State-Space Network Topology Identification from Partial Observations

Coutino, Mario; Isufi, Elvin; Maehara, Takanori; Leus, Geert

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

State-Space Network Topology Identification From Partial Observations

Mario Coutino ⬤ , *Student Member, IEEE*, Elvin Isufi ⬤ *, Member, IEEE*, Takanori Maehara,
and Geert Leus ⬤ *, Fellow, IEEE*

Abstract—In this article, we explore the state-space formulation of a network process to recover from partial observations the network topology that drives its dynamics. To do so, we employ subspace techniques borrowed from system identification literature and extend them to the network topology identification problem. This approach provides a unified view of network control and signal processing on graphs. In addition, we provide theoretical guarantees for the recovery of the topological structure of a deterministic continuous-time linear dynamical system from input-output observations even when the input and state interaction networks are different. Our mathematical analysis is accompanied by an algorithm for identifying from data, a network topology consistent with the system dynamics and conforms to the prior information about the underlying structure. The proposed algorithm relies on alternating projections and is provably convergent. Numerical results corroborate the theoretical findings and the applicability of the proposed algorithm.

Index Terms—Inverse eigenvalue problems, graph signal processing, signal processing over networks, state-space models, network topology identification.

I. INTRODUCTION

THE TOPOLOGY of networks is fundamental to model interactions between entities and to improve our understanding about the processes evolving over them. We find examples of such processes in transportation networks [1], brain activity [2], and epidemic dynamics or gene regulatory networks [3], to name a few. The coupling between the process and the network topology extended signal processing (SP) techniques to tools that take into account the network structure for defining

signal estimators [4]–[6], filters [7]–[9], and optimal detectors [10]–[12].

While in several scenarios, the network structure is available, in many others, it is unknown and needs to be estimated. This is not only for enhancing data processing tasks but also for data *interpretability*, i.e., the network topology provides an abstraction for the underlying data dependencies. Therefore, retrieving the network structure or the dependencies of the involved members (variables) has become a research topic of large interest [13]–[21].

Despite many works focus on the problem of topology identification [22], [23] or Gaussian graphical modeling [24], [25], most of these approaches only leverage a model based on the so-called *graph filters* [26], graph signal smoothness [27], [28], or enforce a particular structure by dictonary learning [29] or by a penalized likelihood approach with sparsity constraints [30], [31]. Among the works that consider an alternative interaction model e.g., [17], [21], [32], [33], only a few of them consider the network data as states of an underlying process. However, none of the above works study the case where the input, i.e., excitation or probing signal of the process, and the process itself evolve according to different topologies.

In many instances, physical systems can be defined through a state-space formulation with known dynamics. An example is the diffusion of molecules in tissue [34]. This process is used to analyze brain functions by mapping the interaction of the molecules with obstacles. The area of neural dynamics considers also the problem of network design in transport theory [35]. In such applications, the topology that provides a stable desired response needs to be found, following a differential equation. Finally, we recall the problem of finding the connections between reactants in chemical reaction networks [36], [37]. Here, molecules evolve in a solution according to the interaction between reactants present in it. Hence, to understand the underlying chemical process, the relation between reactants is required. Considering these systems, it is clear that a more general approach, parting from first-principles, to find the underlying connections is preferred. So, we focus on the problem of *retrieving the network structure* of a process modeled through a *deterministic continuous-time dynamical system* whose system matrices depend on the underlying topology.

Within the system identification literature [38]–[40], several methods have been devised to address this problem instance. In particular, methods for inferring the network structure using response dynamics [41], [42] and linearized network

dynamics [43] resulted successful in recovering the network topology. However, these approaches require exact knowledge of the dynamics, i.e., node driving model, and measurements at each node.

To relax the assumption of exact nodal dynamic model knowledge, model-free methods [38] have been proposed to infer network interactions. Most of these methods, e.g., [37], [44]–[46], assume the coupling (interaction) between two network elements belongs to a set of candidate functions or that such dynamics can be represented with a few candidate basis functions. Under this assumption, compressive sensing and sparse recovery techniques are leveraged to retrieve the network structure. Although these methods achieve good performance when the basis functions are guided by physical principles, they need access to all system states and consider the system states' rate of change is either measured directly or estimated accurately. Hence, when access to these quantities is not available, i.e., there are hidden nodes, these methods are not directly applicable as pointed out in [37], [43].

To address the problem of hidden nodes in a network, e.g., partial observation of the true system states, several works have advocated indirect methods to locate their position with respect to the visible network structure [47], [48]. For instance, [47] showed that when a sparse recovery problem has to be solved to unveil the network structure, anomalously dense regressor vectors (non-sparse vectors) are correlated with the existence of a hidden node or with a node-dependent noise source. This observation lead to a series of attempts to differentiate the effect of a hidden node and a noise source in the network data. Most of these approaches rely on pair-wise comparisons using the so-called cancellation ratio [48]. Although this kind of approaches can be extended to unveil multiple entangled hidden nodes, the cancellation-coefficient based technique needs to be performed on candidate groups of nodes that are believed to be connected to hidden nodes. In addition, despite these methods perform well when the basis functions are the true ones, they are sensitive to basis expansion mismatch. Hence, setting an anomalous density threshold is not straightforward.

With the resurgence of statistical system identification in machine learning [49], [50], methods using Hankel-structured matrices, inspired by the celebrated Kalman-Ho algorithm [51], estimate system realizations from a single time series; see, e.g., [52], [53]. Many of these methods are the least-squares counterparts of the subspace-based system identification methods [54] which estimate a system realization from a Hankel embedding. Although these new approaches provide insights into lowest-order system approximation, they do not leverage any prior knowledge with respect to the system matrices and focus on approximate rather than exact system realizations.

In this work, we focus on retrieving a network topology, from partial observations, that not only captures the interactions of the network elements but also the dynamics of the underlying network process. To address this task, we devise a framework, using well-established tools from system identification [38], to estimate the network topology from partial observations up to ambiguities defined through the equivalence class of restricted cospectral graphs [55]. These ambiguity results extend some of

the observations in [56], where fundamental limits for network reconstruction from temporal data were derived from a group-theoretic perspective.

Finally, we remark that another related area to topology identification is metric learning [57]. These works recover a *metric* that preserves distances between similar objects. This metric is represented by a matrix which can be linked with a graph [58]. Although [58] considers recovering a network matrix from partial observations, their assumptions make their approach unapplicable to our setting since: (i) they assume graph smoothness [cf. [27]]; and (ii) they consider available feature vectors per node. In other words, they construct a similarity graph without considering the dynamics nor the driving process.

A. Overview and Main Contributions

Network topology inference from partial measurements is generally an ill-posed problem and it is still far from being completely understood. Therefore, in this work, we present an approach based on state-space models to leverage first-principles for model-driven topology estimation. Our contributions broadening the state-of-the-art are the following.

– Using a first-order differential graph model for describing the dynamics of a deterministic continuous-time linear network process, we put forth a first-principles based network topology identification framework leveraging subspace-based system identification techniques. We also provide conditions to retrieve the network topology from sampled observations.

– We analyze mathematically the problem of network topology identification from partial observations and show it is ill-posed. Similarly to [56], we describe the ambiguities present when recovering the network topology from measurements that do not uniquely identify the underlying structure using cospectral graphs.

– We present an algorithm based on the alternating projections (AP) approach [59] which is provably globally convergent to estimate the network structure under the partial observation setting. We further prove that under mild conditions the proposed AP method converges locally with a linear rate to a feasible solution.

– Finally, we extend the topology inference problem from partial observations to instances where incomplete or inaccurate process dynamics are present. For these cases, we provide a mathematical analysis and conditions that guarantee the convergence of the proposed AP method when estimating a feasible network topology.

B. Outline and Notation

This paper is organized as follows. Section II formulates the problem of network topology inference for continuous-time dynamical systems from sampled observations. Section III introduces a first-order differential graph model and its state-space description. The system identification framework for the proposed graph-based model is introduced in Section IV. Section V analyzes the ambiguity of network topology identification from

partial observations and provides an AP method to find a feasible network structure. Section VI discusses system consistency constraints that can be enforced into the AP method to match the dynamics. Section VII corroborates the theory with numerical results. Finally, section VIII concludes the paper.

We adopt the following notation. Scalars, vectors, matrices and sets are denoted by lowercase letters (x), lowercase boldface letters $(\boldsymbol{x})$, uppercase boldface letters $(\boldsymbol{X})$, and calligraphic letters $(\mathcal{X})$, respectively. $[\boldsymbol{X}]_{i,j}$ denotes the (i,j)th entry of the matrix $\boldsymbol{X}$, whereas $[\boldsymbol{x}]_i$ represents the ith entry of the vector $\boldsymbol{x}$. $\boldsymbol{X}^T$ and $\boldsymbol{X}^{-1}$ are the transpose and the inverse of $\boldsymbol{X}$, respectively. The Moore-Penrose pseudoinverse of $\boldsymbol{X}$ is denoted by $\boldsymbol{X}^\dagger$. $\mathrm{vec}(\cdot)$ is the vectorization operation. $\mathrm{bdiag}(\boldsymbol{X},\boldsymbol{Y})$ denotes a block diagonal matrix whose blocks are given by the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{I}$ is the identity matrix of appropriate size. $\|\boldsymbol{X}\|_\mathrm{F}$ and $\|\boldsymbol{X}\|_2$ denote the Frobenius- and ℓ_2-norm of $\boldsymbol{X}$, respectively. $\mathrm{span}(\cdot)$ and $\mathrm{rank}(\cdot)$ are the span and rank of a matrix, respectively. Finally, we use $[K]$ to denote the set $\{1,2,\ldots,K\}$ and $\mathcal{D}_K$ to denote the set of $K \times K$ diagonal matrices.

II. PROBLEM STATEMENT

Consider a set of N nodes $\mathcal{V} = \{v_1,\ldots,v_N\}$ representing cooperative agents such as sensors, individuals, and biological structures. On top of these agents, a process $\mathcal{P}$ describes the evolution through time of the agent signals $\boldsymbol{x}(t) \in \mathbb{R}^N$. Signal $\boldsymbol{x}(t)$ is such that the ith component $x_i(t)$ represents the signal evolution of agent v_i. The agent interactions w.r.t. the evolution of signal $\boldsymbol{x}(t)$ are captured by a graph $G_x = (\mathcal{V}, \mathcal{E}_x)$, where $\mathcal{E}_x$ is the edge set of this graph. We consider process $\mathcal{P}$ is represented by a first-order differential model

$$\partial_t \boldsymbol{x}(t) = h(\mathcal{V}, \mathcal{E}_x, \mathcal{E}_u, \boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{1a}$$

$$\boldsymbol{y}(t) = c(\mathcal{V}, \mathcal{E}_x, \mathcal{E}_u, \boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{1b}$$

where $\partial_t \boldsymbol{x}(t) := d\boldsymbol{x}(t)/dt$ and $h(\cdot)$ and $c(\cdot)$ are maps that describe respectively the dynamics of signal $\boldsymbol{x}(t)$ and observables $\boldsymbol{y}(t)$. In model (1), $\boldsymbol{u}(t)$ is the (known) system input and $\mathcal{E}_u$ is the edge set of another graph $G_u = (\mathcal{V}, \mathcal{E}_u)$ that captures the interactions between the elements of $\mathcal{V}$ for input $\boldsymbol{u}(t)$. Put simply, process (1) describes the evolution of signal $\boldsymbol{x}(t)$ under the influence of maps $h(\cdot)$ and $c(\cdot)$ and the network topologies $G_x = (\mathcal{V}, \mathcal{E}_x)$ and $G_u = (\mathcal{V}, \mathcal{E}_u)$. For future reference, we will represent both graphs as $G_* = \{\mathcal{V}, \mathcal{E}_*\}$, where "$*$" is a space holder for x and u. We remark that although (1) leads to a directed process, i.e., there is a directed field flow, the topology (or metric of the space) does not necessarily have to be directed, e.g., heat diffusion. That is, the matrix representation of G_* can indeed be a symmetric matrix (undirected graph).

We can compactly define process $\mathcal{P}$ through the set

$$\mathcal{P} := \{h(\cdot), c(\cdot)\}, \tag{2}$$

which contains the interactions in the system. While process $\mathcal{P}$ describes a continuous-time process, we usually have access to *sampled realizations of it*, i.e., the observables $\boldsymbol{y}(t)$ are collected on a finite set of time instances $\mathcal{T} := \{t_1, t_2, \ldots, t_T\}$. As a

result, we might also want to consider *discrete-time* approximations of (2), where we either have a discrete-time realization of $\mathcal{P}$ and/or a finite number of observables $\mathcal{Y} := \{\boldsymbol{y}(t)\}_{t \in \mathcal{T}}$. Processes belonging to this family of models include, for example, linearized dynamics of biochemical reactions, where the state variables, $\boldsymbol{x}(t)$, represent the concentration of mRNA and proteins in genes; the inputs, $\boldsymbol{u}(t)$, are external stimuli applied to the system such as heat or current, and the underlying network denotes the interactions (dependencies) between the genes. In addition to this, diffusion processes in networks modeled using the heat diffusion equation, also are included in this family of models. Here, the state variables represent nodal quantities that get diffused through the network by local aggregations. For further examples where these models are applicable, we refer the reader to the applications in [60].

With this in place, we ask: *how to retrieve the network topologies $\mathcal{G}_x$ and $\mathcal{G}_u$ for the agent signal $\boldsymbol{x}(t)$ and input signal $\boldsymbol{u}(t)$ given the process $\mathcal{P}$ [cf. (2)], the input signal $\boldsymbol{u}(t)$, and the observables in $\mathcal{Y}$?*

We answer the above question by employing results from Hankel matrices [61] and linear algebra whose foundations lie in system identification theory [62]. We consider subspace techniques that are by definition *cost function free*. This differs from the commonly used techniques in network topology identification [22] where the learned topology heavily depends on the cost function (e.g., smoothness or sparsity). If this prior knowledge is incorrect, it leads to structures not related to the physical interactions. The adopted techniques provide theoretical insights when and how the underlying network structures can be identified. They also have two other benefits: first, they impose no parameterization on the dynamical model and, therefore, avoid solving nonlinear optimization problems as in prediction-error methods [63]; second, they allow identifying $G_* = \{\mathcal{V}, \mathcal{E}_*\}$ from sampled data, i.e., by having access only to a subset of the observables $\boldsymbol{y}(t)$.

Finally, we remark that although more general models can be considered, e.g., higher-order differential models, we restrict ourselves to first-order models to ease exposition and provide a thorough and stand-alone work. Nevertheless, first-order differential models are of broad interest as they include diffusion processes –see [64] and references therein.

III. FIRST ORDER DIFFERENTIAL GRAPH MODEL

The graph $G_* = \{\mathcal{V}, \mathcal{E}_*\}$ is mathematically represented by a matrix $\boldsymbol{S}_*$ (sometimes referred to as a *graph shift operator* [65], [66]) that has as candidates the graph adjacency matrix, the graph Laplacian, or any other matrix that captures the relations of network elements. We then consider process $\mathcal{P}$ is described through the linear continuous-time dynamical system

$$\partial_t \boldsymbol{x}(t) = f_x(\boldsymbol{S}_x)\boldsymbol{x}(t) + f_u(\boldsymbol{S}_u)\boldsymbol{u}(t) + \boldsymbol{w}(t) \in \mathbb{R}^N, \tag{3a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{v}(t) \in \mathbb{R}^L, \tag{3b}$$

where $\boldsymbol{C} \in \mathbb{R}^{L \times N}$ and $\boldsymbol{D} \in \mathbb{R}^{L \times N}$ are the system matrices related to the observables $\boldsymbol{y}(t)$. The variables, $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ represent perturbations in the states and additive noise in the

observables, respectively. The matrix function $f_* : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ is defined via the Cauchy integral [67]

$$f_*(\boldsymbol{S}_*) := \frac{1}{2\pi i} \int_{\Gamma_{f_*}} f_{\mathrm{s},*}(z) R(z, \boldsymbol{S}_*) dz, \tag{4}$$

where $f_{\mathrm{s},*}(\cdot)$ is the scalar version of $f_*(\cdot)$ and is analytic on and over the contour Γ_{f_*}. Here, $R(z, \boldsymbol{S}_*)$ is the resolvent of $\boldsymbol{S}_*$ given by

$$R(z, \boldsymbol{S}_*) := (\boldsymbol{S}_* - z\boldsymbol{I})^{-1}. \tag{5}$$

From the dynamical system in (1), it can be seen that (3) represents first-order (linear) differential models, where the system matrices $f_x(\boldsymbol{S}_x)$ and $f_u(\boldsymbol{S}_u)$ are *matrix functions* of the matrix representing the graphs $\boldsymbol{S}_x$ and $\boldsymbol{S}_u$.

Model (3) captures different settings of practical interest such as diffusion on networks [64], graph filtering operations [9], [26], random walks [68], and first-order autoregressive graph processes [69]. The corresponding discrete-time state-space system related to (3) is

$$\boldsymbol{x}(k+1) = \tilde{f}_x(\boldsymbol{S}_x)\boldsymbol{x}(k) + \tilde{f}_u(\boldsymbol{S}_u)\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{6a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + \boldsymbol{v}(k), \tag{6b}$$

where $\tilde{f}_*(\cdot)$ is a matrix function (to be specified in the sequel) and $\boldsymbol{x}(k) \in \mathbb{R}^N$, $\boldsymbol{u}(k) \in \mathbb{R}^N$, and $\boldsymbol{y}(k) \in \mathbb{R}^L$ are the discrete counterparts of $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and $\boldsymbol{y}(t)$, respectively. The variables, $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ represent the discrete counterparts of perturbation $\boldsymbol{w}(t)$ and noise $\boldsymbol{v}(t)$, respectively. By defining then the matrices $\boldsymbol{A}(\boldsymbol{S}_x) := \tilde{f}_x(\boldsymbol{S}_x)$ and $\boldsymbol{B}(\boldsymbol{S}_u) := \tilde{f}_u(\boldsymbol{S}_u)$, the connection between the continuous-time (3) and the discrete-time representation (6) is given by

$$\boldsymbol{A}(\boldsymbol{S}_x) := \tilde{f}_x(\boldsymbol{S}_x) = e^{f_x(\boldsymbol{S}_x)\tau} \tag{7}$$

$$\boldsymbol{B}(\boldsymbol{S}_u) := \tilde{f}_u(\boldsymbol{S}_u) = \left(\int_0^\tau e^{f_x(\boldsymbol{S}_x)t} dt \right) f_u(\boldsymbol{S}_u), \tag{8}$$

where τ is the sampling period and $e^{\boldsymbol{X}} = \sum_{k=0}^{\infty} \frac{1}{k!}\boldsymbol{X}^k$ is the matrix exponential function [67]. Using then (7) and (8), we can compactly write model (6) as a linear discrete-time state-space model

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{9a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + \boldsymbol{v}(k), \tag{9b}$$

where we dropped the dependency of the system matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ from $\boldsymbol{S}_x$ and $\boldsymbol{S}_u$ to simplify the notation. Throughout the paper, we assume that the matrices $\boldsymbol{C}$ and $\boldsymbol{D}$ are known, i.e., we know how the observables are related to the states and inputs.

Remark 1: Although we consider known $\boldsymbol{C}$ and $\boldsymbol{D}$, this assumption can be relaxed as $\boldsymbol{D}$ can be obtained unambiguously and as long as $\boldsymbol{A}$ is full rank, the order of the system can be recovered [54]. However, without explicit knowledge of $\boldsymbol{C}$ disambiguating the involved system matrices requires additional prior knowledge.

IV. State-Space Identification

The family of subspace state-space system identification methods [54] relies on geometrical properties of the dynamical

model (9). By collecting a batch of α different observables over time, $\boldsymbol{y}(t) \in \mathbb{R}^L$, into the $\alpha L-$dimensional vector

$$\boldsymbol{y}_{k,\alpha} \triangleq [\boldsymbol{y}(k)^T, \ldots, \boldsymbol{y}(k+\alpha-1)^T]^T,$$

we get the input-output relationship

$$\boldsymbol{y}_{k,\alpha} = \mathcal{O}_\alpha \boldsymbol{x}(k) + \mathcal{T}_\alpha \boldsymbol{u}_{k,\alpha} + \boldsymbol{n}_{k,\alpha}, \tag{10}$$

where

$$\mathcal{O}_\alpha \triangleq \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{\alpha-1} \end{bmatrix}, \tag{11}$$

is the *extended observability* matrix of the system [70] and

$$\mathcal{T}_\alpha \triangleq \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{B} & \boldsymbol{D} & \ddots & & \cdots & \boldsymbol{0} \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & & & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}^{\alpha-2}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^{\alpha-3}\boldsymbol{B} & \cdots & \cdots & \boldsymbol{C}\boldsymbol{B} & \boldsymbol{D} \end{bmatrix}, \tag{12}$$

is the matrix that relates the batch input vector

$$\boldsymbol{u}_{k,\alpha} \triangleq [\boldsymbol{u}(k)^T, \ldots, \boldsymbol{u}(k+\alpha-1)^T]^T,$$

with the batch observables $\boldsymbol{y}_{k,\alpha}$. The vector $\boldsymbol{n}_{k,\alpha}$ comprises the batch noise that depends on the system perturbation $\{\boldsymbol{w}(k)\}$ and on the observable noise $\{\boldsymbol{v}(k)\}$; the detailed structure of $\boldsymbol{n}_{k,\alpha}$ is unnecessary for our framework. The size of the batch α is user-specified but must be larger than the number of states. Assuming the number of nodes is the number of states, this implies $\alpha > N$.

Given then the input-output relation (10) and the structures for the $\mathcal{O}_\alpha$ in (11) and $\mathcal{T}_\alpha$ in (12), we proceed by estimating first the state matrix $\boldsymbol{A}$ using the algebraic properties of (11) and subsequently the input matrix $\boldsymbol{B}$ from the structure of (12) and a least squares problem.

Retrieving the state matrix $\boldsymbol{A}$. A basic requirement for estimating $\boldsymbol{A}$ is system *observability* [70]. Observability allows to infer the system state from the outputs for any initial state and sequence of input vectors. Put differently, we can estimate the entire system dynamics from input-output relations. System (9) is observable if the system matrices $\{\boldsymbol{A}, \boldsymbol{C}\}$ satisfy $\mathrm{rank}(\mathcal{O}_N) = N$.[1]

Consider now a set of $Q \triangleq T + \alpha - 1$ input-output pairs $\{\boldsymbol{y}(k), \boldsymbol{u}(k)\}_{k=1}^Q$. By stacking the discrete batch vectors $\boldsymbol{y}_{k,\alpha}$ for all observations into the matrix

$$\boldsymbol{Y} = [\boldsymbol{y}_{1,\alpha}, \ldots, \boldsymbol{y}_{T,\alpha}],$$

and using expression (10), we can generate the Hankel-structured data equation [71]

$$\boldsymbol{Y} = \mathcal{O}_\alpha \boldsymbol{X} + \mathcal{T}_\alpha \boldsymbol{U} + \boldsymbol{N}, \tag{13}$$

---

[1]Although this assumption might not hold for all instances in practice, for a full rank and well-conditioned system matrix $\boldsymbol{A}$, this holds. This condition is also required to provide mathematical rigor to the approach and obtain theoretical guarantees.

where $\boldsymbol{X}$ is the matrix that contains the evolution of the states accross the columns, i.e.,

$$\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(T)],$$

and where the input $\boldsymbol{U}$ and noise $\boldsymbol{N}$ matrices are block Hankel matrices defined as

$$\boldsymbol{U} = [\boldsymbol{u}_{1,\alpha}, \ldots, \boldsymbol{u}_{T,\alpha}] \quad \text{and} \quad \boldsymbol{N} = [\boldsymbol{n}_{1,\alpha}, \ldots, \boldsymbol{n}_{T,\alpha}].$$

The structure in (13) is at the core of system identification methods [62] and this arrangement leads naturally to a subspace-based approach to find $\boldsymbol{A}$. To detail this, consider the noise-free case

$$\boldsymbol{Y} = \mathcal{O}_\alpha \boldsymbol{X} + \mathcal{T}_\alpha \boldsymbol{U}. \tag{14}$$

To continue, $\boldsymbol{U}$ should allow us to remove the part of the output, $\boldsymbol{Y}$, that was not generated from the state signal, $\boldsymbol{X}$. That is, given $\boldsymbol{U}$, right-multiplying $\boldsymbol{Y}$ with the projection matrix $\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$ should give us

$$\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp} = \mathcal{O}_\alpha \boldsymbol{X}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}, \tag{15}$$

while ensuring the full row-rankness of $\boldsymbol{X}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$. Given that $\boldsymbol{U}$ is full row-rank, such a projection matrix, $\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$, can be constructed as

$$\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp} \triangleq \boldsymbol{I} - \boldsymbol{U}^T(\boldsymbol{U}\boldsymbol{U}^T)^{-1}\boldsymbol{U}. \tag{16}$$

Since $\boldsymbol{U}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp} = \boldsymbol{0}$, this leads to (15). The condition of the inputs guaranteeing the invertibility of $\boldsymbol{U}\boldsymbol{U}^T$ and the full row-rankness of $\boldsymbol{X}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$ is known as the *persistent excitation* condition [71]. This property requieres the inputs to excite all the modes of the system. Despite that in many scenarios this property might be difficult to enforce, e.g., there is no total control on the inputs of the system, we consider the case where full control of the input signal is guaranteed; hence, as discussed in [71], it is always possible to meet such condition with high probability.

Being able to project out the contribution of the input from the output, while ensuring the rank of the projected state signal matrix, leads to the following property.

$$\mathrm{span}(\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}) = \mathrm{span}(\mathcal{O}_\alpha). \tag{17}$$

That is, the signal subspace of the projected observables $\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$ coincides with that of the extended observability matrix. Therefore, $\mathrm{span}(\mathcal{O}_\alpha)$ can be estimated from $\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$ and, subsequently, the system matrix $\boldsymbol{A}$ by using the block structure of $\mathcal{O}_\alpha$ in (11). To detail this procedure, consider the *economy-size* singular value decomposition (SVD) of $\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$

$$\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp} = (\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp})_N = \boldsymbol{W}_{\alpha,N}\boldsymbol{\Sigma}_N\boldsymbol{V}_{\alpha,N}^T, \tag{18}$$

where $(\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp})_N$ is the N-rank approximation of $\boldsymbol{Y}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$ and equality holds because of the full row-rank assumption of the projected input $\boldsymbol{X}\boldsymbol{\Pi}_{\boldsymbol{U}^T}^{\perp}$. Then, from condition (17), we have

$$\mathcal{O}_\alpha = \boldsymbol{W}_{\alpha,N}\boldsymbol{T}, \tag{19}$$

for some invertible similarity transform matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$. Given $\boldsymbol{C}$ known with full column-rank, we can estimate the transform matrix $\boldsymbol{T}$ from the structure of $\mathcal{O}_\alpha$ [cf. (11)] as

$$\hat{\boldsymbol{T}} = (\boldsymbol{J}\boldsymbol{W}_{\alpha,N})^{\dagger}\boldsymbol{C}, \tag{20}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudoinverse and $\boldsymbol{J}\boldsymbol{W}_{\alpha,N}$ denotes the first L rows of $\boldsymbol{W}_{\alpha,N}$. If $\boldsymbol{C}$ is not full column-rank, the transform matrix $\boldsymbol{T}$ is not unique. We will deal with the non-uniqueness of $\boldsymbol{T}$ in Section V.

Finally, to get $\boldsymbol{A}$, we exploit the *shift invariant* structure of $\mathcal{O}_\alpha$ w.r.t. $\boldsymbol{A}$, i.e.,

$$\boldsymbol{J}_{\mathrm{u}}\mathcal{O}_\alpha \boldsymbol{A} = \boldsymbol{J}_1\mathcal{O}_\alpha \in \mathbb{R}\mathbb{e}^{(\alpha-1)L \times N}, \tag{21}$$

where $\boldsymbol{J}_{\mathrm{u}}\mathcal{O}_\alpha$ and $\boldsymbol{J}_1\mathcal{O}_\alpha$ denote the upper and lower $(\alpha - 1)L$ blocks of $\mathcal{O}_\alpha$ [cf. (11)], respectively. By substituting then the expression (19) for $\mathcal{O}_\alpha$ into (21) and by using the estimate $\hat{\boldsymbol{T}}$ (20) for the transform matrix $\boldsymbol{T}$, the least squares estimate for $\boldsymbol{A}$ is given by

$$\hat{\boldsymbol{A}} = (\boldsymbol{J}_{\mathrm{u}}\boldsymbol{W}_{\alpha,N}\hat{\boldsymbol{T}})^{\dagger}\boldsymbol{J}_1\boldsymbol{W}_{\alpha,N}\hat{\boldsymbol{T}}. \tag{22}$$

Retrieving the input matrix $\boldsymbol{B}$. While the input matrix $\boldsymbol{B}$ can be obtained with a similar approach as $\boldsymbol{A}$ [72] (yet with a more involved shift-invariant structure), we compute it together with the initial state $\boldsymbol{x}(0)$ by solving a least squares problem. To do so, we expand system (9) to all its terms as

$$\boldsymbol{y}(k) - (\boldsymbol{u}(k)^T \otimes \boldsymbol{I}_L)\mathrm{vec}(\boldsymbol{D}) = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{x}(0)$$
$$+ \left(\sum_{q=0}^{k-1} \boldsymbol{u}(q)^T \otimes \boldsymbol{C}\boldsymbol{A}^{k-q-1}\right)\mathrm{vec}(\boldsymbol{B}), \tag{23}$$

where $\boldsymbol{I}_L$ is the $L \times L$ identity matrix and $\mathrm{vec}(\cdot)$ is the vectorization operator. We then collect the unknowns $\boldsymbol{x}(0)$ and $\mathrm{vec}(\boldsymbol{B})$ into vector $\boldsymbol{\theta} = [\boldsymbol{x}(0)^T \, \mathrm{vec}(\boldsymbol{B})^T]^T$ and define the matrix

$$\hat{\boldsymbol{\Psi}} \triangleq \left[\boldsymbol{C}\hat{\boldsymbol{A}}^k, \sum_{q=0}^{k-1} \boldsymbol{u}(q)^T \otimes \boldsymbol{C}\hat{\boldsymbol{A}}^{k-q-1}\right],$$

where we substituted the state transition matrix $\boldsymbol{A}$ with its estimate (22) while the other quantities $\boldsymbol{C}, \boldsymbol{D}$ and $\boldsymbol{u}(k)$ are known. Finally, we get the input matrix $\boldsymbol{B}$ by solving

$$\min_{\boldsymbol{\theta}} \frac{1}{Q} \sum_{k=1}^{Q} \|\boldsymbol{y}(k) - \hat{\boldsymbol{\Psi}}\boldsymbol{\theta}\|_2^2. \tag{24}$$

Given the system matrices $\{\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \boldsymbol{C}, \boldsymbol{D}\}$, the state interaction graph $\mathcal{G}_x$ ($\boldsymbol{S}_x$) and input interaction graph $\mathcal{G}_u$ ($\boldsymbol{S}_u$) can be obtained by enforcing the constraints derived from the information of the physical process, i.e., the model dynamics. Hence, the network structure depends heavily on the estimate of the subspace span of $\mathcal{O}_\alpha$.

A. Noisy Setting

We discuss now a method for estimating $\boldsymbol{A}$ and $\boldsymbol{B}$ with perturbations in the state evolution $\boldsymbol{x}(t)$ and noise in the observables $\boldsymbol{y}(t)$. To tackle these challenges, we leverage *instrumental variables*.

Consider the following partition of the observables and input matrix

$$\boldsymbol{Y} = [\boldsymbol{Y}_1^T, \, \boldsymbol{Y}_2^T]^T \quad \text{and} \quad \boldsymbol{U} = [\boldsymbol{U}_1^T, \, \boldsymbol{U}_2^T]^T,$$

where $\boldsymbol{Y}_1$ (resp. $\boldsymbol{U}_1$) and $\boldsymbol{Y}_2$ (resp. $\boldsymbol{U}_2$) have respectively β and γ blocks of size L with $\gamma = \alpha - \beta$. The model for the observables $\boldsymbol{Y}_2$ is [cf.(13)]

$$\boldsymbol{Y}_2 = \mathcal{O}_\gamma \boldsymbol{X}_2 + \mathcal{T}_\gamma \boldsymbol{U}_2 + \boldsymbol{N}_2. \tag{25}$$

Following then the projection-based strategy to remove the dependency from $\boldsymbol{U}_2$, we can write the projected noisy observables as [cf. (15)]

$$\boldsymbol{Y}_2 \boldsymbol{\Pi}_{\boldsymbol{U}_2^T}^\perp = \mathcal{O}_\gamma \boldsymbol{X}_2 \boldsymbol{\Pi}_{\boldsymbol{U}_2^T}^\perp + \boldsymbol{N}_2 \boldsymbol{\Pi}_{\boldsymbol{U}_2^T}^\perp, \tag{26}$$

with $\boldsymbol{X}_2$ and $\boldsymbol{N}_2$ being the respective partitions of the state evolution and perturbation matrix.

Observe from (26) that the signal subspace is corrupted with the noise projection term $\boldsymbol{N}_2 \boldsymbol{\Pi}_{\boldsymbol{U}_2^T}^\perp$. To remove the latter, we follow the instrumental variable approach. Since the noise is uncorrelated with the input $\boldsymbol{U}_1$ and since the noise present in the second block $\boldsymbol{N}_2$ is uncorrelated with the noise in the first block $\boldsymbol{N}_1$, it holds that

$$\lim_{T \to \infty} \frac{1}{T} \boldsymbol{N}_2 [\boldsymbol{U}_1^T, \, \boldsymbol{Y}_1^T] = [\boldsymbol{0}, \, \boldsymbol{0}]. \tag{27}$$

Subsequently, we define $\boldsymbol{Z}_1 \triangleq [\boldsymbol{U}_1^T, \, \boldsymbol{Y}_1^T]$ and consider the matrix

$$\boldsymbol{G}_1 \triangleq \frac{1}{N} \boldsymbol{Y}_2 \boldsymbol{\Pi}_{\boldsymbol{U}_2^T}^\perp \boldsymbol{Z}_1, \tag{28}$$

to estimate of the signal subspace. Matrix $\boldsymbol{G}_1$ is asymptotically "noise free" due to (27). From the economy-size SVD (N-rank approximation) of $\boldsymbol{G}_1$, we get

$$\boldsymbol{G}_1 \approx \boldsymbol{W}_{\gamma, N} \boldsymbol{\Sigma}_N \boldsymbol{V}_{\gamma, N}^T. \tag{29}$$

Finally, by using $\boldsymbol{W}_{\gamma, N}$, the system matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be estimated from expressions (20), (22), and (24).

Note that the estimator for the signal subspace [cf.(29)] has intrinsic statistical properties. To control the variance and bias, different works have proposed to left and right weigh the matrices in (28) before the SVD [73]. As establishing optimal weighting matrices requires further analysis, we do not detail them here and refer interested readers to [74].

B. Continuous-Time Model Identification

Given the discrete-time system matrices $\{\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \boldsymbol{C}, \boldsymbol{D}\}$, we can estimate the continuous-time transition matrices of (3). Observe from (7) that estimating $\hat{f}_x(\boldsymbol{S}_x)$ from $\hat{\boldsymbol{A}}$ requires computing only the matrix logarithm of $\hat{\boldsymbol{A}}$. Nevertheless, as stated by the following proposition, there are conditions process $\mathcal{P}$ should meet for this matrix logarithm to (i) exist and (ii) be unique. For clarity, the involved matrices are *real* since we have matrix functions that map real matrices onto real matrices.

Proposition 1: Let the analytic function $f_{\text{s,x}}(\cdot)$ of $f_x(\boldsymbol{S}_x)$ in (7) satisfy

   a) $e^{f_{\text{s,x}}(z)} \notin \mathbb{R}^-, \, \forall \, z \in \text{eig}(\boldsymbol{S}_x)$
   b) $f_{\text{s,x}}(z) > -\infty, \, \forall \, z \in \text{eig}(\boldsymbol{S}_x)$

where $\mathbb{R}^-$ is the closed negative real axis. Then, process $\mathcal{P}$ guarantees that (i) and (ii) are met.

If the conditions of Proposition 1 are met for $f_{\text{s,x}}(\cdot)$, then $\boldsymbol{A}$ has no eigenvalues on $\mathbb{R}^-$. This implies that the principal logarithm[2] of $\boldsymbol{A}$, $\ln(\boldsymbol{A}) = \tau f_x(\boldsymbol{S}_x)$, exists and is unique. Note that if $\boldsymbol{A}$ is real, its principal logarithm is also real. Hence, if $f_{\text{s,x}}(\cdot)$ satisfies the conditions of Proposition 1, the continuous-time transition matrices for nonsingular $\hat{\boldsymbol{A}} - \boldsymbol{I}$ are

$$\hat{f}_x(\boldsymbol{S}_x) = \frac{1}{\tau} \ln(\hat{\boldsymbol{A}}) \tag{30a}$$

$$\hat{f}_u(\boldsymbol{S}_u) = (\hat{\boldsymbol{A}} - \boldsymbol{I})^{-1} \hat{f}_x(\boldsymbol{S}_x) \hat{\boldsymbol{B}}. \tag{30b}$$

The expression for $\hat{f}_u(\boldsymbol{S}_u)$ is derived from

$$\int_0^\tau e^{f_x(\boldsymbol{S}_x)t} dt = f_x(\boldsymbol{S}_x)^{-1} (e^{f_x(\boldsymbol{S}_x)\tau} - \boldsymbol{I}). \tag{31}$$

Given $\hat{f}_x(\boldsymbol{S}_x)$ and $\hat{f}_u(\boldsymbol{S}_u)$, we are left to estimate of the underlying topologies. Depending on the available prior information, the network topology can be estimated with methods promoting particular properties. As $\hat{f}_x(\boldsymbol{S}_x)$ and $\hat{f}_u(\boldsymbol{S}_u)$ are matrix functions of $\boldsymbol{S}_x$ and $\boldsymbol{S}_u$, respectively, these matrices share their eigenvectors with the corresponding network matrices. Then, when the mappings $f_{s,x}(\cdot)$ and $f_{s,u}(\cdot)$ are known, the network matrices can be recovered by applying the inverse maps $\{f_{s,x}^{-1}(\cdot), f_{s,u}^{-1}(\cdot)\}$ (if they exist) or by solving a (possibly) nonlinear root finding problem. When $f_{s,x}(\cdot)$ and $f_{s,u}(\cdot)$ are unknown but structural properties such as sparsity are assumed, methods using the eigenvectors of $\hat{f}_x(\boldsymbol{S}_x)$ and $\hat{f}_u(\boldsymbol{S}_u)$ can be employed to find the corresponding network matrices, $\boldsymbol{S}_x$ and $\boldsymbol{S}_u$. Examples of methods that make use of such an eigenbasis knowledge to recover the network topology can be found in [15], [75]–[77] and references therein. As the main focus of this work is not on the full-observability case, e.g., all node signals are observed, we do not further discuss this, and focus now our attention on the case of partial observability. Full-observability network topology identification has been extensively studied in recent times; see e.g., [22].

For the case of partial observations, the similarity transform $\boldsymbol{T}$ cannot be uniquely identified making the above methods not applicable for identifying the network topology due to the ambiguities in the system matrices. Therefore, in the next section, we next discuss the existing ambiguities in the solution and introduce an AP method for estimating the network topology in this setting.

Remark 2: Although we focus principally on continuous-time models, all results hold also for purely discrete models with appropriate minor changes in the functional dependencies of the system matrices.

V. PARTIALLY OBSERVED NETWORK

So far, we considered the observables $\boldsymbol{y}(t)$ are available for the whole network, i.e., $\boldsymbol{C} = \boldsymbol{I}$ or more generally $\text{rank}(\boldsymbol{C}) = N$.

---

[2]For two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\boldsymbol{X}$ is said to be the matrix logarithm of $\boldsymbol{Y}$ if $e^{\boldsymbol{X}} = \boldsymbol{Y}$. If a matrix in invertible and has no-negative real eigenvalues, there is a unique logarithm which is called principal logarithm [67].

The latter allows a unique estimate for the similarity matrix $\boldsymbol{T}$ in (20). We here move to the more involved case where we cannot observe the process on all nodes, i.e., $\boldsymbol{C} \in \{0, 1\}^{L \times N}$ is a selection matrix, or the observations are not sufficiently rich, i.e., $\text{rank}(\boldsymbol{C}) < N$. This setting appears in biological or chemical networks [37] where the trajectories of the genes or compounds are not directly measurable but only a few observables capturing mixtures of them. Hidden nodes are also present in networks for which no direct information is available [47]. For example, in epidemics spreading, the original carrier of a virus may be hidden; in ecological networks, measurements cannot be retrieved from all ecological niches due to budgeted constraints. Finally, in social network difussions [78], hidden users influencing the network dynamics have their data (information) hidden from third-party collectors. Hence, an analysis of the inference limits, similar to the one presented in [56] for network discovery, in these scenarios is required.

When matrix $\boldsymbol{C}$ is rank deficient, we cannot find a unique transform matrix $\boldsymbol{T}$, hence instead of retrieving the original system matrices, we estimate a set of equivalent matrices

$$\boldsymbol{A}_{\text{T}} \triangleq \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}, \quad \boldsymbol{B}_{\text{T}} \triangleq \boldsymbol{T}\boldsymbol{B}, \tag{32}$$

which also *realize* the system in (10). It follows from (32) that (although $\boldsymbol{A}_{\text{T}} \neq \boldsymbol{A}$ in general) if $\boldsymbol{A}$ is diagonalizable as $\boldsymbol{A} = \boldsymbol{Q}_A \boldsymbol{\Lambda}_A \boldsymbol{Q}_A$, then

$$\text{eig}(\boldsymbol{A}) = \text{eig}(\boldsymbol{A}_{\text{T}}), \tag{33}$$

holds, where $\text{eig}(\boldsymbol{A})$ are the eigenvalues of $\boldsymbol{A}$. The equality (33) yields since $\boldsymbol{A}$ and $\boldsymbol{A}_{\text{T}}$ are *similar* matrices.[3]

In these situations, we cannot remove the ambiguity in the system matrices without additional information. In the sequel, we motivate next why this disambiguation problem is particularly hard. We further derive a method to estimate an *approximately feasible* realization of the *network topology* related to the signal subspace and to the knowledge of the (bijective) scalar mappings $\{f_{s,x}(\cdot), f_{s,u}(\cdot)\}$.

Finally, despite we concentrate in recovering the state network topology $\mathcal{G}_x$, this is not a problem for cases when the input network topology $\mathcal{G}_u$ is of interest. For such cases, an additional step can always be performed to retrieve $\boldsymbol{S}_u$ from the transformed matrix $\boldsymbol{B}_{\text{T}}$. For instance, from the estimate of $\boldsymbol{A}$ based on the state network topology and $\boldsymbol{A}_T$, we can estimate the transform matrix $\hat{\boldsymbol{T}}$. Subsequently, the inverse operation to $\boldsymbol{B}_T$ to get $\hat{\boldsymbol{B}}$ and, therefore, $\mathcal{G}_u$.

A. The Graph Inverse Eigenvalue Problem

To start, consider the shift operator $\boldsymbol{S}_x$ belongs to a set $\mathcal{S}$ which contains all permissible matrices representing the state dynamics that lead to $\boldsymbol{A}_{\text{T}}$. Set $\mathcal{S}$ describes the properties of the graph representation matrix, e.g., zero diagonal (adjacency) $[\boldsymbol{S}_x]_{n,n} = 0 \, \forall n \in [N]$, unitary diagonal (normalized Laplacian) $[\boldsymbol{S}_x]_{n,n} = 1 \, \forall n \in [N]$, zero eigenvalue related to the constant eigenvector (combinatorial Laplacian) $\boldsymbol{S}_x \boldsymbol{1} = \boldsymbol{0}$, symmetry (undirected graphs) $\boldsymbol{S}_x = \boldsymbol{S}_x^T$.

---

[3]Two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are said to be similar if there exists an invertible matrix $\boldsymbol{P}$ such that $\boldsymbol{X} = \boldsymbol{P}\boldsymbol{Y}\boldsymbol{P}^{-1}$.
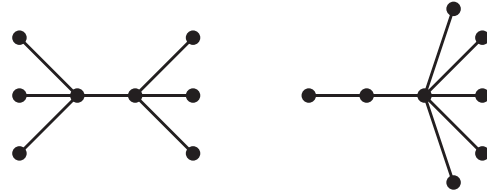

Fig. 1. Two cospectral trees with the same number of edges. With respect to the adjacency matrix, almost all trees are non determined by their spectrum. Both graphs have the same characterisitic polynomial $t^4(t^4 - 7t^2 + 9)$, hence cospectral.

The ambiguity (33) introduced by the similarity transform matrix $\boldsymbol{T}$ transforms the problem of finding $\boldsymbol{S}_x$ into

$$\begin{aligned} \text{find} \quad & \boldsymbol{S} \\ \text{subject to} \quad & \boldsymbol{S} \in \mathcal{S} \\ & \text{eig}(\boldsymbol{S}) = \boldsymbol{\lambda}_x, \end{aligned} \tag{34}$$

where $\boldsymbol{\lambda}_x = \text{eig}(\boldsymbol{S}_x)$ is the vector containing the eigenvalues of $\boldsymbol{S}_x$ obtained by applying the inverse map to the eigenvalues of $\boldsymbol{A}_T$. Problem (34) recasts the network topology identification problem to that of finding a graph shift operator matrix $\boldsymbol{S}_x$ that has a fixed spectrum. This problem belongs to the family of *inverse eigenvalue problems* [79]. In a way, problem (34) is the complement of the network spectral template approach [15]. Here, instead of having an eigenbasis and searching for a set of eigenvalues, we have a set of eigenvalues and search for an eigenbasis.

Problem (34) is ill-posed since its solution is non-unique in most cases. Thus, it leads to ambiguities in its solution. In what follows, we characterize this ambiguity in terms of equivalence classes between graphs and provide a method able to find a network topology satisfying the conditions of (34).

B. Ambiguous Graphs: Cospectral Graphs

We find ambiguities in graphs that belong to an equivalence class [80]. An example of spectrally equivalent graphs are isomorphic graphs [55]. Two graphs $\mathcal{G}$ and $\mathcal{G}'$ with respective graph shift operator matrices $\boldsymbol{S}$ and $\boldsymbol{S}'$ are equivalent if there exists a permutation matrix $\boldsymbol{P}$ such that

$$\boldsymbol{S} = \boldsymbol{P}\boldsymbol{S}'\boldsymbol{P}^T. \tag{35}$$

That is, the graph representation matrices are row- and column-permuted versions of each other. The permutation matrix $\boldsymbol{P}$ implements the isomorphism. Therefore, if only the graph eigenvalues $\boldsymbol{\lambda}$ are available, the graphs are always indistinguishable up to node reordering. This situation is not at all undesirable as the ordering of the nodes is often not important. However, isomorphic graphs are not the only ones sharing the spectrum.

Graphs that share the spectrum are called *cospectral* (or isospectral) graphs [81]. Note, however, that cospectral graphs are not necessarily isomorphic. Fig. 1 illustrates an example of two cospectral graphs. Graph cospectrallity renders the feasible set of (34) not a singleton and, therefore, we need to settle with any feasible graph satisfying the constraints. Put is simply,

the identified topology from (34) will be a graph within the equivalence class of cospectral graphs regarding $\boldsymbol{\lambda}$ and $\mathcal{S}$. The following definition formalizes the latter.

Definition 1: (Equivalent cospectral graphs) Two graphs $\mathcal{G}$ and $\mathcal{G}'$ of N nodes are cospectral equivalents with respect to the spectrum $\boldsymbol{\lambda}$ and the graph representation set $\mathcal{S}$, if they belong to the set

$$\mathcal{C}_{\mathcal{S}}^{\boldsymbol{\lambda}} := \{\mathcal{G} \,|\, \boldsymbol{S} \in \mathcal{S}, \, \mathrm{eig}(\boldsymbol{S}) = \boldsymbol{\lambda}\}.$$

As a result, the feasibility problem (34) reduces to a *graph construction* problem (graph inverse eigenvalue problem). So, the problem at hand can be rephrased as *given a spectrum $\boldsymbol{\lambda}$ and a set $\mathcal{S}$, construct a graph shift operator matrix $\boldsymbol{S} \in \mathcal{S}$ with spectrum $\boldsymbol{\lambda}$*. We shall discuss next a method that addresses this construction problem.

C. Graph Construction by Alternating Projections

Before detailing the graph construction method, we introduce the following assumptions.

A.1) The set $\mathcal{S}$ is closed.

A.2) For any $\boldsymbol{S} \in \mathbb{R}^{N \times N}$, the projection $P_{\mathcal{S}}(\boldsymbol{S})$ of $\boldsymbol{S}$ onto the set $\mathcal{S}$ is unique.

The first assumption is technical and guarantees set $\mathcal{S}$ includes all its limit points. The second assumption is slightly more restrictive and ensures the problem

$$P_{\mathcal{S}}(\boldsymbol{S}) := \underset{\hat{\boldsymbol{S}}}{\text{minimize}} \, \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_{\mathrm{F}}, \text{ s.t. } \hat{\boldsymbol{S}} \in \mathcal{S}, \quad (36)$$

has a unique solution. Although this assumption might seem restrictive, in most cases we only have access to a convex description of the feasible set $\mathcal{S}$ which satisfies A.2 (as the set is assumed closed) or only to the projection onto the convex approximation of the feasible set. Thus, it is fair to consider A.2 holds in practice. To ease exposition, we focus on the case of symmetric matrices, i.e., undirected graphs but remark that a similar approach can be followed for directed graphs.[4] Further, denote by $\mathcal{S}^N$ the set of symmetric $N \times N$ matrices and by $\mathcal{S}_+^N$ the set of positive semidefinite matrices. We then recall the following result from [83, Thm. 5.1].

Theorem 1 (adapted): Given $\boldsymbol{S} \in \mathcal{S}^N$ and let $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ be the spectral decomposition of $\boldsymbol{S}$ with non-increasing eigenvalues $[\boldsymbol{\Lambda}]_{ii} \geq [\boldsymbol{\Lambda}]_{jj}$ for $i < j$. For a fixed $\boldsymbol{\Lambda}_o \in \mathcal{D}_N$ with non-increasing elements $[\boldsymbol{\Lambda}_o]_{ii} \geq [\boldsymbol{\Lambda}_o]_{jj}$ for $i < j$, a best approximant, in the Frobenius norm sense, of $\boldsymbol{S}$ in the set of matrices with fixed eigenvalues

$$\mathcal{M} := \{\boldsymbol{M} \in \mathcal{S}^N \,|\, \boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}_o\boldsymbol{V}^T, \boldsymbol{V} \in \mathrm{O}(N)\},$$

is given by

$$P_{\mathcal{M}}(\boldsymbol{S}) := \boldsymbol{Q}\boldsymbol{\Lambda}_o\boldsymbol{Q}^T,$$

where $\mathrm{O}(N)$ denotes the set of the $N \times N$ orthogonal matrices.

Theorem 1 implies the projection of $\boldsymbol{S}$ onto $\mathcal{M}$ is not necessarily *unique*. As an example, consider the graph shift operator

$\boldsymbol{S}$ with repeated eigenvalues. Here, it is not possible to uniquely define a basis for the directions related to the eigenvalues with multiplicity larger than one. Hence, infinitely many eigendecompositions exist that lead to many projections of $\boldsymbol{S}$ onto $\mathcal{M}$. Since every element of $\mathcal{M}$ is uniquely determined by an element of $\mathrm{O}(N)$, the structure of $\mathcal{M}$ is completely defined by the structure of $\mathrm{O}(N)$. Therefore, as $\mathrm{O}(N)$ is a *smooth manifold*, $\mathcal{M}$ is one as well.

Alternating projections method. As it follows from Assumptions A.1 and A.2 and Theorem 1, we can project any graph shift operator matrix $\boldsymbol{S} \in \mathcal{S}^N$ onto $\mathcal{S}$ and $\mathcal{M}$. Further, by noticing that the construction problem (34) is equivalent to finding a matrix in

$$\mathcal{S} \cap \mathcal{M}, \quad (37)$$

we can consider the *alternating projections* (AP) [59] to find a point in (37). The AP method finds a point in the intersection of two *closed convex* sets by iteratively projecting a point onto the two sets. It performs the updates

$$\boldsymbol{S}_{k+1/2} = P_{\mathcal{S}}(\boldsymbol{S}_k) \quad (38\text{a})$$

$$\boldsymbol{S}_{k+1} \in P_{\mathcal{M}}(\boldsymbol{S}_{k+1/2}), \quad (38\text{b})$$

starting from a point $\boldsymbol{S}_0 \in \mathcal{M}$. The AP method has guaranteed convergence for convex sets and it does that linearly. However, for alternating projections on a combination of different types of sets (we have a set $\mathcal{S}$ satisfying A.1 and A.2, and a smooth manifold $\mathcal{M}$), additional conditions on both sets are necessary to guarantee convergence.

First, let us formalize the notion of a fixed point for the iterative procedure (38).

Definition 2 (Fixed point): A matrix $\boldsymbol{S} \in \mathcal{S}^N$ is a fixed point of the alternating projections procedure in (38) if there exists an eigendecomposition of $P_{\mathcal{S}}(\boldsymbol{S})$,

$$P_{\mathcal{S}}(\boldsymbol{S}) = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \in \mathcal{S},$$

with non-increasing elements $[\boldsymbol{\Lambda}]_{ii} \geq [\boldsymbol{\Lambda}]_{jj}$ if $i < j$ such that

$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}_o\boldsymbol{Q}^T \in \mathcal{M}.$$

This definition makes explicit two things. First, it defines $\boldsymbol{S}$ as a fixed point if and only if

$$\boldsymbol{S} \in P_{\mathcal{M}}(P_{\mathcal{S}}(\boldsymbol{S})). \quad (39)$$

Second, when $\boldsymbol{S}$ is a fixed point and $P_{\mathcal{S}}(\boldsymbol{S})$ has eigenvalues with multiplicity larger than one, progress can still be made towards a feasible solution when $P_{\mathcal{M}}(P_{\mathcal{S}}(\boldsymbol{S})) \notin \mathcal{S} \cap \mathcal{M}$. To see the latter, consider the case where an alternative eigendecomposition

$$P_{\mathcal{S}}(\boldsymbol{S}) = \tilde{\boldsymbol{Q}}\boldsymbol{\Lambda}\tilde{\boldsymbol{Q}}^T, \quad (40)$$

is available for the fixed point $\boldsymbol{S}$. Assuming that

$$\tilde{\boldsymbol{S}} = \tilde{\boldsymbol{Q}}\boldsymbol{\Lambda}_o\tilde{\boldsymbol{Q}} \neq \boldsymbol{Q}\boldsymbol{\Lambda}_o\boldsymbol{Q}^T = \boldsymbol{S}, \quad (41)$$

with $\boldsymbol{Q}$ the eigenbasis that makes $\boldsymbol{S}$ a fixed point, we can see that the new point $\tilde{\boldsymbol{S}}$ escapes from the fixed point. Further, since the successive projections between two closed sets is a nonincreasing function over the iterations [84, Thm. 2.3], we

---

[4]This could be done by exchanging the spectral decomposition for the Schur decomposition [82] which decomposes a matrix into unitary matrices and an upper triangular matrix.

can prove that $\tilde{S}$ presents a progress towards a feasible solution in $\mathcal{S} \cap \mathcal{M}$.[5]

With this in place, the following theorem shows that the AP method converges in the graph construction problem.

Theorem 2: Let $\mathcal{S}$ meet Assumptions A.1–A.2 and consider the set $\mathcal{M}$ defined in Theorem 1. Let also $S_0, S_1, S_2, \ldots$, be a sequence generated by the alternating projections method in (38). Then, there exists a limit point S of this sequence that is a fixed point of (38) [cf. 2] satisfying

$$\|S - P_{\mathcal{S}}(S)\| = \lim_{k \to \infty} \|S_k - P_{\mathcal{S}}(S_k)\|.$$

If the limit is zero, then $S \in \mathcal{S} \cap \mathcal{M}$.

Proof: See Appendix IX-A.[6] ∎

The above theorem proves the AP method retrieves a matrix $S \in \mathcal{M}$ that realizes the original system, i.e., it preserves the underlying system modes and is an approximately *feasible network representation*. Nevertheless, Theorem 2 does not quantify the rate of convergence of such a method. By particularizing results for super-regular sets [85, Thm. 5.17], the following theorem shows that if the problem is feasible, locally, the proposed method converges linearly to a point in (37).

Theorem 3: Let the set of all permissible matrices $\mathcal{S}$ [cf. (34)] be convex and meet Assumptions A.1–A.2. Let also the set $\mathcal{M}$ be defined as in Theorem 1. Denote by $N_{\mathcal{S}}(S)$ the normal cone of the closed set $\mathcal{S}$ at a point S and, similarly, by $N_{\mathcal{M}}(S)$ the normal cone of the set $\mathcal{M}$ at S. Further, suppose that $\mathcal{M}$ and $\mathcal{S}$ have a strongly regular intersection at $\bar{S}$, i.e., the constant

$$\bar{c} = \max\{\langle u, v \rangle : u \in N_{\mathcal{M}}(\bar{S}) \cap B, v \in -N_{\mathcal{S}}(\bar{S}) \cap B\},$$

is strictly less than one with B being a closed unit Euclidean ball. Then, for any initial point $S_0 \in \mathcal{M}$ close to $\bar{S}$, any sequence generated by the alternating projections method in (38) converges to a point in $\mathcal{M} \cap \mathcal{S}$ with R-linear rate

$$r \in (\bar{c}, 1).$$

Proof: See Appendix IX-B ∎

These results guarantee that the AP method converges *globally* (at least) to a fixed point in $\mathcal{M}$ and *locally*, i.e., within the neighborhood of the solution (if it exists), to a fixed point in $\mathcal{M} \cap \mathcal{S}$.

D. Inaccurate and Partial Information

We now consider the case where the estimated eigenvalues are inexact because of noise or are incomplete because the full eigendecomposition of the system matrices is not feasible. To deal with such cases, we modify the structure of the set $\mathcal{M}$ (in Theorem 1) to reflect the uncertainty and the partial eigendecomposition. The modified set has to be compatible with the structure used in Definition 2 and Theorems 2 and 3 to guarantee the convergence of the AP method. Thus, in the sequel,

we focus on proving *compactness* for the modified versions of $\mathcal{M}$, which suffices to guarantee convergence of the AP method by the result of Theorem 2.

Uncertainty in the system matrices. The following proposition shows that the set $\mathcal{M}_\epsilon$, which allows the estimated eigenvalues to lie within an ϵ−uncertainty ball, is compact.

Proposition 2: Let $\Lambda_o \in \mathcal{D}_N$ with $[\Lambda_o]_{ii} \geq [\Lambda_o]_{jj}$ for $i < j$ be fixed. If $0 \leq \epsilon < \infty$ is a fixed scalar accounting for uncertainties on the elements of Λ_o, then the set

$$\mathcal{M}_\epsilon := \{M \in \mathcal{S}^N | M = V(\Lambda_o + \Lambda_\epsilon)V^T, V \in \mathrm{O}(N),$$
$$\Lambda_\epsilon \in \mathcal{D}_N, \|\Lambda_\epsilon\|_2 \leq \epsilon\} \qquad (42)$$

is compact.

Proof: See Appendix IX-C ∎

The following result provides a best approximant of a matrix S in the set $\mathcal{M}_\epsilon$ in the Frobenius norm.

Theorem 4: Given $S \in \mathcal{S}^N$ with eigendecomposition $S = Q\Lambda Q^T$ and non-increasing eigenvalues $[\Lambda]_{ii} \geq [\Lambda]_{jj}$ for $i < j$. For a fixed $\Lambda_o \in \mathcal{D}_N$ with $[\Lambda_o]_{ii} \geq [\Lambda_o]_{jj}$ for $i < j$, a best approximant of S in $\mathcal{M}_\epsilon$, in the Frobenius norm sense, is given by

$$P_{\mathcal{M}_\epsilon}(S) := Q(\Lambda_o + \Lambda_\epsilon^*)Q^T,$$

where

$$\Lambda_\epsilon^* := \operatorname*{argmin}_{\Lambda_\epsilon \in \mathcal{D}_N} \|\Lambda - \Lambda_o - \Lambda_\epsilon\|_{\mathrm{F}}, \text{ s.t } \|\Lambda_\epsilon\|_2 \leq \epsilon.$$

Proof: See Appendix IX-D ∎

Corollary 1: The nonzero entries of Λ_ϵ^* are

$$[\Lambda_\epsilon^*]_{ii} = \mathrm{sign}(\gamma_i) \cdot \min\{\epsilon, |\gamma_i|\},$$

with $\gamma_i := [\Lambda]_{ii} - [\Lambda_o]_{ii}$.

Partial eigendecomposition. In physical systems, a discrete model of N degrees of freedom provides accurate information of about $N/3$ of the system natural frequencies [79, Ch. 5]. In other cases, the full eigendecomposition of the system matrix is not always possible. We, therefore, provide a projection onto a set that only considers a noisy part of the system matrix spectrum is available.

The following theorem provides the main result.

Theorem 5: Let $\Lambda_m \in \mathcal{D}_m$ with $[\Lambda_m]_{ii} \geq [\Lambda_m]_{jj}$ for $i < j$ be fixed. If $0 \leq \epsilon < \infty$ is a fixed scalar accounting for uncertainties on the elements of Λ_m and $\rho := \max_{S \in \mathcal{S}} \|S\|_2$, then a best approximant, in the Frobenius norm sense, of $S \in \mathcal{S}^N$, with $\|S\|_2 \leq \rho$, in the set

$$\mathcal{M}_\epsilon^m := \{M \in \mathcal{S}^N | M = V\mathrm{bdiag}(\Lambda_m + \Lambda_\epsilon, \bar{\Lambda})V^T,$$
$$V \in \mathrm{O}(N), \Lambda_\epsilon \in \mathcal{D}_m, \|\Lambda_\epsilon\|_2 \leq \epsilon, \bar{\Lambda} \in \mathcal{D}_{N-m}, \|\bar{\Lambda}\|_2 \leq \rho\} \qquad (43)$$

is given by

$$P_{\mathcal{M}_\epsilon^m}(S) := Q\mathrm{bdiag}(\Lambda_m + \Lambda_\epsilon^*, \Lambda_{\bar{\sigma}})Q^T.$$

Here, σ denotes the permutation of the subset of $[N]$ that solves the combinatorics problem

$$\min_{1 \leq [\sigma]_1 < \ldots < [\sigma]_m \leq N} \sum_{i=1}^m ([\Lambda]_{[\sigma]_i[\sigma]_i} - [\Lambda_m]_{ii})^2, \qquad (44)$$

---

where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues of $\boldsymbol{S}$, and $\bar{\boldsymbol{\sigma}}$ is the complementary set of $\boldsymbol{\sigma}$. Matrix $\boldsymbol{Q}$ is given by the (sorted) eigendecomposition of $\boldsymbol{S}$, i.e.,

$$\boldsymbol{S} = \boldsymbol{Q}\mathrm{bdiag}(\boldsymbol{\Lambda_{\sigma}}, \boldsymbol{\Lambda_{\bar{\sigma}}})\boldsymbol{Q}^T,$$

where $\boldsymbol{\Lambda_{\sigma}}$ is the permuted version of $\boldsymbol{\Lambda}$ and

$$\boldsymbol{\Lambda}_\epsilon^* := \underset{\boldsymbol{\Lambda}_\epsilon \in \mathcal{D}_m}{\mathrm{argmin}} \|\boldsymbol{\Lambda_{\sigma}} - \boldsymbol{\Lambda}_m - \boldsymbol{\Lambda}_\epsilon\|_\mathrm{F}, \text{ s.t.} \|\boldsymbol{\Lambda}_\epsilon\|_2 \leq \epsilon.$$

Furthermore, the set $\mathcal{M}_\epsilon^m$ is compact.

Proof: See Appendix IX-F ∎

Corollary 2: The optimal permutation $\boldsymbol{\sigma}$ of the indices $[N]$ that solves (44) can be found by solving a minimum-weight bipartite perfect matching problem.

Put simply, Theorems 4 and 5 show the sets $\mathcal{M}_\epsilon$ and $\mathcal{M}_\epsilon^m$ are compact and provide a best Frobenius-norm approximant for $\boldsymbol{S}$ in each case. Therefore, we can apply the AP method in (38) to these scenarios using the appropriate modifications. Finally, since sets $\mathcal{M}_\epsilon$ and $\mathcal{M}_\epsilon^m$ meet the conditions of Theorem 2, the convergence results for the AP method extend also to these scenarios.

VI. SYSTEM CONSISTENCY CONSTRAINTS

The set $\mathcal{S}$ [cf. (34)] plays an important role in the system topology that the AP method identifies. As such, it should be constrained such that the AP method yields a consistent system, i.e., the AP estimated state network should define an equivalent system to the original one. We briefly discuss here two constraints that can be added to $\mathcal{S}$ to enforce system consistency.

By considering the system matrix [cf. (7)] is a bijective matrix function and by using the same construction as for the shift invariance property in (21), we can build the linear system

$$\begin{bmatrix} \boldsymbol{C}_T \\ \boldsymbol{C}_T f_x^{-1}(\boldsymbol{A}_T) \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{S}_x \end{bmatrix}. \tag{45}$$

Here, we leverage the invariance of the matrix function to nonsingular transforms, i.e.,

$$f(\boldsymbol{A}_T) = \boldsymbol{T}f(\boldsymbol{A})\boldsymbol{T}^{-1}, \tag{46}$$

where $f(\cdot)$ is a matrix function, hence, can be applied to $\boldsymbol{A}_T$. In this way, we get a linear system that depends on $\boldsymbol{S}_x$ and enforces the shift invariance condition. Nevertheless, the shift invariance condition does not change the optimality nor the uniqueness of the projection onto $\mathcal{S}$. This is because $\boldsymbol{T}$ is a free optimization variable and does not affect the projection distance [cf. (36)].

If other constraints for the transform matrix $\boldsymbol{T}$ are known, they can be included when solving for the projection (36). These additional constraints will not impact the projection optimality because they do not change the cost function for $P_{\mathcal{S}}$. For instance, consider the constraint that requires symmetry in the input matrix $\boldsymbol{B}$. Then, since $\boldsymbol{B}_T = \boldsymbol{T}\boldsymbol{B}$, $\boldsymbol{B}$ is symmetric, and $\boldsymbol{B}_T\boldsymbol{T} = \boldsymbol{T}\boldsymbol{B}\boldsymbol{T}^T$, we have

$$\boldsymbol{B}_T\boldsymbol{T}^T = \boldsymbol{T}\boldsymbol{B}_T^T. \tag{47}$$

Again, such an additional constraint does not change the projection distance as it only modifies the description of the convex set in which the matrix must be projected.

We can introduce other constraints to the set $\mathcal{S}$ to further restrict the family of feasible network representations. However, these constraints are case-specifc and go beyond the main goal of this work. Next, we corroborate the above theoretical findings with numerical results.

VII. NUMERICAL RESULTS

In this section, we present a series of numerical results to illustrate the performance of the proposed methods for different scenarios. We first illustrate how the model and the noise coloring influence the estimation performance of commonly used topology identification methods. Then, we corroborate our theoretical results. Finally, we present results for the topology identification from partially-observed networks.[7]

A. Discrete Model Validation

In this section, we corroborate the discrete model (6) in finding a graph from continuous-time data generated following the model (3). The underlying graphs are two fixed random regular graphs of $N = 50$ nodes with node degree $d = 3$. The data are generated by a continuous-time solver with system evolution matrix $f_x(\boldsymbol{S}_x) = -\boldsymbol{S}_x$ and input matrix $f_u(\boldsymbol{S}_u) = -(\boldsymbol{S}_u + \boldsymbol{I})$. The observable matrix $\boldsymbol{C}$ is set to identity and $\boldsymbol{D}$ is the zero matrix. The input signal is drawn from a standard normal distribution and we set the number of samples to N^3 with a sampling time of $\tau = 10^{-3}$.

We compare the proposed method with the spectral templates techniques in [15]. For the latter, the system matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are first obtained from the continuously sampled data (cf. Section III). Then, the eigenvectors of these matrices are used as spectral templates. These results are shown in Fig. 2.

Fig. 2 (a) shows the estimated network topologies for a particular input signal realization, while Fig. 2 (b) compares the respective spectra. The spectral templates overestimates the number of edges and underestimate the graph eigenvalues. However, the graph obtained with spectral templates is a *matrix function* of the original graph, i.e., there is a function (polynomial) that maps the estimated graph to the original one. This is because $\boldsymbol{S}_x$ has all eigenvalues with multiplicity one. The proposed technique relying on the discrete model (6) retrieves the eigenvalues and the graph structure perfectly. This result is not surprising since subspace-based system identification is a consistent estimator for the transition matrix and the proposed method use the knowledge of $f_{s,x}(\cdot)$ while the spectral templates does not. For this scenario, we also considered building the graph from the data covariance matrix, but this technique did not lead to satisfactory results. We attribute this misbehavior to the fact that the covariance matrix is not diagonalizable by the graph modes (i.e., eigenvectors of $\boldsymbol{S}_x$) due to the presence of an input signal.

B. Instrumental Variables Approach on Social Graph

We now evaluate the instrumental variable approach of Section IV-A on the Karate club graph [86]. The graph represents

---

[7]The code to generate these numerical results can be found in [Online]. Available: https://gitlab.com/fruzti/systemid_codes
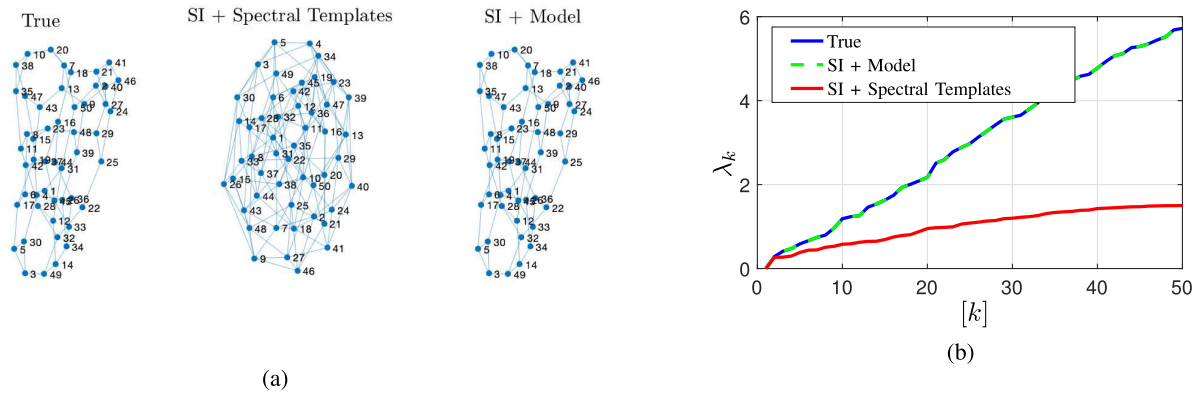
(a)

(b)

Fig. 2. Comparison of the spectral template method within the system identification framework and the model-aware method. (a) Reconstructed state graph. (b) Comparison of eigenvalues of the estimated graphs.
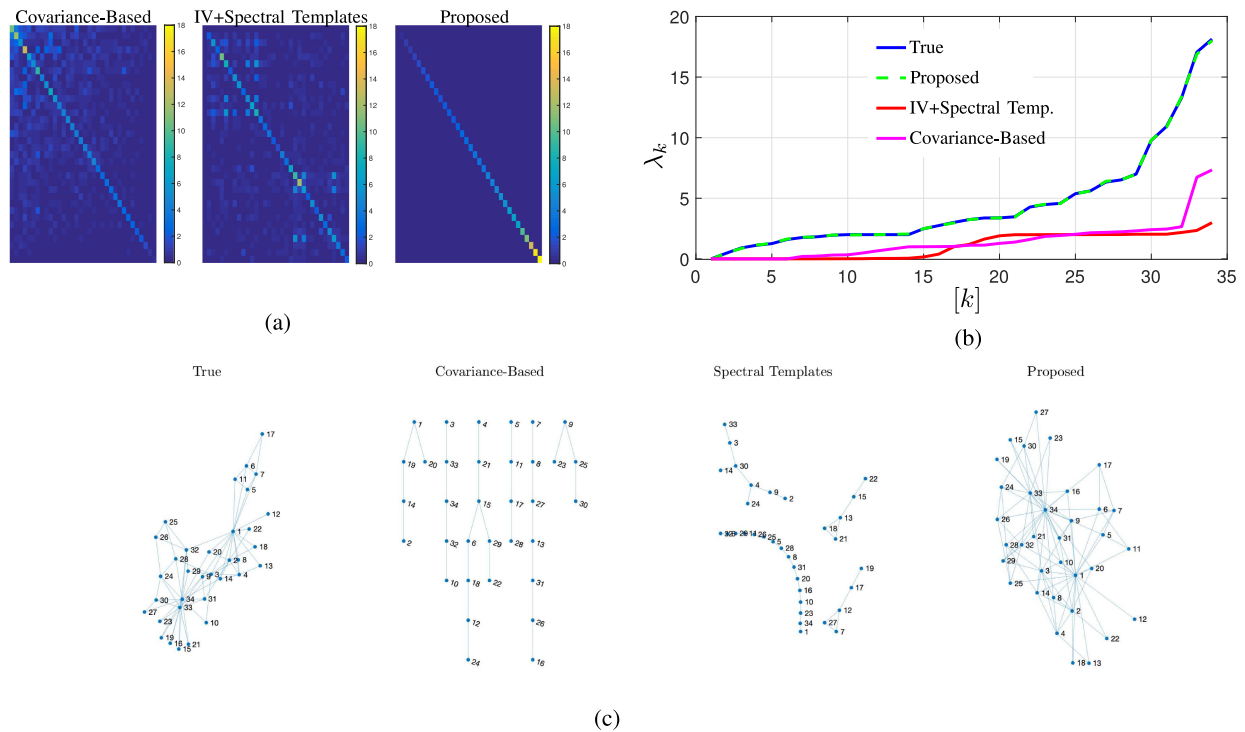

(a)

(b)

(c)

Fig. 3. Comparison of several methods using and not using the instrumental variable approach. (a) Comparison of alignment of the eigenbasis of the estimated graphs with the ones of the true graph. (b) Comparison of eigenvalues of estimated graphs. (c) Comparison of estimated topologies.

the connections of $N = 34$ members through 78 undirected edges. We consider the discrete model (6) with $\boldsymbol{A}$ described by a continuous-time diffusion process $\boldsymbol{A} = e^{-\tau \boldsymbol{L}_x}$. The diffusion rate (or sampling time) is fixed to $\tau = 10^{-3}$. The input signal is randomly generated from a standard normal distribution and the power of both the state and the observation noise is $\sigma^2 = 10^{-3}$. We aim to recover the structure of the underlying graph by collecting the continuous-time diffused signals in the network at discrete times. The input to the network represents exogenous stimuli applied to the nodes. In social settings, this input can be interpreted as modifications to the ratings/preferences of the users which are being diffused in the network by local

aggregations. The states represent the current values of the ratings/preference of the nodes at the different sampled times.

We consider three different approaches to estimate the underlying network topology: $i)$ a covariance-based approach, where the covariance matrix is estimated from the observables; $ii)$ the instrumental variable approach combined with the spectral template method from [15]; and $iii)$ the proposed instrumental variable approach by enforcing the dynamics of the continuous system. These results are reported in Fig. 3.

Fig. 3 (a) shows the fitting accuracy of the subspaces, while Fig. 3 (b) illustrates the fitting of the eigenvalues. In Fig. 3 (c), we show the obtained graphs where the edges with absolute
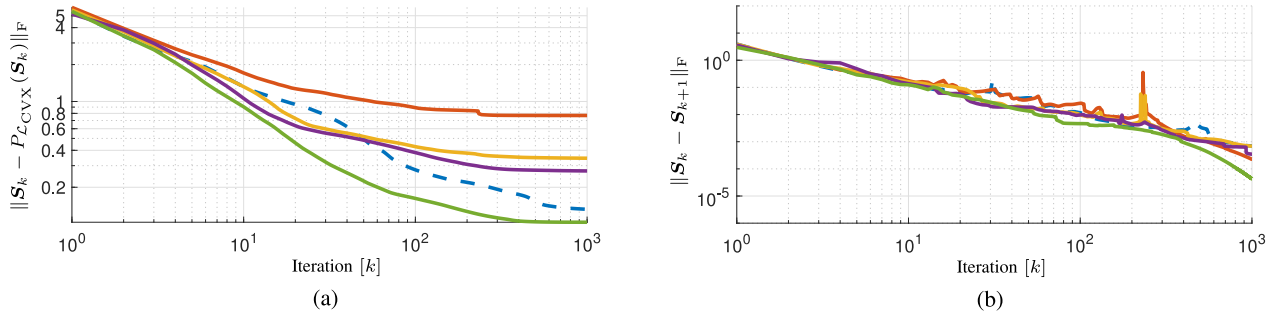
Fig. 4.   Convergence plots for the alternating projections method with $\mathcal{M}_\epsilon^m$ and $\mathcal{L}_{\mathrm{CVX}}$. (a) Error with respect to the projection, i.e., $\|\boldsymbol{S}_k - P_{\mathcal{L}_{\mathrm{CVX}}}(\boldsymbol{S}_k)\|_{\mathrm{F}}$. (b) Iterate error, i.e., $\|\boldsymbol{S}_k - \boldsymbol{S}_{k+1}\|_{\mathrm{F}}$.

weight less than 10^{-3} are omitted. We observe that the system identification flow allows a better graph reconstruction and the proposed method offer the best alignment of the eigenbasis. Further, by levering the underlying physical model of the diffusion, we can reconstruct the graph spectrum with high fidelity. We also remark that despite both the basis and the spectrum are aligned, the retrieved graph looks different from the true one. This is because of the ambiguities discussed in Section V-B. However, the obtained graph has the same eigenvalues as the original one and its basis diagonalizes the original network matrix. Notice that from the three methods, only the one leveraging the model information retrieves a connected graph after thresholding.

Finally, we remark that the task of estimating this topology based purely on a spectral decomposition is hard. This is because the combinatorial Laplacian of the Karate club graph has eigenvalues with a multiplicity larger than one. Thus, there is no unique basis for its eigendecomposition leading to difficulties in reconstructing the underlying topology.

C. Convergence of the Alternating Projections Method

We analyze here the convergence behavior of the alternating projections method (38). We present results using the sets $\mathcal{M}_\epsilon^m$ and $\mathcal{S} = \mathcal{L}_{\mathrm{CVX}}$. The latter is the convex relaxation of the combinatorial Laplacian set; see [15]. These sets are chosen to illustrate the convergence results as $\mathcal{L}_{\mathrm{CVX}}$ encompasses the problem of finding Laplacian matrices with given eigenvalues and $\mathcal{M}_\epsilon^m$ is the most general set proposed in this work. Additional results for the other sets are provided in the supplementary material. For this scenario, we select a regular graph[8] with $N = 30$ and node degree $d = 3$ and consider only half of its eigenvalues known, i.e, $m = N/2$. The AP method is analyzed for five different initial points.

These results are shown in Fig. 4. Here, each solid line represents a different starting point. The (blue) dashed line shows the convergence behavior when the starting point is the (diagonal) eigenvalue matrix. These results show two main things. First, the predicted monotone behavior of the error $\|\boldsymbol{S}_k - P_{\mathcal{S}}(\boldsymbol{S}_k)\|_{\mathrm{F}}$ holds and stagnates when a limit point is reached by the iterative sequence. Second, the error $\|\boldsymbol{S}_k - \boldsymbol{S}_{k+1}\|_{\mathrm{F}}$ converges to the desired accuracy (order 10^{-6}), although not monotonically; the

[8]See supplemental material

error convergence rate is generally linear and the starting point influences the slope. Finally, we emphasize that even when the set of known eigenvalues lies within an ϵ-ball of uncertainty, the alternating projections method convergences. The convergence is guaranteed by the compactness of the set $\mathcal{M}_\epsilon^m$.

D. Partial Observations

In this section, we consider the task of retrieving a graph that realizes a given system from partial observations. We consider two regular random graphs of $N = 14$ nodes and three edges per node. The data are generated from a continuous diffusion on the network and the input matrix is $\boldsymbol{B} = \boldsymbol{L}_u + \boldsymbol{I}$. The matrix $\boldsymbol{C}$ is a Boolean matrix that selects half of the nodes (the odd labeled nodes from arbitrary labeling). Note that none of the previous methods can be employed to retrieve the network topology since $\boldsymbol{B} \neq \boldsymbol{I}$ and the network is not fully observed. Even if the covariance matrix is estimated from sampled data, its eigenstructure does not represent the eigenstructure of the state network topology.

We first estimate the system matrices using the system identification framework and then employ the AP method initialized with a random symmetric matrix that has as eigenvalues the estimated state network eigenvalues. The constraint set in (34) is the convex relaxation of the combinatorial Laplacian set [15]. We enrich this set with the system identification constraints to enforce the feasibility of the realization. Fig. 5 reports the results after 30 iterations of the AP method.

From Fig. 5 (a), we observe that the estimated state graph does not exactly share the eigenbasis with the original one, i.e., the graph mode projections do not form a diagonal matrix. However, we could perfectly match the input graph. This behavior is further seen in the eigenvalues, where those of the input graph are matched by the estimated eigenvalues. For the state graph, a perfect eigenvalue match is possible if a final projection onto $\mathcal{M}$ is performed. These results are also reflected in the estimated topologies in Fig. 5. Perfect reconstruction of the input graph support is achieved, while the state graph presents a different arrangement in the nodes and it is not regular.

Despite the differences in the state graphs, the estimated triple $\{\hat{\boldsymbol{S}}_x, \hat{\boldsymbol{S}}_u, \boldsymbol{C}\}$ realizes (approximately) the same system as the true triple $\{\boldsymbol{S}_x, \boldsymbol{S}_u, \boldsymbol{C}\}$. This is because the product of the
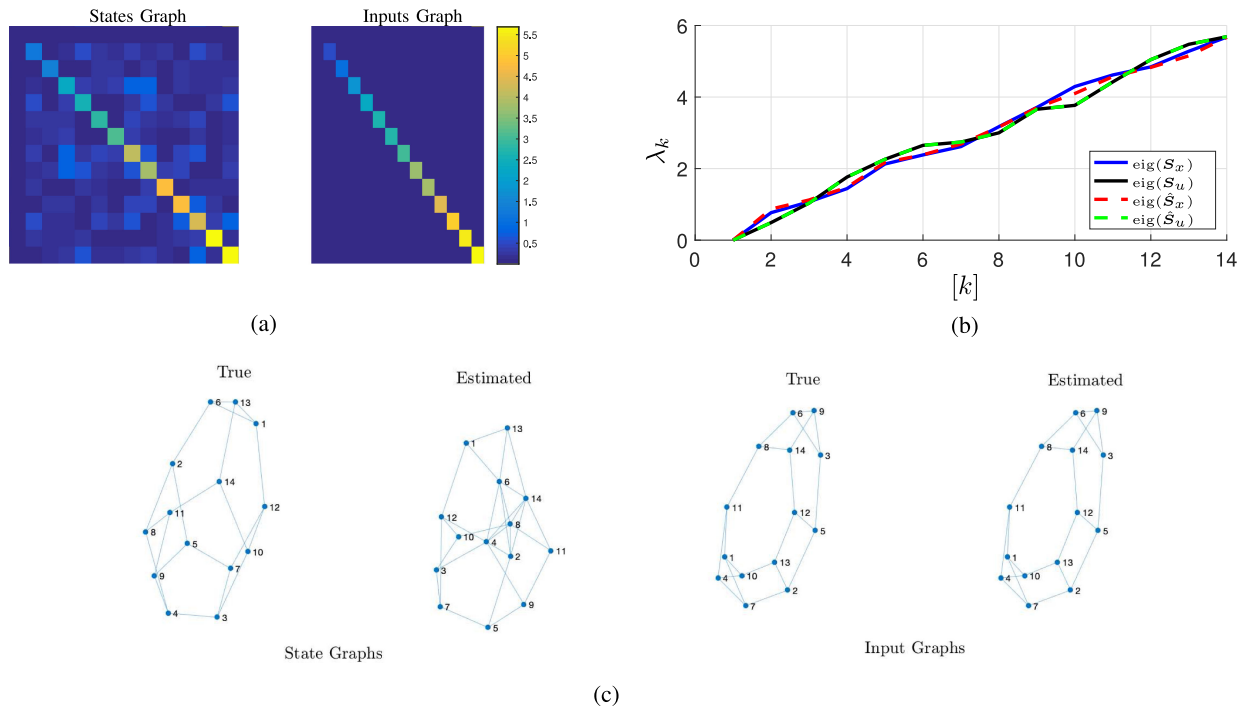
Fig. 5. Results for the reconstruction of a graph from a dynamical system using partial observations and the alternating projections method. (a) Projection results on the modes of the estimated graphs, i.e., $|\hat{Q}_*^T S_* \hat{Q}_*|$ with Q_* being the eigenvectors of the estimated graph. (b) Comparison of the estimated graph eigenvalues. (c) Reconstructed graphs.

involved system matrices is preserved, i.e., although the structure of the state graph is different, the observations can be reproduced with high confidence using the estimated system matrices. With the estimated graphs, we can predict the system output with an NRMSE fitness of $\approx 95\%$. In the supplementary material, we compare the response of the true system to an arbitrary excitation with the obtained system response that uses the estimated graphs.

VIII. CONCLUSION

This paper introduced a general framework for graph topology identification through state-space models and subspace techniques. We showed that it is possible to retrieve the matrix representation of the involved graphs from the system matrices by exploiting the geometric structure of the input and output data. In particular, we discussed the challenges of retrieving the network topology under partial observations and proposed an alternating projections method to recover a set of matrices that realizes the system. The proposed theoretical analysis is corroborated with numerical results. Future research is needed in three main directions. First, the focus should be on improving the scalability of the proposed techniques to larger graphs. Second, research is needed in employing subspace models to learn a coarser graph that drives the system dynamics in large data sets. Third, extensions of the current approach to nonlinear systems, such as the ones in chemical reaction networks [37], could bridge the gap between the GSP community and the complex network literature; see, e.g., [45], [46] by exploiting sparsity in basis functions modeling nonlinear dynamics.

REFERENCES

[1] J. A. Deri and J. M. Moura, "New York City Taxi analysis with graph signal processing," in *Proc. IEEE Conf. Sig. Inf. Process. (GLOBALSIP)*, 2016, pp. 1275–1279.

[2] O. Sporns, *Networks of the Brain*. Cambridge, MA, USA: MIT Press, 2010.

[3] F. Mittler *et al.*, "Reactive oxygen gene network of plants," *Trends Plant Sci.*, vol. 9, no. 10, pp. 490–498, 2004.

[4] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 5445–5449.

[5] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584–3598, 2018.

[6] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Trans. Sig. Inf. Proc. Over Netw.*, vol. 2, no. 4, pp. 539–554, Dec. 2016.

[7] 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 Sig. Proc. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

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

[9] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320–2333, May 2019.

[10] C. Hu, J. Sepulcre, K. A. Johnson, G. E. Fakhri, Y. M. Lu, and Q. Li, "Matched signal detection on graphs: Theory and application to brain imaging data classification," *NeuroImage*, vol. 125, pp. 587–600, 2016.

[11] S. P. Chepuri and G. Leus, "Subgraph detection using graph signals," in *Proc. 50th Asilomar Conf. Sig., Syst. Comp.*, 2016, pp. 532–534.

[12] E. Isufi, A. S. Mahabir, and G. Leus, "Blind graph topology change detection," *Signal Process. Lett.*, vol. 25, no. 5, pp. 655–659, 2018.

[13] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Artif. Intell. Statist.*, 2016, pp. 920–929.

[14] G. Mateos, S. Segarra, and A. G. Marques, "Inference of graph topology," in *Cooperative and Graph Signal Processing: Principles and Applications*, P. M. Djuric and C. Richard, Eds.), Amsterdam, The Netherlands: Elsevier, 2018.

[15] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[16] G. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *Proc. IEEE Int. Conf. Acoust. Speech Sig. Process.*, 2016, pp. 6315–6319.

[17] Y. Shen, B. Baingana, and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. Sig. Process.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[18] K.-S. Lu and A. Ortega, "A graph Laplacian matrix learning method for fast implementation of graph Fourier transform," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 1677–1681.

[19] J. Zhou and J.-a. Lu, "Topology identification of weighted complex dynamical networks," *Physica A Statist. Mech. Appl.*, vol. 386, no. 1, pp. 481–491, 2007.

[20] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Sig. Process.*, vol. 65, no. 14, pp. 3675–3687, Jul. 2017.

[21] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Network topology inference from non-stationary graph signals," in *Proc. IEEE Int. Conf. Acoust. Speech Sig. Process.*, 2017, pp. 5870–5874.

[22] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, May 2019.

[23] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[24] A. P. Dempster, "Covariance selection," *Biometrics*, pp. 157–175, 1972.

[25] S. L. Lauritzen, *Graphical Models*. Oxford, U.K.: Clarendon Press, 1996, vol. 17.

[26] S. Segarra, A. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.

[27] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[28] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[29] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 3, pp. 484–499, Sep. 2017.

[30] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[31] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *J. Mach. Learn. Res.*, vol. 21, no. 22, pp. 1–60, 2020.

[32] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and signals over graphs," in *Proc. IEEE Data Sci. Workshop*, 2018, pp. 165–169.

[33] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2263–2274, May 2019.

[34] Y. I. Medina, R. C. Sotero, E. J. C. Rodríguez, Y. A. Gómez, and L. M. García, "Studying the human brain anatomical network via diffusion-weighted mri and graph theory," *Neuroimage*, vol. 40, no. 3, pp. 1064–1076, 2008.

[35] J. A. Hertz, *Introduction to the Theory of Neural Computation*. Boca Raton, FL, USA: CRC Press, 2018.

[36] D. Angeli, "A tutorial on chemical reaction network dynamics," *Eur. J. Control*, vol. 15, no. 3-4, pp. 398–406, 2009.

[37] W. Pan, Y. Yuan, J. Gonçalves, and G. B. Stan, "Reconstruction of arbitrary biochemical reaction networks: A compressive sensing approach," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 2334–2339.

[38] L. Ljung, "Perspectives on system identification," *Annu. Rev. Control*, vol. 34, no. 1, pp. 1–12, 2010.

[39] M. Timme and J. Casadiego, "Revealing networks from dynamics: an introduction," *J. Phys. A: Math. Theor.*, vol. 47, no. 34, 2014, Art. no. 343001.

[40] W.-X. Wang, Y.-C. Lai, and C. Grebogi, "Data based identification and prediction of nonlinear and complex dynamical systems," *Phys. Reports*, vol. 644, pp. 1–76, 2016.

[41] S. G. Shandilya and M. Timme, "Inferring network topology from complex dynamics," *New J. Phys.*, vol. 13, no. 1, 2011, Art. no. 013004.

[42] D. Yu, M. Righero, and L. Kocarev, "Estimating topology of networks," *Phys. Rev. Lett.*, vol. 97, no. 18, 2006, Art. no. 188701.

[43] M. Timme, "Revealing network connectivity from response dynamics," *Phys. Rev. Lett.*, vol. 98, no. 22, 2007, Art. no. 224101.

[44] X. Han, Z. Shen, W.-X. Wang, and Z. Di, "Robust reconstruction of complex networks from sparse data," *Phys. Rev. Lett.*, vol. 114, no. 2, 2015, Art. no. 028701.

[45] W. Pan, Y. Yuan, L. Ljung, J. Gonçalves, and G.-B. Stan, "Identification of nonlinear state-space systems from heterogeneous datasets," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 2, pp. 737–747, Jun. 2018.

[46] J. Casadiego, M. Nitzan, S. Hallerberg, and M. Timme, "Model-free inference of direct network interactions from nonlinear collective dynamics," *Nature Commun.*, vol. 8, no. 2192, pp. 1–10, 2017.

[47] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, "Detecting hidden nodes in complex networks from time series," *Phys. Rev. E*, vol. 85, no. 6, 2012, Art. no. 065201.

[48] R.-Q. Su, Y.-C. Lai, X. Wang, and Y. Do, "Uncovering hidden nodes in complex networks in the presence of noise," *Scientific Reports*, vol. 4, p. 3944, 2014.

[49] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, "Finite time identification in unstable linear systems," *Automatica*, vol. 96, pp. 342–353, 2018.

[50] M. Simchowitz, H. Mania, S. Tu, M. I. Jordan, and B. Recht, "Learning without mixing: Towards a sharp analysis of linear system identification," 2018, *arXiv:1802.08334*.

[51] B. Ho and R. E. Kálmán, "Effective construction of linear state-variable models from input/output functions," *At-Automatisierungstechnik*, vol. 14, no. 1-12, pp. 545–548, 1966.

[52] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 5655–5661.

[53] T. Sarkar, A. Rakhlin, and M. A. Dahleh, "Finite-time system identification for partially observed LTI systems of unknown order," 2019, *arXiv:1902.01848*.

[54] M. Viberg, "Subspace methods in system identification," *IFAC Proc. Volumes*, vol. 27, no. 8, pp. 1–12, 1994.

[55] F. R. Chung, *Spectral Graph Theory*. New York, NY, USA: American Mathematical Soc., 1997, no. 92.

[56] M. T. Angulo, J. A. Moreno, G. Lippner, A.-L. Barabási, and Y.-Y. Liu, "Fundamental limitations of network reconstruction from temporal data," *J. Roy. Soc. Interface*, vol. 14, no. 127, 2017, Art. no. 20160966.

[57] P. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2464–2471.

[58] W. Hu, X. Gao, G. Cheung, and Z. Guo, "Feature graph learning for 3d point cloud denoising," 2019, *arXiv:1907.09138*.

[59] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Rev.*, vol. 38, no. 3, pp. 367–426, 1996.

[60] Y. De Castro, T. Espinasse, and P. Rochet, "Reconstructing undirected graphs from eigenspaces," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1679–1702, 2017.

[61] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*. Berlin, Germany: Springer, 1981, pp. 366–381.

[62] M. Viberg, "Subspace-based methods for the identification of linear time-invariant systems," *Automatica*, vol. 31, no. 12, pp. 1835–1851, 1995.

[63] L. Ljung, "Prediction error estimation methods," *Circuits, Syst. Signal Process.*, vol. 21, no. 1, pp. 11–21, 2002.

[64] S.-Y. Chung, Y.-S. Chung, and J.-H. Kim, "Diffusion and elastic equations on networks," *Publications Res. Inst. Math. Sci.*, vol. 43, no. 3, pp. 699–726, 2007.

[65] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[66] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[67] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Siam, 2008, vol. 104.

[68] L. Lovász *et al.*, "Random walks on graphs: A survey," *Combinatorics, Paul Erdos Eighty*, vol. 2, no. 1, pp. 1–46, 1993.

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

[70] K. Ogata and Y. Yang, *Modern Control Engineering*. NJ, USA: Prentice hall India, 2002, vol. 4.

[71] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. New York, NY, USA: Cambridge Univ. Press, 2007.

[72] M. Verahegen and P. Dewilde, "Subspace model identification. part i: The output-error state-space model identification class of algorithm," *Int. J. Control*, vol. 56, no. 3, pp. 1187–1210, 1992.

[73] P. V. Overschee and B. De Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.

[74] P. Van Overschee and B. De Moor, *Subspace Identification for linear systems: Theory Implementation Applications*. Berlin, Germany: Springer Science & Business Media, 2012.

[75] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 3, pp. 481–496, Sep. 2018.

[76] M. Coutino, S. Chepuri, and G. Leus, "Sparsest network support estimation: A submodular approach," in *Proc. IEEE Data Sci. Workshop*, 2018, pp. 200–204.

[77] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from F signals: Graph system and diffusion Kernel identification," *IEEE Trans. Signal and Inf. Process. Over Netw.*, vol. 5, no. 2, pp. 360–374, Jun. 2019.

[78] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM Sigmod Rec.*, vol. 42, no. 2, pp. 17–28, 2013.

[79] M. Chu, G. Golub, and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*. New York, NY, USA: Oxford Univ. Press, 2005, vol. 13.

[80] K. Devlin, *Sets, Functions, and Logic: An Introduction to Abstract Mathematics*. London, U.K.: Chapman and Hall/CRC, 2003.

[81] A. E. Brouwer and W. H. Haemers, *Spectra of Graphs*. Berlin, Germany: Springer Science & Business Media, 2011.

[82] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Hoboken, NJ, USA: JHU Press, 2012, vol. 3.

[83] M. T. Chu and K. R. Driessel, "The projected gradient method for least squares matrix approximations with spectral constraints," *SIAM J. Numerical Anal.*, vol. 27, no. 4, pp. 1050–1060, 1990.

[84] R. Orsi, "Numerical methods for solving inverse eigenvalue problems for nonnegative matrices," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 1, pp. 190–212, 2006.

[85] A. S. Lewis, D. R. Luke, and J. Malick, "Local linear convergence for alternating and averaged nonconvex projections," *Foundations Comput. Math.*, vol. 9, no. 4, pp. 485–513, 2009.

[86] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, 1977.

Mario Coutino (Student Member, IEEE) received the M.Sc. degree (*cum laude*) in electrical engineering in July 2016 from the Delft University of Technology, Delft, The Netherlands, where he has been currently working toward the Ph.D. degree in signal processing since October 2016. He has held positions at Thales Nederlands, during summer 2015, and Bang & Olufsen, during 2015–2016. He received a Best Student Paper Award for his publication at the CAMSAP 2017 conference in Curacao and was a Visiting Researcher with RIKEN AIP and the Digital Technological Center, University of Minnesota, during 2018 and 2019, respectively. His research interests include signal processing on networks, submodular and convex optimization, and numerical lineal algebra.

Elvin Isufi (Member, IEEE) was born in Vlore, Albania, in 1989. He received the Ph.D. degree from Delft University of Technology, Delft, The Netherlands, in 2019, and the M.Sc. degree (*cum laude*) from the University of Perugia, Perugia, Italy, in 2014. He is currently an Assistant Professor with the Multimedia Computing Group, Delft University of Technology. Prior to that, he was a Postdoctoral Researcher with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. His research interests include intersection of signal processing, mathematical modeling, machine learning, and network theory.

Takanori Maehara received the M.Sc. and Ph.D. degrees from The University of Tokyo, Tokyo, Japan, in March 2009 and September 2012, respectively. He is currently the Unit Leader of the Discrete Optimization Group, RIKEN Center of Advanced Intelligence Project, Japan. His research interests include the intersection of graph theory, mathematical modeling, machine learning, and optimization.

Geert Leus (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the KU Leuven, Leuven, Belgium, in June 1996 and May 2000, respectively. He is currently an "Antoni van Leeuwenhoek" Full Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His research interests are in the broad area of signal processing, with a specific focus on wireless communications, array processing, sensor networks, and graph signal processing. He received a 2002 IEEE Signal Processing Society Young Author Best Paper Award and a 2005 IEEE Signal Processing Society Best Paper Award. He is a fellow of EURASIP. He was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society, the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee, a member of the IEEE Sensor Array and Multichannel Technical Committee, and the Editor-in-Chief for the *EURASIP Journal on Advances in Signal Processing*. He was also on the Editorial Boards for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE SIGNAL PROCESSING LETTERS, and the *EURASIP Journal on Advances in Signal Processing*. He is currently the Vice-Chair of the EURASIP Special Area Team on Signal Processing for Multisensor Systems, an Associate Editor for *Foundations and Trends in Signal Processing*, and the Editor-in-Chief for *EURASIP Signal Processing*.