, yet more structured dynamics can also be incorporated if known.

III. Background on Kernel-Based Reconstruction

This section reviews the existing framework for kernel-based reconstruction of time-invariant graph functions, which aims at solving the batch problem in Section II when $T = 1$. To reflect this scenario, the notation will be devoid of time indices. As a result, the problem becomes finding $\boldsymbol{f} \in \mathbb{R}^N$ given $\boldsymbol{A}_\mathcal{V} \in \mathbb{R}_+^{N \times N}$, $\boldsymbol{S} \in \{0,1\}^{S \times N}$, and $\boldsymbol{y} = \boldsymbol{S}\boldsymbol{f} + \boldsymbol{e} \in \mathbb{R}^S$.

At first, one may feel tempted to seek a least-squares estimate $\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} ||\boldsymbol{y} - \boldsymbol{S}\boldsymbol{f}||_2^2$, but noting that the N unknowns in $\boldsymbol{f}$ cannot be generally identified from the $S \leq N$ samples in $\boldsymbol{y}$ dismisses such an approach. This underdeterminacy prompts estimates of the form $\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} ||\boldsymbol{y} - \boldsymbol{S}\boldsymbol{f}||_2^2 + \mu\rho(\boldsymbol{f})$, where $\mu > 0$ and the regularizer $\rho(\boldsymbol{f})$ promotes a certain structure in $\boldsymbol{f}$. A customary $\rho(\boldsymbol{f})$ encourages smooth estimates by penalizing functions that exhibit pronounced variations among neighboring vertices, for instance by means of the so-called *Laplacian* regularizer

$$\rho_{\text{LR}}(\boldsymbol{f}) := \frac{1}{2}\sum_{n=1}^N \sum_{n'=1}^N A_{n,n'}^\mathcal{V}(f_n - f_{n'})^2 \tag{2}$$

which heavily penalizes differences between function values at vertices connected by strong links (large $A_{n,n'}^\mathcal{V}$). Expression (2) formalizes the notion of smoothness introduced in Section I, according to which a function is smooth if it takes similar values at neighboring vertices. Since $\rho_{\text{LR}}(\boldsymbol{f})$ is small if $\boldsymbol{f}$ is smooth, and large otherwise, $\rho_{\text{LR}}(\boldsymbol{f})$ acts as a proxy quantifying smoothness of $\boldsymbol{f}$, in the sense that given two functions $\boldsymbol{f}$ and $\boldsymbol{f}'$, the former is said to be smoother than the latter iff $\rho_{\text{LR}}(\boldsymbol{f}) < \rho_{\text{LR}}(\boldsymbol{f}')$ and vice versa. More general proxies are reviewed next.

Upon defining the $N \times N$ *Laplacian* matrix $\boldsymbol{L}_\mathcal{V} := \text{diag}\{\boldsymbol{A}_\mathcal{V}\boldsymbol{1}\} - \boldsymbol{A}_\mathcal{V}$, the functional in (2) can be rewritten after some algebra as $\rho_{\text{LR}}(\boldsymbol{f}) = \boldsymbol{f}^\top \boldsymbol{L}_\mathcal{V}\boldsymbol{f}$; see e.g. [1, Ch. 2]. It readily follows from (2) that $\rho_{\text{LR}}(\boldsymbol{f}) \geq 0 \, \forall \boldsymbol{f}$, which in turn implies that $\boldsymbol{L}_\mathcal{V}$ is positive semidefinite. Therefore, $\boldsymbol{L}_\mathcal{V}$ admits an eigenvalue decomposition $\boldsymbol{L}_\mathcal{V} = \boldsymbol{U}_\mathcal{V}\,\text{diag}\{\boldsymbol{\lambda}_\mathcal{V}\}\boldsymbol{U}_\mathcal{V}^\top$, where the eigenvectors in $\boldsymbol{U}_\mathcal{V} := [\boldsymbol{u}_1^\mathcal{V}, \ldots, \boldsymbol{u}_N^\mathcal{V}]$ and the eigenvalues in $\boldsymbol{\lambda}_\mathcal{V} := [\lambda_1^\mathcal{V}, \ldots, \lambda_N^\mathcal{V}]$ are sorted so that $0 = \lambda_1^\mathcal{V} \leq \ldots \leq \lambda_N^\mathcal{V}$.

By letting $\check{f}_n := (\boldsymbol{u}_n^\mathcal{V})^\top \boldsymbol{f}$, one finds that

$$\rho_{\text{LR}}(\boldsymbol{f}) = \sum_{n=1}^N \lambda_n^\mathcal{V} |\check{f}_n|^2 \tag{3}$$

which means that $\rho_{\text{LR}}(\boldsymbol{f})$ is the weighted superposition of the squared magnitude of the projections of $\boldsymbol{f}$ onto the eigenvectors of $\boldsymbol{L}_\mathcal{V}$ with weights given by the corresponding eigenvalues.

As described next, (3) provides an insightful interpretation of $\rho_{\text{LR}}(\boldsymbol{f})$ in a transformed domain. Specifically, a number of works advocate the term *graph Fourier transform* or frequency representation of f to refer to $\{\check{f}_n\}_{n=1}^N$; see e.g., [4]. The main argument resides in that $\{\boldsymbol{u}_n^\mathcal{V}\}_{n=1}^N$ play a role analogous to complex exponentials in signal processing for time signals, in the sense that (i) complex exponentials are eigensignals of the continuous counterpart of the Laplacian operator $\boldsymbol{f} \mapsto \boldsymbol{L}_\mathcal{V}\boldsymbol{f}$, and (ii) $\{\boldsymbol{u}_n^\mathcal{V}\}_{n=1}^N$ are eigensignals of the so-called linear, shift-invariant filters [5], which are the graph counterparts of linear, time-invariant filters in signal processing for time signals. Thus, $\boldsymbol{f} = \sum_{n=1}^N \check{f}_n \boldsymbol{u}_n^\mathcal{V}$ resembles in some sense the synthesis equation of the Fourier transform, and one can therefore interpret $\{\boldsymbol{u}_n^\mathcal{V}\}_{n=1}^N$ as a Fourier basis. Because $\lambda_1^\mathcal{V} \leq \ldots \leq \lambda_N^\mathcal{V}$, it follows from $\rho_{\text{LR}}(\boldsymbol{u}_n^\mathcal{V}) = (\boldsymbol{u}_n^\mathcal{V})^\top \boldsymbol{L}_\mathcal{V}\boldsymbol{u}_n^\mathcal{V} = \lambda_n^\mathcal{V}$ that $\rho_{\text{LR}}(\boldsymbol{u}_1^\mathcal{V}) \leq \ldots \leq \rho_{\text{LR}}(\boldsymbol{u}_N^\mathcal{V})$. Hence, sorting the eigenvectors $\{\boldsymbol{u}_n^\mathcal{V}\}_{n=1}^N$ in increasing order of their associated eigenvalue is tantamount to sorting them in decreasing order of smoothness. Similarly, the complex exponentials in the traditional Fourier basis are indexed by their frequency, which can be thought of as an (inverse) proxy of time-domain smoothness. Comparing both scenarios suggests interpreting $\lambda_n^\mathcal{V}$, or the index n, as the *graph frequency* of $\boldsymbol{u}_n^\mathcal{V}$.

Back to (3), it is seen that $\rho_{\text{LR}}(\boldsymbol{f})$ penalizes high-frequency components more heavily than low-frequency ones, thus promoting estimates with a "low-pass" graph Fourier transform. A finer control of how energy is distributed across frequency can be attained upon applying a transformation $r : \mathbb{R} \to \mathbb{R}_+$ to $\lambda_n^\mathcal{V}$, giving rise to regularizers of the form

$$\rho_{\text{LKR}}(\boldsymbol{f}) = \sum_{n=1}^N r(\lambda_n^\mathcal{V})|\check{f}_n|^2 = \boldsymbol{f}^\top \boldsymbol{K}^\dagger \boldsymbol{f} \tag{4a}$$

where

$$\boldsymbol{K}^\dagger := r(\boldsymbol{L}_\mathcal{V}) := \boldsymbol{U}_\mathcal{V}^\top \,\text{diag}\{r(\boldsymbol{\lambda}_\mathcal{V})\}\boldsymbol{U}_\mathcal{V} \tag{4b}$$

TABLE I
EXAMPLES OF LAPLACIAN KERNELS AND THEIR ASSOCIATED SPECTRAL WEIGHT FUNCTIONS

Kernel name	Function	Parameters
Diffusion kernel [2]	$r(\lambda) = \exp\{\sigma^2\lambda/2\}$	$\sigma^2 \geq 0$
p-step random walk kernel [3]	$r(\lambda) = (a-\lambda)^{-p}$	$a \geq 2$, p positive integer
Laplacian regularization [3], [4], [29]	$r(\lambda) = 1 + \sigma^2\lambda$	σ^2 sufficiently large
Bandlimited [17]	$r(\lambda) = \begin{cases} 1/\beta & \lambda \leq \lambda_{\max} \\ \beta & \text{otherwise} \end{cases}$	$\beta > 0$ sufficiently large, $\lambda_{\max}$

is referred to as *Laplacian kernel* [3]. Table I summarizes some well-known examples arising for specific choices of r.

Further broadening the scope of the generalized Laplacian kernel regularizers in (4), the so-called *kernel ridge regression* (KRR) estimators are given by

$$\hat{\boldsymbol{f}} := \arg\min_{\boldsymbol{f}} \frac{1}{S}||\boldsymbol{y} - \boldsymbol{S}\boldsymbol{f}||_2^2 + \mu \boldsymbol{f}^\top \boldsymbol{K}^\dagger \boldsymbol{f} \tag{5}$$

for an arbitrary positive semidefinite matrix $\boldsymbol{K}$, not necessarily a Laplacian kernel. The user-selected parameter $\mu > 0$ balances the importance of the regularizer relative to the fitting term $S^{-1}||\boldsymbol{y} - \boldsymbol{S}\boldsymbol{f}||_2^2$. KRR estimators have well-documented merits and solid grounds on statistical learning theory; see e.g. [25]. Different regularizers and fitting functions lead to even more general algorithms; see e.g. [17].

IV. KERNEL-BASED RECONSTRUCTION OF TIME-VARYING GRAPH FUNCTIONS

The framework in Section III cannot accommodate functions evolving over both space and time. To flexibly exploit spatial and temporal dynamics, the present section generalizes this framework through the notion of *graph extension*.

An immediate approach to reconstructing time-evolving functions is to apply (5) separately for each $t = 1, \ldots, T$. This yields the instantaneous estimator (IE)

$$\hat{\boldsymbol{f}}_{\text{IE}}[t] := \arg\min_{\boldsymbol{f}} \frac{1}{S[t]}||\boldsymbol{y}[t] - \boldsymbol{S}[t]\boldsymbol{f}||_2^2 + \mu \boldsymbol{f}^\top \boldsymbol{K}^\dagger[t] \boldsymbol{f}. \tag{6}$$

Unfortunately, this estimator does not account for the possible relation between e.g. $f_n[t]$ and $f_n[t-1]$. If, for instance, f varies slowly over time, an estimate of $f_n[t]$ may as well benefit from leveraging observations $y_s[\tau]$ at time instants $\tau \neq t$. Exploiting temporal dynamics potentially reduces the number of vertices that have to be sampled to attain a target reconstruction performance, which in turn can markedly reduce sampling costs.

Incorporating temporal dynamics into kernel-based reconstruction, which can only handle a single snapshot (cf. Section III), necessitates an appropriate reformulation of time-evolving function reconstruction as a problem of reconstructing a time-invariant function. An appealing possibility is to replace $\mathcal{G}$ with its *extended version* $\bar{\mathcal{G}} := (\bar{\mathcal{V}}, \bar{\boldsymbol{A}})$, where each vertex in $\mathcal{V}$ is replicated T times to yield the extended vertex set $\bar{\mathcal{V}} := \{v_n[t],\ n = 1, \ldots, N,\ t = 1, \ldots, T\}$, and the $(n + N(t-1), n' + N(t'-1))$-th entry of the $TN \times TN$ extended adjacency matrix $\bar{\boldsymbol{A}}$ equals the weight of the edge $(v_n[t], v_{n'}[t'])$. The time-varying function f can thus be replaced with its extended time-invariant counterpart $\bar{f} : \bar{\mathcal{V}} \to \mathbb{R}$ with $\bar{f}(v_n[t]) = f_n[t]$.

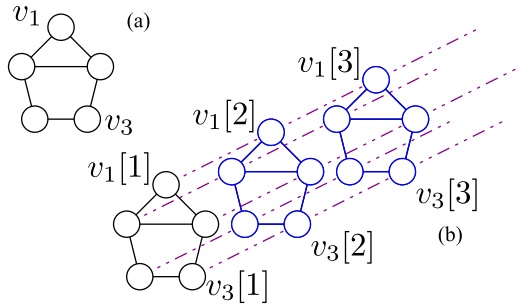

Fig. 1. (a) Original graph $\mathcal{G}$. (b) Extended graph $\bar{\mathcal{G}}$ for diagonal $\boldsymbol{B}_{\mathcal{T}}[t]$. Edges connecting vertices at the same time instant are represented by solid lines, whereas edges connecting vertices at different time instants are represented by dot-dashed lines.

As captured by the following definition, this paper focuses on graph extensions respecting the connectivity of $\mathcal{G}$ per time slot t, that is, $\{v_n[t]\}_{n=1}^N$ are connected according to $\boldsymbol{A}_{\mathcal{V}}[t]$, $\forall t$:

Definition 1: Let $\mathcal{V} := \{v_1, \ldots, v_N\}$ denote a vertex set and let $\mathcal{G} := (\mathcal{V}, \{\boldsymbol{A}_{\mathcal{V}}[t]\}_{t=1}^T)$ be a time-varying graph. A graph $\bar{\mathcal{G}}$ with vertex set $\bar{\mathcal{V}} := \{v_n[t],\ n = 1, \ldots, N,\ t = 1, \ldots, T\}$ and $NT \times NT$ adjacency matrix $\bar{\boldsymbol{A}}$ is an extended graph of $\mathcal{G}$ if the t-th $N \times N$ diagonal block of $\bar{\boldsymbol{A}}$ equals $\boldsymbol{A}_{\mathcal{V}}[t]$.

In general, there exist multiple graph extensions for a given time-varying graph. This is because only the diagonal blocks of $\bar{\boldsymbol{A}}$ are dictated by $\{\boldsymbol{A}_{\mathcal{V}}[t]\}_{t=1}^T$, whereas the remaining entries of $\bar{\boldsymbol{A}}$ can be freely selected so long as $\bar{\boldsymbol{A}}$ is a valid adjacency matrix. In the reconstruction problem, one is interested in selecting such off-diagonal entries to capture the space-time dynamics of f. As an example, consider an extended graph with

$$\bar{\boldsymbol{A}} = \text{btridiag}\{\boldsymbol{A}_{\mathcal{V}}[1], \ldots, \boldsymbol{A}_{\mathcal{V}}[T]; \boldsymbol{B}_{\mathcal{T}}[2], \ldots, \boldsymbol{B}_{\mathcal{T}}[T]\} \tag{7}$$

where $\boldsymbol{B}_{\mathcal{T}}[t] \in \mathbb{R}_+^{N \times N}$ connects $\{v_n[t-1]\}_{n=1}^N$ to $\{v_n[t]\}_{n=1}^N$, $t = 2, \ldots, T$. For instance, one can connect each vertex to its neighbors at the previous time instant by setting $\boldsymbol{B}_{\mathcal{T}}[t] = \boldsymbol{A}_{\mathcal{V}}[t-1]$, or one can connect each vertex to its replicas at adjacent time instants by setting $\boldsymbol{B}_{\mathcal{T}}[t]$ to be diagonal. Fig. 1 pictorially illustrates the latter choice.

Notice that the extended graph treats the time dimension just as the spatial dimension. Thus, the flexibility of graphs to convey relational information carries over to the time domain. As a major benefit, this approach lays the grounds for the design of doubly-selective kernels in Section V-A. The extended graph also enables a generalization of the estimators in Section III to reconstruct time-evolving functions. The rest of this section develops two KRR estimators along these lines.

Consider first the *batch formulation* (see Section II), where all the $\bar{S} := \sum_{t=1}^T S[t]$ samples in $\bar{\boldsymbol{y}} := [\boldsymbol{y}^\top[1], \ldots, \boldsymbol{y}^\top[T]]^\top$ are available, and the goal is to estimate $\bar{\boldsymbol{f}} := [\boldsymbol{f}^\top[1], \ldots, \boldsymbol{f}^\top[T]]^\top$. Directly applying the KRR criterion in (5) to reconstruct $\bar{\boldsymbol{f}}$ on the extended graph $\bar{\mathcal{G}}$ yields

$$\hat{\bar{\boldsymbol{f}}} := \arg\min_{\bar{\boldsymbol{f}}} ||\bar{\boldsymbol{y}} - \bar{\boldsymbol{S}}\bar{\boldsymbol{f}}||_{\boldsymbol{D}_S}^2 + \mu \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^\dagger \bar{\boldsymbol{f}} \tag{8a}$$

where $\bar{\boldsymbol{K}}$ is now a $TN \times TN$ "space-time" kernel matrix to be designed in Section V, $\bar{\boldsymbol{S}} := \text{bdiag}\{\boldsymbol{S}[1], \ldots, \boldsymbol{S}[T]\}$, and $\boldsymbol{D}_S := \text{bdiag}\{S[1]\boldsymbol{I}_{S[1]}, \ldots, S[T]\boldsymbol{I}_{S[T]}\}$. If $\bar{\boldsymbol{K}}$ is invertible, (8a) can be solved in closed form as

$$\hat{\bar{\boldsymbol{f}}} = \bar{\boldsymbol{K}}\bar{\boldsymbol{S}}^\top(\bar{\boldsymbol{S}}\bar{\boldsymbol{K}}\bar{\boldsymbol{S}}^\top + \mu\boldsymbol{D}_S)^{-1}\bar{\boldsymbol{y}}. \tag{8b}$$

Note that if $\bar{\boldsymbol{K}}$ is such that $\bar{\boldsymbol{K}}^\dagger = \text{bdiag}\{\boldsymbol{K}^\dagger[1], \ldots, \boldsymbol{K}^\dagger[T]\}$, where $\boldsymbol{K}[t]$ is an $N \times N$ kernel matrix for $\mathcal{G}$ at time t, then (8a) separates into T sub-problems, each as in (6). This implies that only matrices $\bar{\boldsymbol{K}}^\dagger$ with non-zero entries off its block diagonal are capable of accounting for temporal dynamics.

In the *online formulation* (cf. Section II), one aims to estimate $\boldsymbol{f}[t]$ after the $\bar{S}[t] := \sum_{\tau=1}^{t} S[\tau]$ samples in $\bar{\boldsymbol{y}}[t] := [\boldsymbol{y}^\top[1], \ldots, \boldsymbol{y}^\top[t]]^\top$ become available. Based on these samples, the KRR estimate of $\bar{\boldsymbol{f}}$, denoted as $\hat{\bar{\boldsymbol{f}}}|t]$, is clearly

$$\hat{\bar{\boldsymbol{f}}}|t] := \arg\min_{\bar{\boldsymbol{f}}} ||\bar{\boldsymbol{y}}[t] - \bar{\boldsymbol{S}}[t]\bar{\boldsymbol{f}}||^2_{\boldsymbol{D}_S[t]} + \mu\bar{\boldsymbol{f}}^\top\bar{\boldsymbol{K}}^{-1}\bar{\boldsymbol{f}} \tag{9a}$$

$$= \bar{\boldsymbol{K}}\bar{\boldsymbol{S}}^\top[t](\bar{\boldsymbol{S}}[t]\bar{\boldsymbol{K}}\bar{\boldsymbol{S}}^\top[t] + \mu\boldsymbol{D}_S[t])^{-1}\bar{\boldsymbol{y}}[t]. \tag{9b}$$

where $\bar{\boldsymbol{K}}$ is assumed invertible for simplicity, $\boldsymbol{D}_S[t] := \text{bdiag}\{S[1]\boldsymbol{I}_{S[1]}, \ldots, S[t]\boldsymbol{I}_{S[t]}\}$, and $\bar{\boldsymbol{S}}[t] := [\text{diag}\{\boldsymbol{S}[1], \ldots, \boldsymbol{S}[t]\}, \boldsymbol{0}_{\bar{S}[t]\times(T-t)N}] \in \{0,1\}^{\bar{S}[t]\times TN}$. The estimate in (9) comprises the per slot estimates $\{\hat{\boldsymbol{f}}[\tau|t]\}_{\tau=1}^T$; that is, $\hat{\bar{\boldsymbol{f}}}|t] := [\hat{\boldsymbol{f}}^\top[1|t], \hat{\boldsymbol{f}}^\top[2|t], \ldots, \hat{\boldsymbol{f}}^\top[T|t]]^\top$ with $\hat{\boldsymbol{f}}[\tau|t] = [\hat{f}_1[\tau|t], \ldots, \hat{f}_N[\tau|t]]^\top$, where $\hat{\boldsymbol{f}}[\tau|t]$ (respectively $\hat{f}_n[\tau|t]$) is the KRR estimate of $\boldsymbol{f}[\tau]$ ($f_n[\tau]$) given the observations up to time t. Observe that, with this notation, it follows that

$$\hat{\boldsymbol{f}}[\tau|t] = (\boldsymbol{i}_{T,\tau}^\top \otimes \boldsymbol{I}_N)\hat{\bar{\boldsymbol{f}}}|t] \tag{10}$$

for all t, τ.

Regarding t as the present, (9) therefore provides estimates of past, present, and future values of f. The solution to the online problem formulated in Section II comprises the sequence of *present* KRR estimates for all t, that is, $\{\hat{\boldsymbol{f}}[t|t]\}_{t=1}^T$. This can be obtained by solving (9a) in closed form per t as in (9b) and then applying (10). However, such an approach does not yield a desirable online algorithm since its complexity per time slot is cubic in t (see Remark 1) and therefore increasing with t. For this reason, this approach is not satisfactory since the online problem formulation in Section II requires the complexity per time slot of the desired algorithm to be bounded. An algorithm that does satisfy this bounded-complexity requirement yet provides the exact KRR estimate is developed next for the case where the kernel matrix is any positive definite matrix $\bar{\boldsymbol{K}}$ satisfying

$$\bar{\boldsymbol{K}}^{-1} = \text{btridiag}\{\boldsymbol{D}[1], \ldots, \boldsymbol{D}[T]; \boldsymbol{C}[2], \ldots, \boldsymbol{C}[T]\} \tag{11}$$

for some $N \times N$ matrices $\{\boldsymbol{D}[t]\}_{t=1}^T$ and $\{\boldsymbol{C}[t]\}_{t=2}^T$. Kernels in this important family are designed in Section V. Broader classes of kernels can be accommodated as described in Remark 3.

The process of developing the desired online algorithm involves two steps. The first step expresses (9a) as a weighted least-squares problem amenable to a KF solver. In the second step, the KF is applied to solve such a problem. The first step is accomplished by the following result.

Lemma 1: Let $\bar{\boldsymbol{K}}$ be of the form (11) and let $\hat{\bar{\boldsymbol{f}}}|t]$ be the KRR estimate defined in (9a). If $\{\boldsymbol{P}[\tau]\}_{\tau=2}^T$ and $\{\boldsymbol{\Sigma}[\tau]\}_{\tau=1}^T$ are obtained by Algorithm 1 and $\sigma_e^2[\tau] = \mu S[\tau]\,\forall\tau$, then

$$\hat{\bar{\boldsymbol{f}}}|t] = \arg\min_{\{\boldsymbol{f}[\tau]\}_{\tau=1}^T} \sum_{\tau=1}^{t} \frac{1}{\sigma_e^2[\tau]}||\boldsymbol{y}[\tau] - \boldsymbol{S}[\tau]\boldsymbol{f}[\tau]||^2 + \sum_{\tau=2}^{T} ||\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]||^2_{\boldsymbol{\Sigma}[\tau]} + \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1]. \tag{12}$$

Proof: See Appendix A. ■

Relative to (9a), matrices $\{\boldsymbol{D}[\tau], \boldsymbol{C}[\tau]\}$ in $\bar{\boldsymbol{K}}^{-1}$ have been replaced in (12) with matrices $\{\boldsymbol{\Sigma}[\tau], \boldsymbol{P}[\tau]\}$, which can be found through Algorithm 1. To enable the upcoming interpretation, the following result establishes that $\{\boldsymbol{\Sigma}[\tau]\}_{\tau=1}^T$ are positive definite.

Lemma 2: Let $\bar{\boldsymbol{K}}$ be a positive definite matrix of the form (11) and let $\{\boldsymbol{P}[\tau]\}_{\tau=2}^T$ and $\{\boldsymbol{\Sigma}[\tau]\}_{\tau=1}^T$ be obtained by Algorithm 1. Then, the matrices $\{\boldsymbol{\Sigma}[\tau]\}_{\tau=1}^T$ are positive definite.

Proof: The proof follows by noting that $\boldsymbol{\Sigma}^{-1}[\tau]$ is the Schur complement of a submatrix of $\bar{\boldsymbol{K}}^{-1}$ that comprises its last $T - \tau + 1$ block rows and columns. Since this submatrix is positive definite, so is $\boldsymbol{\Sigma}^{-1}[\tau]$ and therefore $\boldsymbol{\Sigma}[\tau]$. ■

Although no probabilistic assumption is required throughout the derivation of the proposed online algorithm, exploring the connections between (12) and the conventional probabilistic setup for state estimation provides the intuition behind why (12) can be solved through Kalman filtering. To this end, suppose that $\boldsymbol{f}[\tau]$ adheres to the model $\boldsymbol{f}[\tau] = \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1] + \boldsymbol{w}[\tau]$ for $\tau = 2, \ldots, T$, initialized by $\boldsymbol{f}[1] = \boldsymbol{w}[1]$, with zero-mean noise $\boldsymbol{w}[\tau]$ having covariance $\boldsymbol{\Sigma}[\tau]$, and that the observations follow the model $\boldsymbol{y}[\tau] = \boldsymbol{S}[t]\boldsymbol{f}[\tau] + \boldsymbol{e}[\tau]$ for $\tau = 1, \ldots, T$, with $\boldsymbol{e}[\tau]$ zero-mean noise having covariance $\sigma_e^2[\tau]\boldsymbol{I}$. Note that $\boldsymbol{P}[\tau]$ is commonly referred to as the state-transition matrix within this framework. In this scenario, one can easily see that obtaining the maximum a posteriori (MAP) and the minimum mean square error (MMSE) estimators of $\hat{\bar{\boldsymbol{f}}}$ given the observations up to time T when $\{\boldsymbol{w}[\tau], \boldsymbol{e}[\tau]\}_{\tau=1}^T$ are Gaussian distributed reduces to minimizing (12). This link informally reveals that (12) can be minimized using the celebrated KF [30, Ch. 17].

The following theorem formalizes the latter claim. The resulting algorithm, termed KKF, is summarized as Algorithm 2. In the probabilistic KF terminology, step 2 yields the prediction of $\boldsymbol{f}[t]$, step 2 provides the covariance matrix of the prediction error, step 2 yields the Kalman gain, step 2 returns the posterior estimate upon correcting the prediction with the innovations scaled by the Kalman gain, and step 2 finds the error of this posterior estimate.

Theorem 1: Let $\bar{\boldsymbol{K}}$ be of the form (11) and let $\hat{\bar{\boldsymbol{f}}}|t]$ be the KRR estimate defined in (9a). If $\{\boldsymbol{P}[\tau]\}_{\tau=2}^T$ and $\{\boldsymbol{\Sigma}[\tau]\}_{\tau=1}^T$ are obtained by Algorithm 1 and $\sigma_e^2[\tau] = \mu S[\tau]\,\forall\tau$, then the kernel Kalman filter (KKF) in Algorithm 2 returns the sequence $\{\hat{\boldsymbol{f}}[t|t]\}_{t=1}^T$, where $\hat{\boldsymbol{f}}[t|t]$ is given by (10).

Proof: See Appendix B ■.

Recapitulating, given $\bar{\boldsymbol{K}}^{-1}$ in (11), one just has to run Algorithms 1 and 2 to find the online KRR estimate of f given by (10). As in the explanation after Lemma 2, probabilistic notions such

Algorithm 1: Recursion to set the parameters of the KKF.

Input: $\boldsymbol{D}[t], t=1,\ldots,T$, $\boldsymbol{C}[t], t=2,\ldots,T$.

1: **Set** $\boldsymbol{\Sigma}^{-1}[T]=\boldsymbol{D}[T]$
2: **for** $t=T, T-1,\ldots,2$ **do**
3: $\quad\boldsymbol{P}[t]=-\boldsymbol{\Sigma}[t]\boldsymbol{C}[t]$
4: $\quad\boldsymbol{\Sigma}^{-1}[t-1]=\boldsymbol{D}[t-1]-\boldsymbol{P}^{\top}[t]\boldsymbol{\Sigma}^{-1}[t]\boldsymbol{P}[t]$

Output: $\boldsymbol{\Sigma}[t], t=1,\ldots,T$, $\boldsymbol{P}[t], t=2,\ldots,T$

Algorithm 2: Kernel Kalman filter (KKF).

Input: $\{\boldsymbol{\Sigma}[t]\in\mathbb{S}_{+}^{N}\}_{t=1}^{T}$, $\{\boldsymbol{P}[t]\in\mathbb{R}^{N\times N}\}_{t=2}^{T}$, $\{\boldsymbol{y}[t]\in\mathbb{R}^{S[t]}\}_{t=1}^{T}$, $\{\boldsymbol{S}[t]\in\{0,1\}^{S[t]\times N}\}_{t=1}^{T}$, $\{\sigma_e^2[t]>0\}_{t=1}^{T}$.

1: **Set** $\hat{\boldsymbol{f}}[0|0]=\boldsymbol{0}$, $\boldsymbol{M}[0|0]=\boldsymbol{0}$, $\boldsymbol{P}[1]=\boldsymbol{0}$
2: **for** $t=1,\ldots,T$ **do**
3: $\quad\hat{\boldsymbol{f}}[t|t-1]=\boldsymbol{P}[t]\hat{\boldsymbol{f}}[t-1|t-1]$
4: $\quad\boldsymbol{M}[t|t-1]=\boldsymbol{P}[t]\boldsymbol{M}[t-1|t-1]\boldsymbol{P}^{\top}[t]+\boldsymbol{\Sigma}[t]$
5: $\quad\boldsymbol{G}[t]=\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t]$
$\qquad\times(\sigma_e^2[t]\boldsymbol{I}+\boldsymbol{S}[t]\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t])^{-1}$
6: $\quad\hat{\boldsymbol{f}}[t|t]=\hat{\boldsymbol{f}}[t|t-1]+\boldsymbol{G}[t](\boldsymbol{y}[t]-\boldsymbol{S}[t]\hat{\boldsymbol{f}}[t|t-1])$
7: $\quad\boldsymbol{M}[t|t]=(\boldsymbol{I}-\boldsymbol{G}[t]\boldsymbol{S}[t])\boldsymbol{M}[t|t-1]$

Output: $\hat{\boldsymbol{f}}[t|t], t=1,\ldots,T$; $\boldsymbol{M}[t], t=1,\ldots,T$.

as mean, covariance, statistical independence, or mean-square error, can be used to elucidate connections with the classical KF. Nonetheless, they are not required at any point of the derivation, which involves fully deterministic notions. Furthermore, the proposed KKF does not explicitly involve any state-space model, which is a major novelty and surprising result of the present paper.

The proposed KKF generalizes the probabilistic KF since the latter is recovered upon setting $\bar{\boldsymbol{K}}$ to be the covariance matrix of $\bar{\boldsymbol{f}}$ in the probabilistic setup alluded to previously. It is therefore natural that the assumptions required by the probabilistic KF are stronger than those involved in the KKF. Specifically, in the probabilistic KF, $\boldsymbol{f}[t]$ must adhere to a linear state-space model with known transition matrix $\boldsymbol{P}[t]$, where the state noise $\boldsymbol{w}[t]$ is uncorrelated over time and has known covariance matrix $\boldsymbol{\Sigma}[t]$. Furthermore, the observation noise $\boldsymbol{e}[t]$ must be uncorrelated over time and have known covariance matrix. Correspondingly, the performance guarantees of the probabilistic KF are also stronger: the resulting estimate is optimal in the mean-square error sense among all linear estimators. Furthermore, if $\boldsymbol{w}[t]$ and $\boldsymbol{y}[t]$ are jointly Gaussian, $t=1,\ldots,T$, then the probabilistic KF estimate is optimal in the mean-square error sense among all (not necessarily linear) estimators. In contrast, the requirements of the proposed KKF are much weaker since they only specify that f must evolve smoothly with respect to a given extended graph. As expected, the performance guarantees are similarly weaker; see e.g. [25, Ch. 5]. However, since the KKF generalizes the probabilistic KF, the reconstruction performance of the former for judiciously selected $\bar{\boldsymbol{K}}$ cannot be worse than the reconstruction performance of the latter for any given criterion. The caveat, however, is that such a selection is not necessarily easy.

Remark 1: Algorithm 2 requires $\mathcal{O}(N^3)$ operations per time slot, whereas the complexity of evaluating (9b) for the t-th time slot is at least $\mathcal{O}(\bar{S}^3[t])$, which increases with t and becomes eventually prohibitive. For large t, Algorithm 2 is computationally more efficient than a *single* plain evaluation of (9b): whereas the overall complexity of the former is $\mathcal{O}(tN^3)$, the latter is $\mathcal{O}(NT\bar{S}^2[t])$, which e.g. for constant $S[t]=S$ is $\mathcal{O}(NTt^2S^2)$.

Remark 2: Algorithm 2 provides estimates of the form $\hat{\boldsymbol{f}}[t|t]$ and $\hat{\boldsymbol{f}}[t|t-1]$. To obtain estimates $\hat{\boldsymbol{f}}[t|t']$ for $t>t'+1$, one may set $\mathcal{S}[\tau]=\emptyset$ for $\tau\geqslant t'+1$ and execute Algorithm 2 up to time t. Conversely, to obtain estimates $\hat{\boldsymbol{f}}[t|t']$ for which $t<t'$, one may extend Algorithm 2 along the lines of the Kalman smoother [31] to also operate backwards.

Remark 3: Similar to the probabilistic KF, which requires the inverse covariance matrix of $\bar{\boldsymbol{f}}$ to be block tridiagonal, the proposed KKF requires the inverse kernel matrix to be of the form (11). Fortunately, it is straightforward to extend both algorithms to accommodate inverse covariance or kernel matrices with any number of non-zero diagonals at the price of increasing the time interval between consecutive estimates. To illustrate such an approach, suppose that $\bar{\boldsymbol{K}}^{-1}$ is not block tridiagonal when blocks are of size $N\times N$, but it is block tridiagonal if blocks are of size $2N\times 2N$. In such a case, one can use the proposed KKF to estimate $\{\boldsymbol{f}'[t']\}_{t'=1}^{T/2}$, where $\boldsymbol{f}'[t']:=[\boldsymbol{f}^{\top}[2t'-1],\boldsymbol{f}^{\top}[2t']]^{\top}\in\mathbb{R}^{2N}$, just by replacing $\boldsymbol{y}[t]$ with $\boldsymbol{y}'[t']:=[\boldsymbol{y}^{\top}[2t'-1],\boldsymbol{y}^{\top}[2t']]^{\top}$, $\boldsymbol{S}[t]$ with $\boldsymbol{S}'[t']:=\mathrm{bdiag}\,\{\boldsymbol{S}[2t'-1],\boldsymbol{S}[2t']\}$, and $\boldsymbol{e}[t]$ with $\boldsymbol{e}'[t']:=[\boldsymbol{e}^{\top}[2t'-1],\boldsymbol{e}^{\top}[2t']]^{\top}$, $t'=1,\ldots,T/2$. Note that the sampling interval associated with the index t' is twice the one associated with t.

V. Design of Space-Time Kernels

Section IV assumed that the kernel matrix $\bar{\boldsymbol{K}}$ is given and described no methodology to address its design. An immediate approach is to mimic the Laplacian kernels of Section III by setting $\bar{\boldsymbol{K}}=r^{\dagger}(\bar{\boldsymbol{L}})$, where $\bar{\boldsymbol{L}}:=\mathrm{diag}\{\bar{\boldsymbol{A}}\boldsymbol{1}\}-\bar{\boldsymbol{A}}$ denotes the Laplacian matrix of the extended graph. Unfortunately, such a design prevents separate control of the spatial and temporal variability of the estimates, thus limiting the user's ability to flexibly account for spatial and temporal information. For instance, sampling intervals that are small relative to the time dynamics of f, meaning that f does not vary significantly between samples $t-1$ and t, favors estimates that sacrifice spatial smoothness to increase temporal smoothness.

This section proposes families of space-time kernels for which temporal and spatial smoothness can be separately tuned. Section V-A describes designs for time-invariant topologies, whereas Section V-B deals with the time-varying case.

A. Doubly-Selective Space-Time Kernels

In Section III, the frequency interpretation of (3) proved decisive to interpret and design Laplacian kernels for reconstructing time-invariant functions. Introducing the time dimension in Section IV prompts an analogous methodology, where kernels are specified in a bidimensional plane of spatio-temporal

frequency.[3] This section accomplishes this task by generalizing the Laplacian kernels from Section III in such a way that the weight that the regularizers $\rho(\bar{\boldsymbol{f}}) = \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^\dagger \bar{\boldsymbol{f}}$ associated with the proposed kernels confer to each spatial and temporal frequency component of $\bar{\boldsymbol{f}}$ can be separately prescribed. Throughout this section, a time-invariant topology will be assumed, i.e., $\boldsymbol{A}_\mathcal{V}[t] = \boldsymbol{A}_\mathcal{V}$, $t = 1, \ldots, T$.

Clearly, (4a) can be rewritten as $\rho_{\text{LKR}}(\boldsymbol{f}) = \boldsymbol{r}^\top(\check{\boldsymbol{f}} \odot \check{\boldsymbol{f}})$ for $\check{\boldsymbol{f}} := \boldsymbol{U}_\mathcal{V}^\top \boldsymbol{f}$ the frequency transform of $\boldsymbol{f}$ and $\boldsymbol{r} := [r(\lambda_1^\mathcal{V}), \ldots, r(\lambda_N^\mathcal{V})]^\top$. One can separately weight each frequency component by selecting $\boldsymbol{r}$, which can therefore be thought of as the "frequency response" of the regularizer. For instance, one may promote low-pass estimates by setting the first entries of $\boldsymbol{r}$ to low values and the rest to high values.

Inspired by this view, one may seek kernels $\bar{\boldsymbol{K}}$ for which

$$\rho(\bar{\boldsymbol{f}}) = \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^\dagger \bar{\boldsymbol{f}} = \text{Tr}\left(\boldsymbol{R}^\top(\check{\check{\boldsymbol{F}}} \odot \check{\check{\boldsymbol{F}}})\right) \tag{13}$$

where $\boldsymbol{R}$ and $\check{\check{\boldsymbol{F}}}$ are $N \times T$ matrices to be specified later respectively containing the frequency response of the regularizer and the bidimensional transform of f. The $(\check{n}, \check{t})$-th entry of these matrices corresponds to the $\check{n}$-th spatial frequency and $\check{t}$-th temporal frequency. Kernels satisfying the second equality in (13) will be termed *doubly (frequency) selective*. Such kernels preserve the flexibility of their counterparts for time-invariant functions. For instance, if $\bar{\boldsymbol{K}}$ promotes doubly low-pass estimates, then the top left entries of $\boldsymbol{R}$ are small whereas the rest are large.

To determine the form of a doubly-selective kernel, let $\bar{\boldsymbol{F}} := [\boldsymbol{f}[1], \ldots, \boldsymbol{f}[T]]$ and recall that any linear bidimensional transform can be expressed as $\check{\check{\boldsymbol{F}}} := \boldsymbol{U}_\mathcal{V}^\top \bar{\boldsymbol{F}} \boldsymbol{U}_\mathcal{T}$, where the $N \times N$ matrix $\boldsymbol{U}_\mathcal{V}$ and the $T \times T$ matrix $\boldsymbol{U}_\mathcal{T}$ stand for orthogonal transformations along space and time, respectively. On the other hand, vectorizing the rightmost term of (13) yields

$$\rho(\bar{\boldsymbol{f}}) = \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^\dagger \bar{\boldsymbol{f}} = \check{\check{\boldsymbol{f}}}^\top \,\text{diag}\,\{\boldsymbol{r}\}\, \check{\check{\boldsymbol{f}}} \tag{14}$$

where $\boldsymbol{r} := \text{vec}\{\boldsymbol{R}\}$ and

$$\check{\check{\boldsymbol{f}}} := \text{vec}\{\check{\check{\boldsymbol{F}}}\} = \text{vec}\{\boldsymbol{U}_\mathcal{V}^\top \bar{\boldsymbol{F}} \boldsymbol{U}_\mathcal{T}\} = (\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})^\top \bar{\boldsymbol{f}}. \tag{15}$$

Any doubly-selective kernel, or equivalently any kernel satisfying the second equality of (14), is therefore of the form

$$\bar{\boldsymbol{K}}^\dagger = (\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})\,\text{diag}\,\{\boldsymbol{r}\}\,(\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})^\top \tag{16}$$

for some orthogonal $T \times T$ matrix $\boldsymbol{U}_\mathcal{T}$, some orthogonal $N \times N$ matrix $\boldsymbol{U}_\mathcal{V}$, and some entrywise non-negative vector $\boldsymbol{r}$.

Expression (16) provides the general form of a doubly-selective kernel, but a specific construction for $\boldsymbol{U}_\mathcal{T}$, $\boldsymbol{U}_\mathcal{V}$, and $\boldsymbol{r}$ capturing the spatiotemporal dynamics of f is still required. The next procedure serves this purpose by paralleling the approach in Section III. This involves the following two steps. **S1**: Since a Laplacian kernel matrix shares eigenvectors with the Laplacian matrix, one should construct an extended graph $\bar{\mathcal{G}}$ so that its Laplacian matrix $\bar{\boldsymbol{L}}$ is diagonalizable by a matrix of the form $\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V}$ for some orthogonal $\boldsymbol{U}_\mathcal{T} \in \mathbb{R}^{T \times T}$ and $\boldsymbol{U}_\mathcal{V} \in \mathbb{R}^{N \times N}$. **S2**: One must design a spectral weight map r to obtain the eigenvalues of $\bar{\boldsymbol{K}}$ from those of $\bar{\boldsymbol{L}}$.

Regarding S1, an explicit construction of an extended graph whose Laplacian matrix is diagonalizable by a matrix of the form $\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V}$ with orthogonal $\boldsymbol{U}_\mathcal{T} \in \mathbb{R}^{T \times T}$ and $\boldsymbol{U}_\mathcal{V} \in \mathbb{R}^{N \times N}$ is provided next. To this end, consider the extended adjacency matrix

$$\bar{\boldsymbol{A}} = \boldsymbol{A}_\mathcal{T} \oplus \boldsymbol{A}_\mathcal{V} \tag{17}$$

where $\boldsymbol{A}_\mathcal{V}$ is the given adjacency matrix of $\mathcal{G}$ and the $T \times T$ adjacency matrix $\boldsymbol{A}_\mathcal{T}$ is selected to capture temporal dynamics. Specifically, with $\bar{\boldsymbol{A}}$ as in (17), the definition of extended adjacency matrix in Section IV dictates that the weight of the edge $(v_{n_1}[t], v_{n_2}[t])$ for all t is given by the (n_1, n_2)-th entry of $\boldsymbol{A}_\mathcal{V}$, whereas the weight of the edge $(v_n[t_1], v_n[t_2])$ for all n is given by the (t_1, t_2)-th entry of $\boldsymbol{A}_\mathcal{T}$. A simple choice for $\boldsymbol{A}_\mathcal{T}$ will be described later. Note that (17) differs from *Kronecker graphs* [33], for which $\bar{\boldsymbol{A}} = \boldsymbol{A}_\mathcal{T} \otimes \boldsymbol{A}_\mathcal{V}$, although it can be interpreted as the *Cartesian graph* of $\mathcal{V}$ and $\{1, \ldots, T\}$ [34], [35]. Cartesian graphs have been considered in the graph signal processing literature for graph filtering and Fourier transforms of time-varying functions [35], but not for signal reconstruction.

With $\bar{\boldsymbol{A}}$ as in (17), it can be readily seen that $\bar{\boldsymbol{L}} := \text{diag}\{\bar{\boldsymbol{A}}\mathbf{1}\} - \bar{\boldsymbol{A}} = \boldsymbol{L}_\mathcal{T} \oplus \boldsymbol{L}_\mathcal{V}$, where $\boldsymbol{L}_\mathcal{T} := \text{diag}\{\boldsymbol{A}_\mathcal{T}\mathbf{1}\} - \boldsymbol{A}_\mathcal{T}$ and $\boldsymbol{L}_\mathcal{V} := \text{diag}\{\boldsymbol{A}_\mathcal{V}\mathbf{1}\} - \boldsymbol{A}_\mathcal{V}$ are the Laplacian matrices associated with $\boldsymbol{A}_\mathcal{T}$ and $\boldsymbol{A}_\mathcal{V}$, respectively. If now one sets $\boldsymbol{U}_\mathcal{T}$ and $\boldsymbol{U}_\mathcal{V}$ to be respectively the eigenvector matrices of $\boldsymbol{L}_\mathcal{T}$ and $\boldsymbol{L}_\mathcal{V}$, then $\boldsymbol{L}_\mathcal{T} = \boldsymbol{U}_\mathcal{T}\,\text{diag}\,\{\boldsymbol{\lambda}_\mathcal{T}\}\,\boldsymbol{U}_\mathcal{T}^\top$ and $\boldsymbol{L}_\mathcal{V} = \boldsymbol{U}_\mathcal{V}\,\text{diag}\,\{\boldsymbol{\lambda}_\mathcal{V}\}\,\boldsymbol{U}_\mathcal{V}^\top$ for some $\boldsymbol{\lambda}_\mathcal{T}$ and $\boldsymbol{\lambda}_\mathcal{V}$. This implies that

$$\begin{aligned}\bar{\boldsymbol{L}} &= (\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})\,[\text{diag}\,\{\boldsymbol{\lambda}_\mathcal{T}\} \oplus \text{diag}\,\{\boldsymbol{\lambda}_\mathcal{V}\}]\,(\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})^\top \\ &= (\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})\,\text{diag}\,\{\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N + \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V}\}\,(\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})^\top.\end{aligned}$$

This expression reveals that the graph extension proposed in (17) indeed satisfies the objective of S1, which requires the eigenvector matrix of $\bar{\boldsymbol{L}}$ to be of the form $\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V}$. Thus, it is always possible to construct a graph extension satisfying the goal of S1.

For S2, one must construct a spectral map r that yields $\boldsymbol{r}$ upon entrywise application to $\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N + \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V}$. To separately control the frequency response along the spatial and temporal frequencies $\lambda_\mathcal{V}$ and $\lambda_\mathcal{T}$, such a map must take two arguments as $r(\lambda_\mathcal{T}, \lambda_\mathcal{V})$. This results in $\boldsymbol{r} = r(\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N, \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V})$ and (16) becomes

$$\begin{aligned}\bar{\boldsymbol{K}}^\dagger = (\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V}) \\ \times \text{diag}\,\{r(\boldsymbol{\lambda}_\mathcal{T} \otimes \mathbf{1}_N, \mathbf{1}_T \otimes \boldsymbol{\lambda}_\mathcal{V})\}\,(\boldsymbol{U}_\mathcal{T} \otimes \boldsymbol{U}_\mathcal{V})^\top.\end{aligned} \tag{18}$$

Kernels of this form will be referred to as *Kronecker space-time* kernels. The transformation r can be selected in several ways. For instance, the immediate construction at the beginning of Section V is recovered for $r(\lambda_\mathcal{T}, \lambda_\mathcal{V}) = r(\lambda_\mathcal{T} + \lambda_\mathcal{V})$, with $r(\lambda)$ a one-dimensional spectral weight map such as the ones in Table I. Another possibility is to focus on separable maps of the form $r(\lambda_\mathcal{T}, \lambda_\mathcal{V}) = r_\mathcal{T}(\lambda_\mathcal{T}) r_\mathcal{V}(\lambda_\mathcal{V})$ where $r_\mathcal{T}$ and $r_\mathcal{V}$ denote one-dimensional spectral maps. The resulting Kronecker kernel

[3] See [32] for graph filter designs in this domain.

can expressed as[4]

$$\bar{\boldsymbol{K}} = \boldsymbol{K}_{\mathcal{T}} \otimes \boldsymbol{K}_{\mathcal{V}} \tag{19}$$

where $\boldsymbol{K}_{\mathcal{T}} := \boldsymbol{U}_{\mathcal{T}} \operatorname{diag}\{r_{\mathcal{T}}(\boldsymbol{\lambda}_{\mathcal{T}})\}\boldsymbol{U}_{\mathcal{T}}^{\top}$ and $\boldsymbol{K}_{\mathcal{V}} := \boldsymbol{U}_{\mathcal{V}} \operatorname{diag}\{r_{\mathcal{V}}(\boldsymbol{\lambda}_{\mathcal{V}})\}\boldsymbol{U}_{\mathcal{V}}^{\top}$. For example, doubly bandlimited estimates can be obtained by setting both $\boldsymbol{K}_{\mathcal{T}}$ and $\boldsymbol{K}_{\mathcal{V}}$ to be bandlimited kernels (Table I). A further possibility is to consider maps of the form $r(\lambda_{\mathcal{T}}, \lambda_{\mathcal{V}}) = r_{\mathcal{T}}(\lambda_{\mathcal{T}}) + r_{\mathcal{V}}(\lambda_{\mathcal{V}})$, which clearly result in kernels of the form

$$\bar{\boldsymbol{K}}^{\dagger} = \boldsymbol{K}_{\mathcal{T}}^{\dagger} \oplus \boldsymbol{K}_{\mathcal{V}}^{\dagger}. \tag{20}$$

To sum up, the proposed Kronecker kernels arise from an intuitive graph extension and can afford flexible adjustment of their frequency response. Unfortunately, not any Kronecker kernel is suitable for the online algorithm in Section IV since the latter requires the inverse of the kernel matrix $\bar{\boldsymbol{K}}$ to be block tridiagonal. The rest of this section describes a subfamily of Kronecker kernels that is suitable for this algorithm.

Clearly, in order for $\bar{\boldsymbol{K}}^{\dagger}$ as in (19) or (20) to be block tridiagonal, it is necessary that $\boldsymbol{K}_{\mathcal{T}}^{\dagger}$ be tridiagonal, i.e., the (t, t')-th entry of $\boldsymbol{K}_{\mathcal{T}}^{\dagger}$ must be zero if $|t - t'| > 1$. Such a $\boldsymbol{K}_{\mathcal{T}}^{\dagger}$ can be obtained if, for instance, one sets the (t, t')-th entry of $\boldsymbol{A}_{\mathcal{T}}$ to be 0 unless $|t - t'| = 1$. In this extended graph construction, vertex $v_n[t]$, $1 < t < T$, is connected to $v_n[t-1]$ and $v_n[t+1]$, which are its replicas in adjacent time slots. For $\boldsymbol{K}_{\mathcal{T}}^{\dagger}$ to be tridiagonal, one may set $r_{\mathcal{T}}(\lambda_{\mathcal{T}}) = \lambda_{\mathcal{T}} + \epsilon$, where $\epsilon > 0$ ensures that $\boldsymbol{K}_{\mathcal{T}}$ is invertible.

Thus, the price to be paid for an online implementation with the KKF from Section IV is limited flexibility in specifying the temporal frequency response. Note that this is not an intrinsic limitation of the proposed algorithm, but it is inherent to the classical KF as well; just recall that the latter assumes vector autoregressive processes of order 1. Remark 3 describes a technique to alleviate this limitation. In any case, the temporal frequency response of a kernel for which $(\boldsymbol{A}_{\mathcal{T}})_{t,t'} = \delta[|t - t'| - 1]$ can be obtained analytically by approximating the resulting Laplacian $\boldsymbol{L}_{\mathcal{T}}$ for sufficiently large T with a circulant matrix. This implies that (i) the eigenvectors in $\boldsymbol{U}_{\mathcal{T}}$ are approximately those in the conventional Fourier basis and therefore the notion of temporal frequency embodied in $\boldsymbol{U}_{\mathcal{T}}$ preserves its conventional meaning; and (ii), upon applying [17, Example 3], the resulting frequency response is low pass. Both (i) and (ii) are intuitively reasonable. Thus, although the KKF solves only a subset of KRR problems, this subset is of practical relevance.

Remark 4: In this paper, the rows of $\bar{\boldsymbol{F}}$ can be thought of as graph functions over a graph with adjacency matrix $\boldsymbol{A}_{\mathcal{T}}$, whereas the columns of $\bar{\boldsymbol{F}}$ can be thought of as graph functions over the graph with adjacency matrix $\boldsymbol{A}_{\mathcal{V}}$. In principle, each column of $\bar{\boldsymbol{F}}$ does not need to correspond to a different time instant, but e.g. to a different movie in a recommender system application. The estimators (8a)-(9b) can therefore be used for matrix completion upon properly constructing an extended graph and graph kernel matrix. Towards this end, the space-time kernels defined in (19) and (20) readily generalize to space-space kernels that promote smoothness over both graphs.

[4]The notion of Kronecker kernels together with (19) shows up in the literature of pairwise classification [34], but the resemblance is merely illusional since the underlying kernel is a function of two *pairs* of vertices.

B. Space-Time Kernels for Time-Varying Topologies

For time-invariant topologies, Section V-A proposed kernels that can be designed and interpreted on a two-dimensional frequency plane. This section deals with changing topologies, for which no bidimensional frequency notion can be defined.

To recognize this claim, suppose that $\boldsymbol{A}_{\mathcal{V}}[t] = \boldsymbol{A}_{\mathcal{V}}$ remains constant over t and recall that $\boldsymbol{u}_{\check{n}}^{\mathcal{V}}$ is the $\check{n}$-th eigenvector of $\boldsymbol{L}_{\mathcal{V}}$ or, equivalently, the $\check{n}$-th column of $\boldsymbol{U}_{\mathcal{V}}$. In this case, a bidimensional transform exists and can be expressed as $\breve{\bar{\boldsymbol{F}}} := \boldsymbol{U}_{\mathcal{V}}^{\top}\bar{\boldsymbol{F}}\boldsymbol{U}_{\mathcal{T}}$, whose $(\check{n}, \check{t})$-th entry corresponds to the $\check{n}$-th spatial frequency and $\check{t}$-th temporal frequency; see section V-A. Fundamentally, the precise meaning of the latter statement is that $(\breve{\bar{\boldsymbol{F}}})_{\check{n},\check{t}}$ is the $\check{t}$-th temporal frequency component of the $\check{n}$-th spatial frequency component of f, i.e., the $\check{t}$-th temporal frequency component of the time series $\{\check{f}_{\check{n}}[t] := (\boldsymbol{u}_{\check{n}}^{\mathcal{V}})^{\top}\boldsymbol{f}[t]\}_{t=1}^{T}$, which is the time evolution of the $\check{n}$-th *spatial* frequency component of f. However, for changing topologies one cannot generally conceive the temporal evolution of a specific spatial frequency component since the eigenvectors of $\boldsymbol{L}_{\mathcal{V}}[t]$ generally differ from those of $\boldsymbol{L}_{\mathcal{V}}[t']$, which precludes any natural definition of the aforementioned sequence and therefore of a bidimensional frequency transform. Nonetheless, it is shown next that the notion of spatial frequency per slot t can still be utilized to design space-time kernels for time-varying topologies.

To this end, consider the extended graph defined by (7) for arbitrary $\boldsymbol{B}_{\mathcal{T}}[t] \in \mathbb{R}_{+}^{N\times N}$. It then follows that

$$\begin{aligned}\bar{\boldsymbol{L}} :=\ & \operatorname{diag}\{\bar{\boldsymbol{A}}\mathbf{1}\} - \bar{\boldsymbol{A}} = \operatorname{bdiag}\{\boldsymbol{L}_{\mathcal{V}}[1], \ldots, \boldsymbol{L}_{\mathcal{V}}[T]\} \\ & + \operatorname{btridiag}\big\{\operatorname{diag}\{\boldsymbol{b}_{\mathcal{T}}[1]\}, \ldots, \operatorname{diag}\{\boldsymbol{b}_{\mathcal{T}}[T]\}; \\ & - \boldsymbol{B}_{\mathcal{T}}[2], \ldots, -\boldsymbol{B}_{\mathcal{T}}[T]\big\}\end{aligned} \tag{21}$$

where

$$\boldsymbol{b}_{\mathcal{T}}[t] := \begin{cases} \boldsymbol{B}_{\mathcal{T}}^{\top}[2]\mathbf{1} & \text{if } t = 1 \\ (\boldsymbol{B}_{\mathcal{T}}^{\top}[t+1] + \boldsymbol{B}_{\mathcal{T}}[t])\mathbf{1} & \text{if } 1 < t < T \\ \boldsymbol{B}_{\mathcal{T}}[T]\mathbf{1} & \text{if } t = T. \end{cases}$$

The rationale behind this graph extension is that, for $\bar{\boldsymbol{L}}$ as in (21) and diagonal $\{\boldsymbol{B}_{\mathcal{T}}[t]\}_{t=1}^{T}$, one can show that

$$\begin{aligned}\bar{\boldsymbol{f}}^{\top}\bar{\boldsymbol{L}}\bar{\boldsymbol{f}} =\ & \sum_{t=1}^{T} \boldsymbol{f}^{\top}[t]\boldsymbol{L}_{\mathcal{V}}[t]\boldsymbol{f}[t] \\ & + \sum_{t=2}^{T} (\boldsymbol{f}[t] - \boldsymbol{f}[t-1])^{\top}\boldsymbol{B}_{\mathcal{T}}[t](\boldsymbol{f}[t] - \boldsymbol{f}[t-1]).\end{aligned} \tag{22}$$

When $\bar{\boldsymbol{f}}^{\top}\bar{\boldsymbol{L}}\bar{\boldsymbol{f}}$ is used as a regularizer, the first and second sums on the right-hand side of (22) respectively penalize spatial and temporal variations. As a special case, if one sets $\boldsymbol{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\boldsymbol{I}$ $\forall t$ for some $b_{\mathcal{T}} > 0$, the second sum becomes $b_{\mathcal{T}} \sum_{t=2}^{T} ||\boldsymbol{f}[t] - \boldsymbol{f}[t-1]||_2^2$, which would promote estimates with small changes over time.

Applying the notion of Laplacian kernels along the spatial dimension (see Section III), but not along time, suggests generalizing (22) to obtain the regularizer

$$\rho(\bar{\boldsymbol{f}}) = \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^\dagger \bar{\boldsymbol{f}} = \sum_{t=1}^{T} \boldsymbol{f}^\top[t]\boldsymbol{K}_{\mathcal{V}}^\dagger[t]\boldsymbol{f}[t] + \sum_{t=2}^{T} (\boldsymbol{f}[t]-\boldsymbol{f}[t-1])^\top \boldsymbol{B}_{\mathcal{T}}[t](\boldsymbol{f}[t]-\boldsymbol{f}[t-1]) \tag{23}$$

where $\boldsymbol{K}_{\mathcal{V}}^\dagger[t] = r_t(\boldsymbol{L}_{\mathcal{V}}[t])$, $t = 1, \ldots, T$, for $\{r_t\}_{t=1}^T$ a collection of user-selected spectral maps such as those in Table I. In that case, (23) corresponds to the kernel matrix

$$\bar{\boldsymbol{K}}^\dagger = \text{bdiag}\left\{\boldsymbol{K}_{\mathcal{V}}^\dagger[1], \ldots, \boldsymbol{K}_{\mathcal{V}}^\dagger[T]\right\} + \text{btridiag}\left\{\text{diag}\left\{\boldsymbol{b}_{\mathcal{T}}[1]\right\}, \ldots, \text{diag}\left\{\boldsymbol{b}_{\mathcal{T}}[T]\right\}; -\boldsymbol{B}_{\mathcal{T}}[2], \ldots, -\boldsymbol{B}_{\mathcal{T}}[T]\right\}. \tag{24}$$

Although kernels of this form do not offer a frequency-domain control of reconstruction along time, they still enjoy the spatial flexibility of the kernels in Section V-A.

Remark 5: To guarantee that $\bar{\boldsymbol{K}}^\dagger$ in (24) qualifies for online implementation, it suffices to guarantee that $\bar{\boldsymbol{K}}$ is invertible since it is already block tridiagonal. This holds e.g. if $\boldsymbol{K}_{\mathcal{V}}[t]$ is invertible for all t.

VI. Simulated Tests

This section compares the performance of the proposed schemes with state-of-the-art alternatives and illustrates some of the trade-offs inherent to time-varying function reconstruction through real-data experiments. The source code for the simulations is available at the authors' websites.

Unless otherwise stated, the compared estimators include distributed least squares reconstruction (DLSR) [21] with step size μ_{DLSR} and parameter β_{DLSR}; the least mean-squares (LMS) algorithm in [22] with step size μ_{LMS}; the bandlimited instantaneous estimator (BL-IE), which results from applying [12]–[14] separately per t; the KRR instantaneous estimator (KRR-IE) in (6) with a diffusion kernel with parameter σ; and the proposed KKF (Algorithms 1 and 2) with kernel given by (24) for $\boldsymbol{B}_{\mathcal{T}}[T] = b_{\mathcal{T}}\boldsymbol{I}$ and $\boldsymbol{K}_{\mathcal{V}}[t]$ a diffusion kernel with parameter σ. DLSR, LMS, and BL-IE also use a bandwidth parameter B. The parameters of competing algorithms, such as μ_{LMS} and B, were set to values approximately achieving the best performance. Yet, the latter is not highly sensitive to the choice of B within a wide interval since most of the spatial signals $\boldsymbol{f}[t]$ utilized turn out not to be markedly bandlimited.

The first data set comprises hourly temperature measurements at $N = 109$ stations across the continental U.S. in 2010 [36]. Temperature reconstruction has been extensively employed in the literature to analyze the performance of inference tools over graphs (see e.g. [18], [19], [21]). A time-invariant graph was constructed following the approach in [19], which relies on geographical distances, with 7 nearest neighbors. Function $f_n[t]$ represents the temperature at the n-th station and t-th sampling instant. In the first experiment, the latter corresponds to the t-th hour, whereas for the rest, it corresponds to the temperature at 12:00 PM of the t-th day.

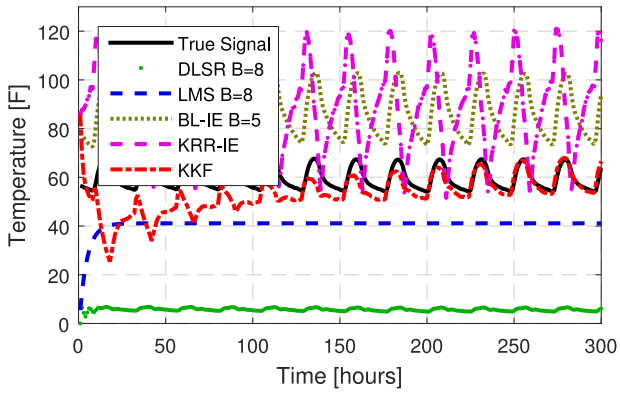

Fig. 2. True temperature and estimates across time at a randomly picked unobserved station ($\mu = 10^{-7}$, $\sigma = 1.8$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 2$, $S = 44$).

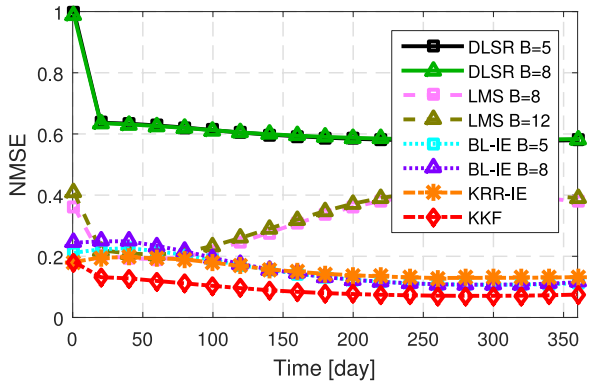

Fig. 3. NMSE of daily temperature estimates over 2010 ($\mu = 10^{-7}$, $\sigma = 1.8$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 2$, $S = 44$).

Fig. 2 depicts the true temperature at an unobserved randomly picked station over the first 300 hours of 2010 along with its estimates for a typical realization of the time-invariant sampling set $\mathcal{S} = \mathcal{S}[t]$, $\forall t$, drawn uniformly at random within all sampling sets with $S = 44$ elements.[5] Different from instantaneous alternatives, whose error does not decrease with time, KKF is observed to successfully leverage time dynamics to track the temperature at the unobserved station. On the other hand, DLSR and LMS are unable to track the rapid variations of f since their design assumes slowly changing functions.

The next experiments compare the cumulative normalized mean-square error (NMSE), defined as

$$\text{NMSE}(t, \{\mathcal{S}[\tau]\}_{\tau=1}^t) := \frac{\sum_{\tau=1}^t ||\boldsymbol{S}^c[\tau](\boldsymbol{f}[\tau] - \hat{\boldsymbol{f}}[\tau|\tau])||_2^2}{\sum_{\tau=1}^t ||\boldsymbol{S}^c[\tau]\boldsymbol{f}[\tau]||_2^2}$$

where $\boldsymbol{S}^c[\tau]$ is an $N - S[\tau] \times N$ matrix comprising the rows of $\boldsymbol{I}_N$ whose indices are not in $\mathcal{S}[t]$.

Fig. 3 shows the NMSE for $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, averaged over all possible $\mathcal{S}$ with $S = 44$ elements. It is observed that the instantaneous estimators outperform DLSR and LMS, which can only

[5]Note that the performance of all methods would improve if one could decide which sampling set to use.

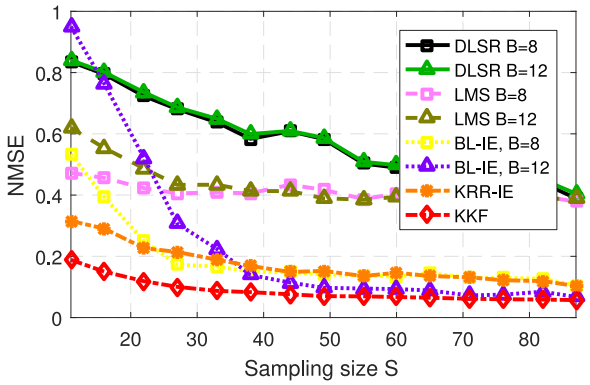

Fig. 4. NMSE for increasing sampling size ($\mu = 10^{-7}$, $\sigma = 1.6$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 2$).

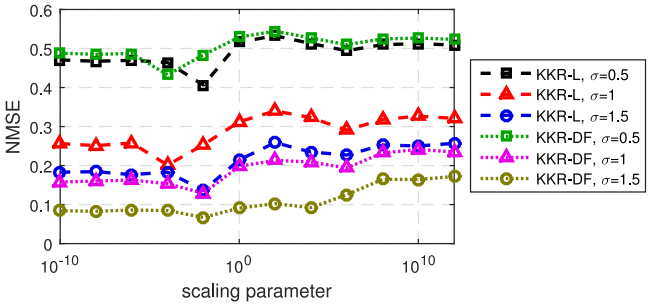

Fig. 5. NMSE for different kernels versus scale parameter $b_{\mathcal{T}}$ ($\mu = 10^{-7}$).

cope with slow variations of f. Furthermore, the error of KKF is nearly half the error of the nearest alternative, demonstrating the importance of exploiting time dynamics.

Fig. 4 shows the impact of the number of observed vertices S in NMSE$(T, \{\mathcal{S}[\tau]\}_{\tau=1}^{T})$, with $T = 365$ days, averaged over all sets $\mathcal{S}[\tau] = \mathcal{S}\,\forall\tau$ with S elements. Observe that KKF consistently outperforms all alternatives. Still, the advantage of KKF is more pronounced for small S, since in that case exploiting the time dynamics is more critical.

To illustrate the trade-off between reliance on temporal versus spatial information, the next experiment analyzes the effects of the scaling parameter $b_{\mathcal{T}}$ in the kernel adopted by KRR (cf. (24)). A large value of $b_{\mathcal{T}}$ leads to an estimator that relies more heavily on time dynamics and vice versa. Fig. 5 shows NMSE$(T, \{\mathcal{S}[\tau]\}_{\tau=1}^{T})$, with $T = 100$ days, averaged over all sets $\mathcal{S}[\tau] = \mathcal{S}\,\forall\tau$ with $S = 44$ elements. The kernel in (24) is adopted with $\boldsymbol{K}_{\mathcal{V}}[t]$ being the regularized Laplacian (KKF-L) or diffusion kernels (KKF-DF) from Table I, while $\boldsymbol{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\boldsymbol{I}$. It is observed that there exists an optimum value for $b_{\mathcal{T}}$ which leads to the best reconstruction performance. This corresponds to the optimal trade-off point between reliance on temporal and spatial information. The optimal NMSE is achieved by a diffusion kernel with $\sigma = 1.5$ and $b_{\mathcal{T}} = 0.01$.

The second data set is provided by the Bureau of Economic Analysis of the U.S. Department of Commerce and contains the annual investments between each pair of sectors among $N = 61$ economic sectors in the interval 1997-2014 [37]. Each entry $A^{\mathcal{V}}_{n,n'}[t]$ of $\boldsymbol{A}_{\mathcal{V}}[t]$ contains the investment in trillions of dollars between sectors n and n' for the year

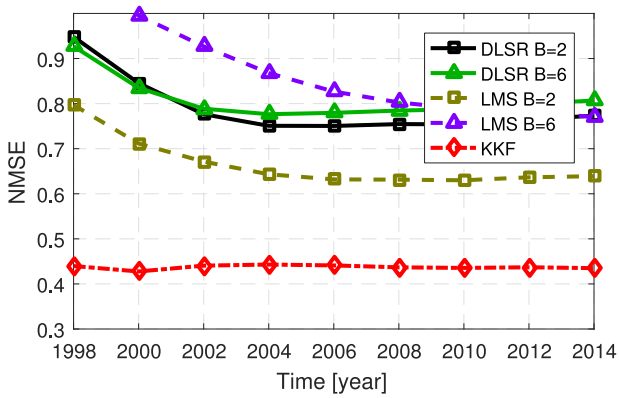

Fig. 6. NMSE for the economic sectors dataset ($\sigma = 6.2$, $\mu = 10^{-4}$, $b_{\mathcal{T}} = 0.01$, $\mu_{\text{DLSR}} = 1.2$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 2$).

$1995 + 2t$ with $t = 1, 2, \ldots, T$, where $T = 9$. DLSR and LMS adopt $\boldsymbol{A}_{\mathcal{V}} = (1/T)\sum_{\tau=1}^{T}\boldsymbol{A}_{\mathcal{V}}[\tau]$ since they cannot handle time-varying topologies. The value $f_n[t]$ corresponds to the total production of the n-th sector in year $1996 + 2t$, $t = 1, 2, \ldots, T$. The sampling interval was set to two years, so that disjoint subsets of years are used for generating the signal and constructing the graphs.

The next experiment demonstrates the ability of KKF to handle time-varying topologies. To this end, Fig. 6 plots NMSE$(t, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, averaged over all sets $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, with $S = 37$ elements. KKF utilizes the kernel in (24) with $\boldsymbol{K}_{\mathcal{V}}[t]$ a diffusion kernel constructed from $\boldsymbol{L}_{\mathcal{V}}[t]$ per t and $\boldsymbol{B}_{\mathcal{T}}[t] = b_{\mathcal{T}}\boldsymbol{I}$, $\forall t$. Again, Fig. 6 showcases the superior performance of the proposed KKF, whose error is significantly less than the error of competing alternatives.

The third data set is obtained from an epilepsy study [7]. Diagnosis of epilepsy is heavily based on analysis of ECoG data; see Section I. The next experiments utilize the ECoG time series obtained in [7] from $N = 76$ electrodes implanted in a patient's brain before and after the onset of a seizure. A symmetric time-invariant adjacency matrix $\boldsymbol{A}_{\mathcal{V}}$ was obtained using the method in [38] with ECoG data before the onset of the seizure. Function $f_n[t]$ comprises the electrical signal at the n-th electrode and t-th sampling instant after the onset of the seizure, for a period of $T = 250$ samples. The values of $f_n[t]$ were normalized by subtracting the temporal mean of each time series before the onset of the seizure. The goal of the experiment is to illustrate the reconstruction performance of the proposed KKF in capturing the complex spatio-temporal dynamics of brain signals.

Fig. 7 depicts the NMSE$(t, \{\mathcal{S}[\tau]\}_{\tau=1}^{t})$, averaged over all sets $\mathcal{S}[t] = \mathcal{S}$, $\forall t$, of size $S = 53$. For the proposed KKF, a space-time kernel was created using (24) with $\boldsymbol{K}_{\mathcal{V}}[t]$ a time-invariant covariance kernel $\boldsymbol{K}_{\mathcal{V}}[t] = \hat{\boldsymbol{\Sigma}}_{\text{tr}}$, where $\hat{\boldsymbol{\Sigma}}_{\text{tr}}$ was set to the sample covariance matrix of the time series before the onset of the seizure, and with a time-invariant $\boldsymbol{B}_{\mathcal{T}} = b_{\mathcal{T}}\boldsymbol{I}$. Fig. 7 showcases the superior reconstruction performance of the KKF, which successfully exploits the statistics of the signal when available, among competing approaches, even with a small number of samples. This result suggests that the ECoG diagnosis

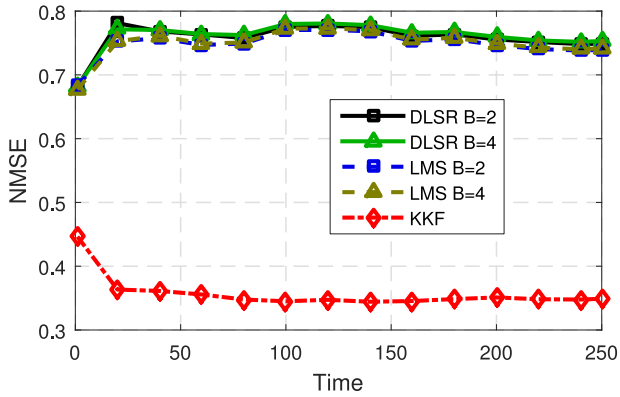

Fig. 7. NMSE for the ECoG dataset ($\sigma = 1.2$, $\mu = 10^{-4}$, $\mu_{\text{DLSR}} = 1.2$, $b_{\mathcal{T}} = 0.01$, $\beta_{\text{DLSR}} = 0.5$, $\mu_{\text{LMS}} = 0.6$).

technique could be efficiently conducted even with a smaller number of intracranial electrodes, which may positively impact the patient's experience.

VII. Conclusion

This paper investigated kernel-based reconstruction of space-time functions on graphs. The adopted approach relied on the construction of an extended graph, which regards the time dimension just as a spatial dimension. Several kernel designs were introduced together with a batch and an online function estimators. The latter is a kernel Kalman filter developed from a purely deterministic standpoint without any need to adopt any state-space model. Future research will deal with multi-kernel and distributed versions of the proposed algorithms.

Appendix

A. Proof of Lemma 1

Start by expanding the norm in the second term of (12) to obtain

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]||^2_{\boldsymbol{\Sigma}[\tau]} \\
&= \sum_{\tau=2}^{T} (\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1])^\top \boldsymbol{\Sigma}^{-1}[\tau](\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]) \\
&= \sum_{\tau=2}^{T} \Big(\boldsymbol{f}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{f}[\tau] - 2\boldsymbol{f}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1] \\
&\quad + \boldsymbol{f}^\top[\tau-1]\boldsymbol{P}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1] \Big). \qquad (25)
\end{aligned}$$

Noting that

$$\begin{aligned}
\sum_{\tau=2}^{T} \boldsymbol{f}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{f}[\tau] &= \sum_{\tau=2}^{T} \boldsymbol{f}^\top[\tau-1]\boldsymbol{\Sigma}^{-1}[\tau-1]\boldsymbol{f}[\tau-1] \\
&\quad - \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1] + \boldsymbol{f}^\top[T]\boldsymbol{\Sigma}^{-1}[T]\boldsymbol{f}[T]
\end{aligned}$$

it readily follows from (25) that

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]||^2_{\boldsymbol{\Sigma}[\tau]} \\
&= \sum_{\tau=2}^{T} \Big(\boldsymbol{f}^\top[\tau-1]\big(\boldsymbol{\Sigma}^{-1}[\tau-1] + \boldsymbol{P}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau]\big)\boldsymbol{f}[\tau-1] \\
&\quad - 2\boldsymbol{f}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1] \Big) \\
&\quad - \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1] + \boldsymbol{f}^\top[T]\boldsymbol{\Sigma}^{-1}[T]\boldsymbol{f}[T].
\end{aligned}$$

From Algorithm 1, it follows that $\boldsymbol{\Sigma}^{-1}[\tau-1] + \boldsymbol{P}^\top[\tau]\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau] = \boldsymbol{D}[\tau-1]$ and $-\boldsymbol{\Sigma}^{-1}[\tau]\boldsymbol{P}[\tau] = \boldsymbol{C}[\tau]$, which in turn imply that

$$\begin{aligned}
&\sum_{\tau=2}^{T} ||\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]||^2_{\boldsymbol{\Sigma}[\tau]} \\
&= \sum_{\tau=2}^{T} \Big(\boldsymbol{f}^\top[\tau-1]\boldsymbol{D}[\tau-1]\boldsymbol{f}[\tau-1] + 2\boldsymbol{f}^\top[\tau]\boldsymbol{C}[\tau]\boldsymbol{f}[\tau-1] \Big) \\
&\quad - \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1] + \boldsymbol{f}^\top[T]\boldsymbol{D}[T]\boldsymbol{f}[T] \\
&= \bar{\boldsymbol{f}}^\top \bar{\boldsymbol{K}}^{-1} \bar{\boldsymbol{f}} - \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1] \qquad (26)
\end{aligned}$$

where the last equality follows from (11). After substituting (26) into (12) and recognizing that the first summand in (12) equals $||\bar{\boldsymbol{y}}[t] - \bar{\boldsymbol{S}}[t]\bar{\boldsymbol{f}}||^2_{\boldsymbol{D}_S[t]}$, expression (9a) is recovered and the proof is completed.

B. Proof of Theorem 1

The first step is to simplify the objective in (12). To this end, note that minimizing (12) with respect to $\{\boldsymbol{f}[\tau]\}_{\tau=t'+1}^{T}$ for any t and t' with $t' \geq t$ yields

$$\hat{\boldsymbol{f}}[\tau|t] = \boldsymbol{P}[\tau]\hat{\boldsymbol{f}}[\tau-1|t],\ \tau = t'+1, \ldots, T, \qquad (27a)$$

$$\begin{aligned}
\{\hat{\boldsymbol{f}}[\tau|t]\}_{\tau=1}^{t'} = \underset{\{\boldsymbol{f}[\tau]\}_{\tau=1}^{t'}}{\arg\min} &\sum_{\tau=1}^{t} \frac{1}{\sigma_e^2[\tau]} ||\boldsymbol{y}[\tau] - \boldsymbol{S}[\tau]\boldsymbol{f}[\tau]||^2 \\
&+ \sum_{\tau=2}^{t'} ||\boldsymbol{f}[\tau] - \boldsymbol{P}[\tau]\boldsymbol{f}[\tau-1]||^2_{\boldsymbol{\Sigma}[\tau]} \\
&+ \boldsymbol{f}^\top[1]\boldsymbol{\Sigma}^{-1}[1]\boldsymbol{f}[1]. \qquad (27b)
\end{aligned}$$

The goal is therefore to show that the t-th iteration of Algorithm 2 returns $\hat{\boldsymbol{f}}[t|t]$ as given by (27b). To simplify notation, collect the function values up to time t as $\bar{\boldsymbol{f}}[t] := [\boldsymbol{f}^\top[1], \boldsymbol{f}^\top[2], \ldots, \boldsymbol{f}^\top[t]]^\top \in \mathbb{R}^{Nt}$ and their estimates given observations up to time t' as $\hat{\bar{\boldsymbol{f}}}[t|t'] := [\hat{\boldsymbol{f}}^\top[1|t'], \hat{\boldsymbol{f}}^\top[2|t'], \ldots, \hat{\boldsymbol{f}}^\top[t|t']]^\top \in \mathbb{R}^{Nt}$. The rest of the proof proceeds along the lines in [30, Ch. 17] by expressing $\hat{\bar{\boldsymbol{f}}}[t|t]$ and $\hat{\bar{\boldsymbol{f}}}[t|t-1]$ as the solutions to two least-squares problems.

To this end, define the $Nt+\bar{S}[t]\times Nt$ matrix

$$\bar{\boldsymbol{\mathcal{A}}}[t]:=\begin{bmatrix} \boldsymbol{I}_N & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{S}[1] & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{P}[2] & \boldsymbol{I}_N & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}[2] & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{S}[t-1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\boldsymbol{P}[t] & \boldsymbol{I}_N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \boldsymbol{S}[t] \end{bmatrix} \tag{28}$$

the $Nt+\bar{S}[t]\times Nt+\bar{S}[t]$ matrix $\bar{\boldsymbol{\Sigma}}[t]:=\text{bdiag}\{\boldsymbol{\Sigma}[1],\sigma_e^2[1]\boldsymbol{I}_{S[1]},\boldsymbol{\Sigma}[2],\sigma_e^2[2]\boldsymbol{I}_{S[2]},\dots,\sigma_e^2[t-1]\boldsymbol{I}_{S[t-1]},\boldsymbol{\Sigma}[t],\sigma_e^2[t]\boldsymbol{I}_{S[t]}\}$, and note from (27b) that

$$\hat{\bar{\boldsymbol{f}}}[t|t]=\arg\min_{\bar{\boldsymbol{f}}[t]}||\bar{\boldsymbol{\psi}}[t]-\bar{\boldsymbol{\mathcal{A}}}[t]\bar{\boldsymbol{f}}[t]||^2_{\bar{\boldsymbol{\Sigma}}[t]} \tag{29}$$

where $\bar{\boldsymbol{\psi}}[t]:=[\mathbf{0}_N^\top,\boldsymbol{y}^\top[1],\mathbf{0}_N^\top,\boldsymbol{y}^\top[2],\mathbf{0}_N^\top,\dots,\mathbf{0}_N^\top,\boldsymbol{y}^\top[t]]^\top\in\mathbb{R}^{Nt+\bar{S}[t]}$. Indeed, expression (29) corresponds to the weighted least-squares solution to

$$\bar{\boldsymbol{\psi}}[t]=\bar{\boldsymbol{\mathcal{A}}}[t]\bar{\boldsymbol{f}}[t]+\bar{\boldsymbol{\epsilon}}[t] \tag{30}$$

where $\bar{\boldsymbol{\epsilon}}[t]\in\mathbb{R}^{Nt+\bar{S}[t]}$ is an error vector, and admits the closed-form solution

$$\hat{\bar{\boldsymbol{f}}}[t|t]=(\bar{\boldsymbol{\mathcal{A}}}^\top[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}[t])^{-1}\bar{\boldsymbol{\mathcal{A}}}^\top[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t]. \tag{31}$$

Similarly, define the $Nt+\bar{S}[t-1]\times Nt$ matrix

$$\bar{\boldsymbol{\mathcal{A}}}'[t]:=\begin{bmatrix} \boldsymbol{I}_N & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{S}[1] & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{P}[2] & \boldsymbol{I}_N & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}[2] & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & -\boldsymbol{P}[t-1] & \boldsymbol{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{S}[t-1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\boldsymbol{P}[t] & \boldsymbol{I}_N \end{bmatrix} \tag{32}$$

which is a submatrix of $\bar{\boldsymbol{\mathcal{A}}}[t]$ that results from removing the last block-row, together with the $Nt+\bar{S}[t-1]\times Nt+\bar{S}[t-1]$ matrix $\bar{\boldsymbol{\Sigma}}'[t]:=\text{bdiag}\{\boldsymbol{\Sigma}[1],\sigma_e^2[1]\boldsymbol{I}_{S[1]},\boldsymbol{\Sigma}[2],\sigma_e^2[2]\boldsymbol{I}_{S[2]},\dots,\boldsymbol{\Sigma}[t-1],\sigma_e^2[t-1]\boldsymbol{I}_{S[t-1]},\boldsymbol{\Sigma}[t]\}$, which is a submatrix of $\bar{\boldsymbol{\Sigma}}[t]$ resulting from removing the last block-row and block-column. Now, replace t with $t-1$ and t' with t in (27b) to obtain

$$\hat{\bar{\boldsymbol{f}}}[t|t-1]=\arg\min_{\bar{\boldsymbol{f}}[t]}||\bar{\boldsymbol{\psi}}'[t]-\bar{\boldsymbol{\mathcal{A}}}'[t]\bar{\boldsymbol{f}}[t]||^2_{\bar{\boldsymbol{\Sigma}}'[t]} \tag{33}$$

where $\bar{\boldsymbol{\psi}}'[t]:=[\mathbf{0}_N^\top,\boldsymbol{y}^\top[1],\mathbf{0}_N^\top,\boldsymbol{y}^\top[2],\mathbf{0}_N^\top,\dots,\boldsymbol{y}^\top[t-1],\mathbf{0}_N^\top]^\top\in\mathbb{R}^{Nt+\bar{S}[t-1]}$ is a submatrix of $\bar{\boldsymbol{\psi}}[t]$ that results from removing its last block-row. In this case, $\hat{\bar{\boldsymbol{f}}}[t|t-1]$ in (33) corresponds to the least-squares solution to (30) after removing the last $S[t]$ equations, and can be obtained in closed form as

$$\hat{\bar{\boldsymbol{f}}}[t|t-1]=(\bar{\boldsymbol{\mathcal{A}}}'^\top[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}'[t])^{-1}\bar{\boldsymbol{\mathcal{A}}}'^\top[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t]. \tag{34}$$

The rest of the proof utilizes (31) and (34) to express $\hat{\boldsymbol{f}}[t|t]$ in terms of $\hat{\boldsymbol{f}}[t|t-1]$ and $\hat{\boldsymbol{f}}[t|t-1]$ in terms of $\hat{\boldsymbol{f}}[t-1|t-1]$. To this end, define $\bar{\boldsymbol{\mathcal{J}}}[t]:=\boldsymbol{i}_{t,t}^\top\otimes\boldsymbol{I}_N$, which can be used to select the last $N\times N$ block-row or block-column of a matrix, as well as

$$\boldsymbol{M}[t|t-1]:=\bar{\boldsymbol{\mathcal{J}}}[t](\bar{\boldsymbol{\mathcal{A}}}'^\top[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}'[t])^{-1}\bar{\boldsymbol{\mathcal{J}}}^\top[t] \tag{35}$$

and

$$\boldsymbol{M}[t|t]:=\bar{\boldsymbol{\mathcal{J}}}[t](\bar{\boldsymbol{\mathcal{A}}}^\top[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}[t])^{-1}\bar{\boldsymbol{\mathcal{J}}}^\top[t], \tag{36}$$

which respectively correspond to the bottom right $N\times N$ blocks of $\bar{\boldsymbol{T}}'[t]:=\bar{\boldsymbol{\mathcal{A}}}'^\top[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}'[t]$ and $\bar{\boldsymbol{T}}[t]:=\bar{\boldsymbol{\mathcal{A}}}^\top[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}[t]$. Expressions (36) and (35) will be used next to express $\boldsymbol{M}[t|t-1]$ in terms of $\boldsymbol{M}[t-1|t-1]$, and $\boldsymbol{M}[t|t]$ in terms of $\boldsymbol{M}[t|t-1]$.

Assume for simplicity that $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$ equal the identity matrices of appropriate sizes, although the proof easily carries over to arbitrary positive definite matrices $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$. Note that

$$\bar{\boldsymbol{T}}'[t]=\begin{bmatrix}\bar{\boldsymbol{T}}[t-1]+\boldsymbol{V}^\top[t]\boldsymbol{V}[t] & \boldsymbol{V}^\top[t] \\ \boldsymbol{V}[t] & \boldsymbol{I}_N\end{bmatrix} \tag{37}$$

where $\boldsymbol{V}[t]:=-\boldsymbol{P}[t]\bar{\boldsymbol{\mathcal{J}}}[t-1]\in\mathbb{R}^{N\times N(t-1)}$ and observe that $\boldsymbol{M}[t|t-1]$ is the bottom right $N\times N$ block of $\bar{\boldsymbol{T}}'^{-1}[t]$. Thus, applying block matrix inversion to (37) yields

$$\begin{aligned}\boldsymbol{M}[t|t-1]&=\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\bar{\boldsymbol{\mathcal{J}}}^\top[t]\\&=\big(\boldsymbol{I}_N-\boldsymbol{V}[t]\big(\bar{\boldsymbol{T}}[t-1]+\boldsymbol{V}^\top[t]\boldsymbol{V}[t]\big)^{-1}\boldsymbol{V}^\top[t]\big)^{-1}.\end{aligned} \tag{38}$$

Moreover, the matrix inversion lemma yields,

$$\begin{aligned}&\big(\bar{\boldsymbol{T}}[t-1]+\boldsymbol{V}^\top[t]\boldsymbol{I}_N\boldsymbol{V}[t]\big)^{-1}=\bar{\boldsymbol{T}}^{-1}[t-1]-\bar{\boldsymbol{T}}^{-1}[t-1]\times\\&\boldsymbol{V}^\top[t]\big(\boldsymbol{I}_N+\boldsymbol{V}[t]\bar{\boldsymbol{T}}^{-1}[t-1]\boldsymbol{V}^\top[t]\big)^{-1}\boldsymbol{V}[t]\bar{\boldsymbol{T}}^{-1}[t-1].\end{aligned} \tag{39}$$

Substituting (39) into (38), applying the definition of $\boldsymbol{V}[t]$, and using (36) to identify $\boldsymbol{M}[t-1|t-1]$ enables one to express $\boldsymbol{M}[t|t-1]$ in terms of $\boldsymbol{M}[t-1|t-1]$ as

$$\boldsymbol{M}[t|t-1]=\boldsymbol{I}_N+\boldsymbol{P}[t]\boldsymbol{M}[t-1|t-1]\boldsymbol{P}^\top[t]. \tag{40}$$

On the other hand, to express $\boldsymbol{M}[t|t]$ in terms of $\boldsymbol{M}[t|t-1]$, note that $\bar{\boldsymbol{\mathcal{A}}}[t]=[\bar{\boldsymbol{\mathcal{A}}}'^\top[t],\boldsymbol{W}^\top[t]]^\top$, where $\boldsymbol{W}[t]:=\boldsymbol{S}[t]\bar{\boldsymbol{\mathcal{J}}}[t]\in\mathbb{R}^{S[t]\times Nt}$. Therefore,

$$\begin{aligned}\bar{\boldsymbol{T}}[t]&=\bar{\boldsymbol{\mathcal{A}}}^\top[t]\bar{\boldsymbol{\mathcal{A}}}[t]\\&=\bar{\boldsymbol{\mathcal{A}}}'^\top[t]\bar{\boldsymbol{\mathcal{A}}}'[t]+\boldsymbol{W}^\top[t]\boldsymbol{W}[t]\\&=\bar{\boldsymbol{T}}'[t]+\boldsymbol{W}^\top[t]\boldsymbol{W}[t].\end{aligned} \tag{41}$$

Applying the matrix inversion lemma to (41) yields

$$\bar{\boldsymbol{T}}^{-1}[t] = \bar{\boldsymbol{T}}'^{-1}[t] - \bar{\boldsymbol{T}}'^{-1}[t]\boldsymbol{W}^{\top}[t]\times \big(\boldsymbol{I}_{S[t]} + \boldsymbol{W}[t]\bar{\boldsymbol{T}}'^{-1}[t]\boldsymbol{W}^{\top}[t]\big)^{-1}\boldsymbol{W}[t]\bar{\boldsymbol{T}}'^{-1}[t]. \tag{42}$$

Substituting the definition of $\boldsymbol{W}[t]$ into (42) leads to

$$\begin{aligned}\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}^{-1}[t] &= \bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t] - \bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\bar{\boldsymbol{\mathcal{J}}}^{\top}[t]\boldsymbol{S}^{\top}[t]\times\\ &\big(\boldsymbol{I}_{S[t]} + \boldsymbol{S}[t]\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\bar{\boldsymbol{\mathcal{J}}}^{\top}[t]\boldsymbol{S}^{\top}[t]\big)^{-1}\boldsymbol{S}[t]\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\\ &= \bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t] - \boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t]\times\\ &\big(\boldsymbol{I}_{S[t]} + \boldsymbol{S}[t]\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t]\big)^{-1}\boldsymbol{S}[t]\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\\ &= (\boldsymbol{I}_N - \boldsymbol{G}[t]\boldsymbol{S}[t])\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\end{aligned} \tag{43}$$

where the second equality follows from (35), and the third from

$$\boldsymbol{G}[t] := \boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t](\boldsymbol{I}_{S[t]} + \boldsymbol{S}[t]\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t])^{-1}. \tag{44}$$

Finally, multiplying both sides of (43) with $\bar{\boldsymbol{\mathcal{J}}}^{\top}[t]$ and using (36) to identify $\boldsymbol{M}[t|t]$ enables one to express $\boldsymbol{M}[t|t]$ in terms of $\boldsymbol{M}[t|t-1]$ as

$$\boldsymbol{M}[t|t] = (\boldsymbol{I}_N - \boldsymbol{G}[t]\boldsymbol{S}[t])\boldsymbol{M}[t|t-1]. \tag{45}$$

If $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\Sigma}}'[t]$ are not identity matrices, then one obtains

$$\boldsymbol{M}[t|t-1] = \boldsymbol{\Sigma}[t] + \boldsymbol{P}[t]\boldsymbol{M}[t-1|t-1]\boldsymbol{P}^{\top}[t] \tag{46}$$

instead of (40), and

$$\boldsymbol{G}[t] = \boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t](\sigma_e^2[t]\boldsymbol{I}_{S[t]} + \boldsymbol{S}[t]\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t])^{-1} \tag{47}$$

instead of (44), whereas (45) remains the same. These equations are precisely those in steps 4, 5 and 7 of Algorithm 2.

To obtain the rest of the steps, set t to $t-1$ and τ to t in (27a) to obtain

$$\hat{\boldsymbol{f}}[t|t-1] = \boldsymbol{P}[t]\hat{\boldsymbol{f}}[t-1|t-1] \tag{48}$$

which coincides with step 3 of Algorithm 2. Finally, since $\hat{\boldsymbol{f}}[t|t]$ is the last block vector of $\hat{\bar{\boldsymbol{f}}}[t|t]$, then

$$\begin{aligned}\hat{\boldsymbol{f}}[t|t] &:= \bar{\boldsymbol{\mathcal{J}}}[t]\hat{\bar{\boldsymbol{f}}}[t|t]\\ &= \bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}^{\top}[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t]\\ &= (\boldsymbol{I} - \boldsymbol{G}[t]\boldsymbol{S}[t])\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\bar{\boldsymbol{\mathcal{A}}}^{\top}[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t]\end{aligned} \tag{49}$$

where the second equality follows from (31) and the third from (43). From the definitions of $\bar{\boldsymbol{\mathcal{A}}}[t]$, $\bar{\boldsymbol{\Sigma}}[t]$ and $\bar{\boldsymbol{\psi}}[t]$, one obtains that

$$\bar{\boldsymbol{\mathcal{A}}}^{\top}[t]\bar{\boldsymbol{\Sigma}}^{-1}[t]\bar{\boldsymbol{\psi}}[t] = \bar{\boldsymbol{\mathcal{A}}}'^{\top}[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t] + \frac{1}{\sigma_e^2[t]}\boldsymbol{W}^{\top}[t]\boldsymbol{y}[t]. \tag{50}$$

Substituting (50) into (49) yields

$$\begin{aligned}\hat{\boldsymbol{f}}[t|t] &= (\boldsymbol{I} - \boldsymbol{G}[t]\boldsymbol{S}[t])\bar{\boldsymbol{\mathcal{J}}}[t]\bar{\boldsymbol{T}}'^{-1}[t]\times\\ &\quad(\bar{\boldsymbol{\mathcal{A}}}'^{\top}[t]\bar{\boldsymbol{\Sigma}}'^{-1}[t]\bar{\boldsymbol{\psi}}'[t] + \frac{1}{\sigma_e^2[t]}\boldsymbol{W}^{\top}[t]\boldsymbol{y}[t])\\ &= (\boldsymbol{I} - \boldsymbol{G}[t]\boldsymbol{S}[t])(\hat{\boldsymbol{f}}[t|t-1]\\ &\quad + \frac{1}{\sigma_e^2[t]}\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t]\boldsymbol{y}[t])\\ &= \hat{\boldsymbol{f}}[t|t-1] + \boldsymbol{G}[t](\boldsymbol{y}[t] - \boldsymbol{S}[t]\hat{\boldsymbol{f}}[t|t-1])\end{aligned} \tag{51}$$

where the second equality follows from (34), $\hat{\boldsymbol{f}}[t|t-1] = \bar{\boldsymbol{\mathcal{J}}}[t]\hat{\bar{\boldsymbol{f}}}[t|t-1]$ and (35); whereas the third follows from

$$(\boldsymbol{I}_N - \boldsymbol{G}[t]\boldsymbol{S}[t])\boldsymbol{M}[t|t-1]\boldsymbol{S}^{\top}[t] = \sigma_e^2[t]\boldsymbol{G}[t] \tag{52}$$

which results from rearranging the terms in (47). Noting that expression (51) coincides with step 6 of Algorithm 2 concludes the proof.

REFERENCES

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY, USA: Springer, 2009.

[2] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Jul. 2002, pp. 315–322.

[3] A. J. Smola and R. I. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*. New York, NY, USA: Springer, 2003, pp. 144–158.

[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, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[6] O. Chapelle *et al.*, *Semi-supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[7] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch, "Emergent network topology at seizure onset in humans," *Epilepsy Res.*, vol. 79, no. 2, pp. 173–186, 2008.

[8] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[9] O. Chapelle, V. Vapnik, and J. Weston, "Transductive inference for estimating values of functions," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1999, vol. 12, pp. 421–427.

[10] C. Cortes and M. Mohri, "On transductive regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2007, pp. 305–312.

[11] J. Lafferty and L. Wasserman, "Statistical analysis of semi-supervised regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2007, pp. 801–808.

[12] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[13] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

[14] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, 2013, pp. 491–494.

[15] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[16] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.

[17] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.

[18] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2189–2199, Aug. 2004.
[19] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.
[20] P. A. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of partially observed dynamical processes over networks via dictionary learning," *IEEE Trans. Signal Process.*, vol. 62, no. 13, pp. 3305–3320, Jul. 2014.
[21] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 728–740, Jun. 2015.
[22] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 555–568, Dec. 2016.
[23] K. Rajawat, E. Dall'Anese, and G. B. Giannakis, "Dynamic network delay cartography," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2910–2920, May 2014.
[24] V. Kekatos, Y. Zhang, and G. B. Giannakis, "Electricity market forecasting via low-rank multi-kernel learning," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 6, pp. 1182–1193, Dec. 2014.
[25] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.
[26] L. Ralaivola and F. D'Alché-Buc, "Time series filtering, smoothing and learning using the kernel Kalman filter," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, vol. 3, pp. 1449–1454.
[27] P. Zhu, B. Chen, and J. C. Príncipe, "Learning nonlinear generative models of time series with a Kalman filter in RKHS," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 141–155, Jan. 2014.
[28] K. Wehmuth, A. Ziviani, and E. Fleury, "A unifying model for representing time-varying graphs," in *Proc. Int. Conf. Data Sci. Adv. Analytics*, Paris, France, Oct. 2015, pp. 1–10.
[29] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Proc. ICML Workshop Statist. Relational Learn. Connections Other Fields*, Banff, AB, Canada, Jul. 2004, vol. 15, pp. 67–68.
[30] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*. Philadelphia, PA, USA: SIAM, 1997.
[31] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *Amer. Inst. Aeronaut. Astronaut. J.*, vol. 3, no. 8, pp. 1445–1450, 1965.
[32] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *Proc. 24th Eur. Signal Process. Conf.*, Budapest, Hungary, Aug. 2016, pp. 200–204.
[33] P. M. Weichsel, "The Kronecker product of graphs," *Proc. Amer. Math. Soc.*, vol. 13, no. 1, pp. 47–52, 1962.
[34] H. Kashima, S. Oyama, Y. Yamanishi, and K. Tsuda, "On pairwise kernels: An efficient alternative and generalization analysis," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 1030–1037.
[35] 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.
[36] A. Arguez, I. Durre, S. Applequist, M. Squires, R. Vose, X. Yin, and R. Bilotta, "NOAA's U.S. climate normals," (1981–2010). [Online]. Available: https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/climate-normals/1981-2010-normals-data, Accessed on: Sep. 2016.
[37] U. S. Bureau of Economic Analysis, "Table input-output of industry data," [Online]. Available: http://www.bea.gov/iTable/index_industry_io.cfm, Accessed on: Sep. 2016.
[38] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity," arXiv preprint arXiv:1610.06551, 2016.

Daniel Romero (M'16) received the M.Sc. and Ph.D. degrees in signal theory and communications from the University of Vigo, Vigo, Spain, in 2011 and 2015, respectively. From July 2015 to November 2016, he was a Postdoctoral Researcher with the Digital Technology Center and the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. In December 2016, he joined the Department of Information and Communication Technology, University of Agder, Grimstad, Norway, as an Associate Professor. His main research interests include signal processing, communications, and machine learning.

Vassilis N. Ioannidis received the Diploma degere in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2015. He is working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, MN, USA. From 2014 to 2015, he worked as a middleware consultant for Oracle in Athens, Greece, and received a Performance Excellence Award. His research interests include statistical signal processing, big data analytics, and machine learning over networks.

Georgios B. Giannakis (F'97) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1981, and the M.Sc. degree in electrical engineering, the M.Sc. degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1983, 1986, and 1986, respectively.

He was with the University of Virginia from 1987 to 1998. Since 1999, he has been a Professor with the University of Minnesota, Minneapolis, MN, USA, where he holds an Endowed Chair in Wireless Telecommunications, a University of Minnesota McKnight Presidential Chair in electronics and communication engineering, and serves as the Director of the Digital Technology Center. His general research interests span the areas of communications, networking and statistical signal processing—subjects on which he has authored or coauthored more than 400 journal papers, 700 conference papers, 25 book chapters, two edited books, and two research monographs (H-index 127). He is the (co-) inventor of 30 patents issued. His current research focuses on learning from Big Data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He has received eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the IEEE SP Society in 2000, from EURASIP in 2005, a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award in 2015. He is a Fellow of EURASIP and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE SP Society.