STATE-SPACE BASED NETWORK TOPOLOGY IDENTIFICATION

Mario Coutino[†], *Elvin Isufi*[†], *Takanori Maehara*[‡], *Geert Leus*[†]

Delft University of Technology, Delft, The Netherlands[†]
AIP RIKEN, Tokyo, Japan[‡]

ABSTRACT

In this work, we explore the state-space formulation of network processes to recover the underlying structure of the network (local connections). To do so, we employ subspace techniques borrowed from system identification literature and extend them to the network topology inference problem. This approach provides a unified view of the traditional network control theory and signal processing on networks. In addition, it provides theoretical guarantees for the recovery of the topological structure of a deterministic linear dynamical system from input-output observations even though the input and state evolution networks can be different.

Index Terms— state-space models, topology identification, graph signal processing, signal processing over networks

1. INTRODUCTION

In recent years, major efforts have been focused on extend traditional tools from signal processing to cases where the acquired data is not defined over typical domains such as time or space but over a network (graph) [1, 2]. The main reason for the increase of research in this area is due to the fact that network-supported signals can model complex processes. For example, by means of signals supported on graphs we are able to model transportation networks [3], brain activity [4], and epidemic diffusions or gene regulatory networks [5], to name a few.

As modern signal processing techniques take into account the network structure to provide signal estimators [6–8], filters [9–12], or detectors [13–15], appropriate knowledge of the interconnections of the network is required. In many instances, the knowledge of the network structure is given and can be used to enhance traditional signal processing algorithms. However, in other cases, the network information is unknown and needs to be estimated. As the importance of studying such structures in the data has been noticed, retrieving the topology of the network has become a topic of extensive research [16–24].

Despite the extensive research done so far (for a comprehensive review the reader is referred to [2, 17] and references therein), most of the approaches do not lever a physical model beyond the one induced by the so-called *graph filters* [18, 25] drawn from graph signal processing (GSP) [10, 26, 27]. Among the ones that propose a different interaction model e.g., [20, 24], none of them considers the network data (a.k.a. graph signals) as states of an underlying process nor considers the that the input and the state may evolve on different underlying networks. However, different physical systems of practical interest can be defined through a state-space formulation with (probably) known functions, i.e., brain activity diffusion, finite element models, circuit/flow systems. For these processes, a more general approach to find the underlying connections is required. In this work, we therefore focus on the general problem of *retrieving the underlying network structure*, from input-output signals, of a process that can be modeled through a *deterministic linear dynamical system* whose system matrices depend on the interconnection of the network.

2. STATE-SPACE MODELS FOR NETWORK PROCESSES

Let us consider a tuple of graphs $\mathcal{G}_1 = \{\mathcal{V}, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}, \mathcal{E}_2\}$ to represent two networks, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ and $\mathcal{E}_i \subseteq \mathcal{V}\times\mathcal{V}$ for $i \in \{1, 2\}$ denote their vertex and edge sets, respectively. Further, let $\mathcal{P}$ be a process over $\{\mathcal{G}_1, \mathcal{G}_2\}$ that describes the evolution through time of a signal (the state) $\boldsymbol{x}(t)$ defined over $\mathcal{G}_1$ coupled with another signal (the input signal) defined over $\mathcal{G}_2$. Such process can be described through the linear dynamical system

$$\boldsymbol{x}(k+1) = f_1(\boldsymbol{S}_1)\boldsymbol{x}(k) + f_2(\boldsymbol{S}_2)\boldsymbol{u}(k) \in \mathbb{R}^n, \quad (1a)$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \in \mathbb{R}^l, \quad (1b)$$

where $\boldsymbol{S}_i$, $i \in \{1, 2\}$, is the matrix representation of the graph $\mathcal{G}_i$, i.e., the shift operator in the GSP terminology, $\boldsymbol{C} \in \mathbb{R}^{l\times n}$, and $\boldsymbol{D} \in \mathbb{R}^{l\times n}$ are the observation matrices and $f_i : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is a matrix function defined via the Cauchy integral by [28]

$$f_i(\boldsymbol{S}) := \frac{1}{2\pi i}\int_{\Gamma_{f_i}} f_i^{\mathrm{s}}(z)R(z, \boldsymbol{S})dz, \quad (2)$$

with f_i^{s} being the scalar version of f_i which is assumed analytic on and over the contour Γ_{f_i}. Here, $R(z, \boldsymbol{S})$ is the resolvent of $\boldsymbol{S}$ given by

$$R(z, \boldsymbol{S}) := (\boldsymbol{S} - z\boldsymbol{I})^{-1}. \quad (3)$$

Model (1) is expressed in terms of its *state-space* representation and captures the relation between the input, the output, and the state through a first-order difference equation [29]. It connects the output (observables), $\boldsymbol{y}(k)$, to a set of variables (states), $\boldsymbol{x}(k)$, which vary over time and depend on their previous value and on external inputs (excitations), $\boldsymbol{u}(k)$.

After observing model (1), a natural question that arises is the following: *assuming that the observation matrix $\boldsymbol{C}$ and the relation between $\mathcal{P}$ and $\{\mathcal{G}_i, \mathcal{G}_j\}$ are known, how can we retrieve $\{\boldsymbol{S}_i, \boldsymbol{S}_j\}$, i.e., the network structures, from a number of samples of the input signal $\boldsymbol{u}(k)$ and the output signal $\boldsymbol{y}(k)$?*

In this work, we aim to answer this question by employing techniques commonly used in control theory which rely on results for Hankel matrices and linear algebra. In particular, we employ subspace techniques which do not require any parametrization of the model, hence the problem of performing nonlinear optimization, as in the prediction-error methods [30], is avoided.

This research is supported in part by the ASPIRE project (project 14926 within the STW OTP programme), financed by the Netherlands Organization for Scientific Research (NWO). Mario Coutino is partially supported by CONACYT and AIP RIKEN.

3. IDENTIFIABILITY CONDITIONS FOR LTI SYSTEMS

For the sake of simplifying notation, from this point on, we omit the dependency on $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ of the system matrices in (1) and refer to the matrices $f_i(\boldsymbol{S}_i)$ and $f_j(\boldsymbol{S}_j)$ as $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

Prior to introducing the methods for network topology identification, we digress the identifiability conditions of LTI systems. In this section, we briefly recap the requirements on the system matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ for applying subspace techniques for estimating them.

The main requirement for system identification is the *minimality* condition of system (1). This property is intrinsically related to two well-known properties of dynamical systems: *reachability* and *observability*. The first property denotes the ability of the input, $\boldsymbol{u}(k)$, to steer the system state to the zero state within a finite time interval. While the second denotes the ability to observe the time evolution of the states through the evolution of the output; that is, it answers the question of the uniqueness of the relation between state and the output.

Before stating these notions mathematically, let us introduce the following two matrices [29]

- **Controllability Matrix:** $C_s \triangleq [\boldsymbol{B}\ \boldsymbol{AB}\ \cdots\ \boldsymbol{A}^{s-1}\boldsymbol{B}]$,
- **Observability Matrix:** $O_s \triangleq [\boldsymbol{C}^T\ \cdots\ (\boldsymbol{A}^{s-1})^T\boldsymbol{C}^T]^T$.

Based on these matrices, the following two lemmas state the concepts of reachability and observability in a more formal way.

Lemma 1. (Reachability) *The LTI system* (1) *is reachable if and only if* $\mathrm{rank}(C_n) = n$.

Lemma 2. (Observability) *The LTI system* (1) *is observable if and only if* $\mathrm{rank}(O_n) = n$.

By using these results we can now formally state the definition of *minimality* of a system.

Definition 1. (Minimality) *The LTI system* (1) *is minimal if and only if it is both reachable and observable. Furthermore, the dimension of the state vector $\boldsymbol{x}(k)$ of the minimal system defines the* order *of the LTI system.*

As the system identification framework only guarantees recovery of a *minimal* system, from this point on, we only consider problem instances where the system of interest is minimal. Note that this is not a restrictive assumption, as even when we retrieve a minimal system of order $p < n$, this can be interpreted as a system on the nodes of a hypergraph, i.e., clusters of nodes that drive the general behavior of the process over the network.

4. SUBSPACE NETWORK IDENTIFICATION

In this section, we introduce a general framework for estimating the topology of the networks, i.e., the associated matrices $\{\boldsymbol{S}_1, \boldsymbol{S}_2\}$, from input-output relations. To do so, we first provide the methods for retrieving the system matrices in (1). Then, we state the required conditions and propose different methods for estimating the graph matrices $\{\boldsymbol{S}_1, \boldsymbol{S}_2\}$ from the obtained system matrices.

4.1. State-Space Identification

It is not hard to show that the state of the system (1) with initial state $\boldsymbol{x}(0)$ at time instant k is given by

$$\boldsymbol{x}(k) = \boldsymbol{A}^k\boldsymbol{x}(0) + \sum_{i=0}^{k-1}\boldsymbol{A}^{k-i-1}\boldsymbol{B}\boldsymbol{u}(i). \tag{4}$$

Observing the expression relating the states, the input and the output in (1), we can specify the following relationship between the batch input $\{\boldsymbol{u}(k)\}_{k=0}^{s-1}$ and the batch output $\{\boldsymbol{y}(k)\}_{k=0}^{s-1}$

$$\begin{bmatrix} \boldsymbol{y}(0) \\ \vdots \\ \boldsymbol{y}(s-1) \end{bmatrix} = O_s\boldsymbol{x}(0) + \mathcal{T}_s \begin{bmatrix} \boldsymbol{u}(0) \\ \vdots \\ \boldsymbol{u}(s-1) \end{bmatrix}, \tag{5}$$

where

$$\mathcal{T}_s \triangleq \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CB} & \boldsymbol{D} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \boldsymbol{D} & \cdots & \boldsymbol{0} \\ \dots & & & \ddots & \ddots \\ \boldsymbol{CA}^{s-2}\boldsymbol{B} & \boldsymbol{CA}^{s-3}\boldsymbol{B} & \dots & \boldsymbol{CB} & \boldsymbol{D} \end{bmatrix}, \tag{6}$$

and s is the size of the batch that must be larger than the number of states (assuming the number of nodes is the number of states this implies $s > n$).

Given that the underlying system is time-invariant (i.e., the graph does not change in time), the following relation holds [31]

$$\boldsymbol{Y}_m = O_s\boldsymbol{X}_m + \mathcal{T}_s\boldsymbol{U}_m \tag{7}$$

where $\boldsymbol{X}_m \triangleq [\boldsymbol{x}(0), \boldsymbol{x}(1), \cdots, \boldsymbol{x}(m-1)]$, $\boldsymbol{y}_{i,s} \triangleq [\boldsymbol{y}(i)^T, \boldsymbol{y}(i+1)^T, \cdots, \boldsymbol{y}(i+s-1)^T]^T$, $\boldsymbol{Y}_m \triangleq [\boldsymbol{y}_{0,s}, \boldsymbol{y}_{1,s}, \cdots, \boldsymbol{y}_{m-1,s}]$, $\boldsymbol{u}_{i,s} \triangleq [\boldsymbol{u}(i)^T, \boldsymbol{u}(i+1)^T, \cdots, \boldsymbol{u}(i+s-1)^T]^T$ and $\boldsymbol{U}_m \triangleq [\boldsymbol{u}_{0,s}, \boldsymbol{u}_{1,s}, \cdots, \boldsymbol{u}_{m-1,s}]$ with $m > s$.

Throughout this work, we assume that that $\boldsymbol{C}$ has rank equal to n. Despite that this assumption seems restrictive, we consider it to simplify the exposition of the approach. Dealing with dynamical models whose output dynamics satisfy $l < n$ is not trivial. As it will become evident, disambiguation of the system matrices requires extra information when $\boldsymbol{C}$ is wide or singular. Therefore, this is left for immediate future work.

To identify the system matrices from (1), we first make use of the following lemma.

Lemma 3. *(Verhaegen and Dewilde [32]) Given the following RQ factorization*

$$\begin{bmatrix} \boldsymbol{U}_m \\ \boldsymbol{Y}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_1 \\ \boldsymbol{Q}_2 \\ \boldsymbol{Q}_3 \end{bmatrix}, \tag{8}$$

for appropriately sized matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$, the following relationship holds for the input-output data matrices

$$\boldsymbol{Y}_m\boldsymbol{\Pi}_m^{\perp} = \boldsymbol{R}_{22}\boldsymbol{Q}_2, \ \ \textit{where}\ \ \boldsymbol{U}_m\boldsymbol{\Pi}_m^{\perp} = \boldsymbol{0}. \tag{9}$$

Using Lemma 3, it can be shown that

$$\mathrm{range}(\boldsymbol{Y}_m\boldsymbol{\Pi}_m^{\perp}) = \mathrm{range}(O_s) = \mathrm{range}(\boldsymbol{R}_{22}\boldsymbol{Q}_2). \tag{10}$$

Therefore, from the singular value decomposition (SVD) of $\boldsymbol{R}_{22}$, i.e., $\boldsymbol{R}_{22} = \boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T$, we can obtain the transition matrix $\boldsymbol{A}$ (up to a similarity transform) as follows. First, from

$$\boldsymbol{U}_R = O_s\boldsymbol{T} = \begin{bmatrix} \boldsymbol{CT} \\ \boldsymbol{CT}(\boldsymbol{T}^{-1}\boldsymbol{AT}) \\ \vdots \\ \boldsymbol{CT}(\boldsymbol{T}^{-1}\boldsymbol{AT})^{s-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_T \\ \boldsymbol{C}_T\boldsymbol{A}_T \\ \vdots \\ \boldsymbol{C}_T\boldsymbol{A}_T^{s-1} \end{bmatrix}, \tag{11}$$

where we have defined $\boldsymbol{A_T} := \boldsymbol{T}^{-1}\boldsymbol{AT}$ and $\boldsymbol{C_T} := \boldsymbol{CT}$ for an unknown similarity transformation matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$, we can compute an estimate $\hat{\boldsymbol{A}}_T$ of $\boldsymbol{A_T}$ by solving the overdetermined system

$$\boldsymbol{U}_{R,l}\boldsymbol{A}_T = \boldsymbol{U}_{R,r}, \tag{12}$$

which exploits the shift-invariance of the system. Here, we have defined the matrices

$$\boldsymbol{U}_{R,r} \quad := \quad \boldsymbol{U}_R(l+1:sl,:), \tag{13}$$

$$\boldsymbol{U}_{R,l} \quad := \quad \boldsymbol{U}_R(1:(s-1)l,:), \tag{14}$$

and abused the MATLAB notation to denote the rows and columns that are considered for building system (12). From (11), we can observe that an estimate $\hat{\boldsymbol{C}}_T$ of $\boldsymbol{C}_T$ can be obtained by selecting the first l rows of $\boldsymbol{U}_R$.

Since of $\boldsymbol{C}$ is full rank, we can estimate the similarity transform $\boldsymbol{T}$ from $\boldsymbol{C}_T$. Therefore, the estimate $\hat{\boldsymbol{A}}$, for $\boldsymbol{A}$, can be obtained as

$$\hat{\boldsymbol{A}} = \boldsymbol{C}^{-1}\hat{\boldsymbol{C}}_T\hat{\boldsymbol{A}}_T(\hat{\boldsymbol{C}}_T)^{-1}\boldsymbol{C}. \tag{15}$$

While a similar approach using the matrices $\boldsymbol{R}_{21}$ and $\boldsymbol{R}_{11}$ can be performed for retrieving a transformed $\boldsymbol{B}$, i.e., $\boldsymbol{B}_T = \boldsymbol{T}^{-1}\boldsymbol{B}$, [32], here we compute it, together with the initial state $\boldsymbol{x}_T(0) = \boldsymbol{T}^{-1}\boldsymbol{x}(0)$, by solving a least squares problem. This is done to keep the exposition of the approach conceptually simple, as the usage of the information in $\boldsymbol{R}_{21}$ and $\boldsymbol{R}_{11}$ requires the introduction of another (more involved) shift-invariant structure.

To do so, first, observe that for given matrices $\boldsymbol{A}_T$ and $\boldsymbol{C}_T$ the output can be expressed linearly in the matrices $\boldsymbol{B}_T$ and $\boldsymbol{D}$ as

$$\begin{aligned}&\boldsymbol{y}(k) = \boldsymbol{C}_T\boldsymbol{A}_T^k\boldsymbol{x}_T(0) + \\ &\left(\sum_{q=0}^{k-1}\boldsymbol{u}(q)^T \otimes \boldsymbol{C}_T\boldsymbol{A}_T^{k-q-1}\right)\mathrm{vec}(\boldsymbol{B}_T) + (\boldsymbol{u}(k)^T \otimes \boldsymbol{I}_l)\mathrm{vec}(\boldsymbol{D}),\end{aligned} \tag{16}$$

where $\boldsymbol{I}_l$ is the $l \times l$-identity matrix. From here, by defining $\boldsymbol{\theta} = [\boldsymbol{x}_T(0)^T\ \mathrm{vec}(\boldsymbol{B}_T)^T\ \mathrm{vec}(\boldsymbol{D})^T]^T$, we can find the system matrices by solving

$$\min_{\boldsymbol{\theta}} \tfrac{1}{m}\textstyle\sum_{k=0}^{m-1}\|\boldsymbol{y}(k) - \boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2, \tag{17}$$

where $\boldsymbol{\Psi} \triangleq [\boldsymbol{C}_T\boldsymbol{A}_T^k,\ (\sum_{q=0}^{k-1}\boldsymbol{u}(q)^T \otimes \boldsymbol{C}_T\boldsymbol{A}_T^{k-q-1}),\ (\boldsymbol{u}(k)^T \otimes \boldsymbol{I}_l)]$.

After the estimates $\hat{\boldsymbol{B}}_T$ and $\hat{\boldsymbol{D}}$ are obtained, we can solve for the original matrices by appropriately multiplying them with the estimate of the similarity transform as we did to retrieve $\boldsymbol{A}$ [cf. (15)].

4.2. Network Identification

At this point, the system matrices have been obtained. Now, we consider different scenarios for estimating the topology of the underlying networks.

Known Scalar Mappings $\{f_1^{\mathrm{s}}, f_2^{\mathrm{s}}\}$. In this case, we first obtain the eigenvalues of the graph matrices by applying the inverse mappings $(f_1^{\mathrm{s}})^{-1}$ and $(f_2^{\mathrm{s}})^{-1}$ to the spectra of the respective matrices. Therefore, for guaranteeing a unique set of eigenvalues for the graph matrices, the functions $\{f_1^{\mathrm{s}}, f_2^{\mathrm{s}}\}$ should be bijective, i.e., a one-to-one mapping, on an appropriate domain. For instance, for $\boldsymbol{S}_i$ being the normalized Laplacian, the mappings should be bijective in the interval $[0,2]$ as the spectrum of the normalized Laplacian lies there.

When the inverse mappings cannot be found analytically (e.g., due to computational reasons), the problem of finding the eigenvalues of the graph matrices boils down to a series of *root finding* problems. That is, consider $[\boldsymbol{\omega}_i]_k$ as the kth eigenvalue of the matrix $\boldsymbol{M}_i$, where $\boldsymbol{M}_1 := \hat{\boldsymbol{A}}$ and $\boldsymbol{M}_2 := \hat{\boldsymbol{B}}$, and f_i^s is the known scalar mapping. Then, the estimation of the eigenvalue vector $\boldsymbol{\lambda}_i$ for each of the matrices can be formulated as

$$[\hat{\boldsymbol{\lambda}}_i]_k = \underset{[\boldsymbol{\lambda}_i]_k \in \mathbb{R}^+}{\arg\min} \quad \|f_i^{\mathrm{s}}([\boldsymbol{\lambda}_i]_k) - [\boldsymbol{\omega}_i]_k\|_2^2, \tag{18}$$

for $i \in \{1,2\}$. Fortunately, there exist efficient algorithms to obtain roots with a high accuracy even for non-linear functions [34]. In addition, note that even when only $\boldsymbol{A}_T$ is known, we can still retrieve the eigenvalues of $\boldsymbol{S}_i$ as this matrix is *similar* to $\boldsymbol{A}$, i.e., $\boldsymbol{A}_T = \boldsymbol{T}^{-1}\boldsymbol{AT}$. As by definition $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ are matrix functions of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ [cf. (2)], respectively, we can use the eigenbasis from these matrices to reconstruct the graph matrices as

$$\hat{\boldsymbol{S}}_i = \hat{\boldsymbol{U}}_i\mathrm{diag}(\hat{\boldsymbol{\lambda}}_i)\hat{\boldsymbol{U}}_i^{-1}, \text{ for } i \in \{1,2\}, \tag{19}$$

with $\hat{\boldsymbol{U}}_1 = \mathrm{eigvecs}(\hat{\boldsymbol{A}})$ and $\hat{\boldsymbol{U}}_2 = \mathrm{eigvecs}(\hat{\boldsymbol{B}})$.

Unknown Scalar Mappings. When the scalar functions $\{f_1^{\mathrm{s}}, f_2^{\mathrm{s}}\}$ are unknown, we can opt to retrieve the *sparsest* graphs that are able to generate the estimated matrices, i.e.,

$$\begin{aligned}\hat{\boldsymbol{S}}_i = \quad &\underset{\boldsymbol{\omega}_i \in \mathbb{R}^n}{\mathrm{argmin}} && \|\boldsymbol{S}_i\|_0 \\ &\text{subject to} && \boldsymbol{S}_i = \hat{\boldsymbol{U}}_i\mathrm{diag}(\boldsymbol{\omega}_i)\hat{\boldsymbol{U}}_i^{-1}, \ \ \boldsymbol{S} \in \mathcal{S},\end{aligned} \tag{20}$$

where $\mathrm{diag}(\cdot)$ denotes a diagonal matrix with its argument on the main diagonal and $\mathcal{S}$ is the set of desired graph matrices, e.g., adjacency matrices, combinatorial Laplacian matrices, etc. To do so, we can employ methods existing in the GSP literature that, given the graph matrix eigenbasis, retrieve a sparse matrix representation of the graph [18,35].

One-Shot State Graph Estimation. In alternative to the previous two cases, we can estimate the network topology related to the states by avoiding the computation of $\boldsymbol{A}_T$ explicitly. That is, after obtaining an estimate of $\boldsymbol{C}_T$, and hence $\boldsymbol{T}$, we can notice that system (12) can be modified to include the graph matrix, i.e.,

$$\boldsymbol{U}_{R,l}\boldsymbol{T}^{-1}\boldsymbol{S}_i\boldsymbol{A} = \boldsymbol{U}_{R,r}\boldsymbol{T}^{-1}\boldsymbol{S}_i, \tag{21}$$

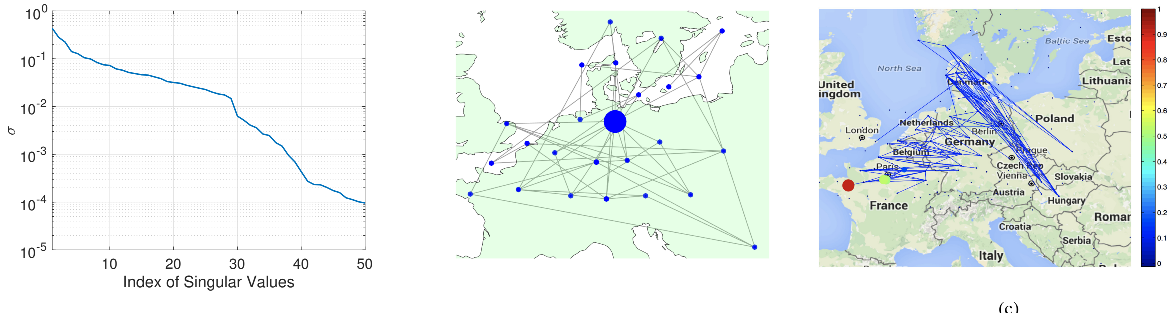
where $\boldsymbol{U}_{R,l}$ and $\boldsymbol{U}_{R,r}$ are the left and right matrices associated with $\boldsymbol{U}_R$ in (12). Notice that in (21), we not only exploit the shift invariance in the $\boldsymbol{U}_R$ matrix but also the fact that $\boldsymbol{S}_i$ and $\boldsymbol{A}$ commute. We can check that this relation holds by recalling that [cf. (11)]

$$\boldsymbol{U}_{R,l}\boldsymbol{T}^{-1}\boldsymbol{S}_i\boldsymbol{A} = \begin{bmatrix}\boldsymbol{CS}_i\boldsymbol{A} \\ \boldsymbol{CAS}_i\boldsymbol{A} \\ \vdots \\ \boldsymbol{CA}^{s-2}\boldsymbol{S}_i\boldsymbol{A}\end{bmatrix} = \begin{bmatrix}\boldsymbol{CAS}_i \\ \boldsymbol{CA}^2\boldsymbol{S}_i \\ \vdots \\ \boldsymbol{CA}^{s-1}\boldsymbol{S}_i\end{bmatrix} = \boldsymbol{U}_{R,r}\boldsymbol{T}^{-1}\boldsymbol{S}_i. \tag{22}$$

As a result, we can pose the following optimization problem

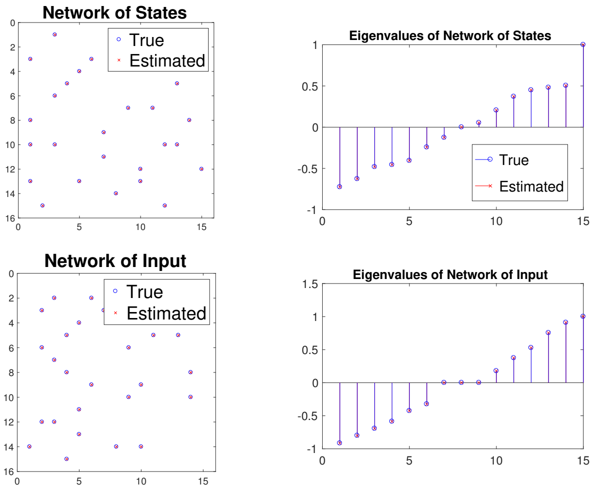
$$\min_{\boldsymbol{S}_i \in \mathcal{S}, \boldsymbol{M} \in \mathcal{M}} \quad \|\boldsymbol{U}_{R,l}\boldsymbol{T}^{-1}\boldsymbol{M} - \boldsymbol{U}_{R,r}\boldsymbol{T}^{-1}\boldsymbol{S}_i\|_{\mathrm{F}}^2 + \mu\|\boldsymbol{S}_i\|_1 \ , \tag{23}$$

where we have defined $\boldsymbol{M} := \boldsymbol{S}_i\boldsymbol{A}$ to convexify the problem. Here, μ is a regularization parameter controlling the sparsity of $\boldsymbol{S}_i$ and the optimization is carried out over the set of desired graph matrices, $\mathcal{S}$, (as in (20)) and $\mathcal{M}$ is a convex set of matrices meeting conditions derived by the matrix representation of the graph, e.g., if $\boldsymbol{S}_i$ is restricted to a combinatorial Laplacian then $\boldsymbol{1}^T\boldsymbol{M} = \boldsymbol{0}^T$ must hold.



(a) (b) (c)

Fig. 2: (a) First 50 singular values (σ) of $\frac{1}{m}\boldsymbol{Y}_m$ for $m = 3 \times 10^4$ and $s = 169$. Here, the original dataset has been interpolated and filter to increase the length of the recordings. The first knee, i.e., the drop in singular values (SV), occurs at the 5th SV, the second drop happens around the 20th SV. (b) Learned graph from dataset using the proposed one-shot state graph estimator. (c) Learned graph in [33] using the ETEX dataset.



(a) (b)

Fig. 1: Comparison of the true (blue circles) and estimated (red crosses) graphs for both the states and input. (a) Adjacency matrices for the states graph (top) and the input signal graph (bottom). (b) Eigenvalues for both states and input graph.

Alternatively, we could solve for $\boldsymbol{S}_i$ and $\boldsymbol{A}$ by means of alternative minimization [36].

Although in principle this approach requires knowledge of $\boldsymbol{T}$, in many instances it is possible to find a graph matrix associated with the transformed system, i.e., a graph associated with the system $\{\boldsymbol{A}_T, \boldsymbol{B}_T, \boldsymbol{C}_T, \boldsymbol{D}\}$, as the shift invariance property is oblivious to this ambiguity.

5. NUMERICAL EXAMPLES

To illustrate the performance of the proposed framework, we carry out a pair of experiments using synthetic and real data.

Synthetic Example. For this example, we consider a simple system where $\boldsymbol{S}_i \neq \boldsymbol{S}_j$ with $n = 15$ nodes, f_1^{s} is a scaled diffusion map (i.e., $f_1^{\mathrm{s}}(x) = \alpha_i e^{-x\tau_i}$, also known as heat kernel), and f_2^{s} is the identity map, i.e., $f_2^{\mathrm{s}}(x) = x$. Here, it is assumed that all states are measured, i.e., $\boldsymbol{C} = \boldsymbol{I}$, and that there is a direct feedback from the input to the observations, i.e., $\boldsymbol{D} = \boldsymbol{I}$. As input, we considered a random piece-wise constant (during the sampling period) binary bipolar signal with 300 samples each. The reconstruction of the topology using the proposed framework is shown in Fig. 1. In Fig. 1a, the true and reconstructed adjacency matrices for the states and input are shown. As expected, when the data follows a practical model, the reconstruction of the matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is guaranteed to be exact. Here, since we have considered simple scalar mappings, we only perform root finding to retrieve the eigenvalues of the graph matrices. The eigenvalues comparison for both graphs is shown in Fig. 1b.

ETEX dataset. We now consider data from the European Tracer Experiment (ETEX) [37]. In this experiment a tracer was released into the atmospheric system and its evolution was sampled and stored from multiple stations in time. As it is unlikely that such kind of process has as many states as stations, we cluster the 168 measuring stations in 25 geographical regions and aggregate its measurements as a preprocessing step. This preprocessing is sustained by looking at the singular values of $\frac{1}{m}\boldsymbol{Y}_m$ in Fig. 1a. In this figure, it is observed that most of the dynamics can be described with a system of order 5, i.e., first knee in the plot. Here, we selected 25 nodes as a trade off between complexity and graph interpretability (second knee). As the propagation of the tracer is considered to be a pure diffusion in an *autonomous* system, i.e., matrices $\boldsymbol{B}$ and $\boldsymbol{D}$ equal zero, we employ the proposed one-shot state graph estimation method [cf. (23)] to retrieve the underlying network structure. In this case, it is also assumed that the observations are the states of the system, i.e., $\boldsymbol{C} = \boldsymbol{I}$. The estimated graph is shown in Fig. 2b. Here, the size of the circle representing a vertex is proportional to the degree of the node. From the estimated graph, we can observe that the region of Berlin presents the highest degree which is consistent with the concentration results in [33]. Further, the strong connectivity along the France–Germany region correlates with the spreading pattern of the agent. Despite that this graph has fewer nodes than the one obtained in [33] (see Fig. 2c), the estimated graph presents a better visual interpretability and exhibits a similar edge behaviour.

6. CONCLUSION

In this paper, we have introduced a general framework for graph topology learning using state space-models and subspace techniques. Specifically, we have shown that it is possible to retrieve the matrix representation of the involved graphs from the system matrices by different means. In the particular case of the graph related to the states, we presented a one-shot method for topology identification that does not require the explicit computation of the system matrix. Numerical experiments for both synthetic and real data have demonstrated the applicability of the proposed method and its capabilities to recover the topology of the underlying graph from data.

7. REFERENCES

[1] Eric D Kolaczyk, *Statistical analysis of network data: methods and models*, Springer Science & Business Media, 2009.

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

[3] Joya A Deri and Jose MF Moura, "New york city taxi analysis with graph signal processing," in *Proc. of the IEEE Conf. on Sig. and Inf. Process. (GLOBALSIP)*, 2016, pp. 1275–1279.

[4] Olaf Sporns, *Networks of the Brain*, MIT press, 2010.

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

[6] Sunil K Narang, Akshay Gadde, and Antonio Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc.* IEEE *Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2013, pp. 5445–5449.

[7] Paolo Di Lorenzo, Paolo Banelli, Elvin Isufi, Sergio Barbarossa, and Geert Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *arXiv preprint arXiv:1709.03726*, 2017.

[8] Siheng Chen, Rohan Varma, Aarti Singh, and Jelena Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," IEEE *Trans. Sig. and Inf. Proc. over Netw.*, vol. 2, no. 4, pp. 539–554, 2016.

[9] Mario Coutino, Elvin Isufi, and Geert Leus, "Distributed Edge-Variant Graph Filters," in *Int. Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP)*. IEEE, 2017.

[10] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre 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, 2013.

[11] Elvin Isufi, Andreas Loukas, Andrea Simonetto, and Geert Leus, "Autoregressive moving average graph filtering," IEEE *Trans. Signal Process*, vol. 65, no. 2, pp. 274–288, 2017.

[12] Mario Coutino, Elvin Isufi, and Geert Leus, "Advances in distributed graph filtering," *arXiv preprint arXiv:1808.03004*, 2018.

[13] Chenhui Hu, Jorge Sepulcre, Keith A Johnson, Georges E Fakhri, Yue M Lu, and Quanzheng Li, "Matched signal detection on graphs: Theory and application to brain imaging data classification," *NeuroImage*, vol. 125, pp. 587–600, 2016.

[14] Sundeep Prabhakar Chepuri and Geert Leus, "Subgraph detection using graph signals," in *Proc. of the 50th Asilomar Conf. Sig., Syst. and Comp.*, 2016, pp. 532–534.

[15] Elvin Isufi, Ashvant S. Mahabir, and Geert Leus, "Blind graph topology change detection," *Signal processing letters*, vol. 25, no. 5, pp. 655–659, 2018.

[16] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, 2016, pp. 920–929.

[17] Gonzalo Mateos, Santiago Segarra, and Antonio G Marques, "Inference of graph topology," *Cooperative and Graph Signal Processing: Principles and Applications (PM Djuric and C. Richard, eds.), Amsterdam, Netherlands: Elsevier*, 2018.

[18] Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro, "Network topology inference from spectral templates," *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[19] GV Karanikolas, Georgios B Giannakis, Konstantinos Slavakis, and Richard M Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *Proc. of the IEEE Int. Conf. on Acoust. Speech and Sig. Process. (ICASSP)*, 2016, pp. 6315–6319.

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

[21] Keng-Shih Lu and Antonio Ortega, "Closed form solutions of combinatorial graph laplacian estimation under acyclic topology constraints," *arXiv preprint arXiv:1711.00213*, 2017.

[22] Jin Zhou and Jun-an Lu, "Topology identification of weighted complex dynamical networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 386, no. 1, pp. 481–491, 2007.

[23] Yanning Shen, Brian Baingana, and Georgios B Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. on Sig. Process.*, vol. 65, no. 14, pp. 3675–3687, 2017.

[24] Rasoul Shafipour, Santiago Segarra, Antonio G Marques, and Gonzalo Mateos, "Network topology inference from non-stationary graph signals," in *Proc. of the IEEE Int. Conf. on Acoust. Speech and Sig. Process. (ICASSP)*, 2017, pp. 5870–5874.

[25] Antonio G Marques, Santiago Segarra, Geert Leus, and Alejandro Ribeiro, "Stationary graph processes and spectral estimation," IEEE *Trans. Signal Process*, 2017.

[26] Aliaksei Sandryhaila and Jose MF Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," IEEE *Sig. Proc. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

[27] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[28] Nicholas J Higham, *Functions of matrices: theory and computation*, vol. 104, Siam, 2008.

[29] Katsuhiko Ogata and Yanjuan Yang, *Modern control engineering*, vol. 4, Prentice hall India, 2002.

[30] Lennart Ljung, "Prediction error estimation methods," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 11–21, 2002.

[31] Michel Verhaegen and Vincent Verdult, *Filtering and system identification: a least squares approach*, Cambridge university press, 2007.

[32] M Verahegen and Patrick Dewilde, "Subspace model identification. part i: The output-error state-space model identification class of algorithm," *Int. J. Control*, vol. 56, pp. 1187–1210, 1992.

[33] Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard, "Learning heat diffusion graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 484–499, 2017.

[34] Josef Stoer and Roland Bulirsch, *Introduction to numerical analysis*, vol. 12, Springer Science & Business Media, 2013.

[35] Mario Coutino, Sundeep Chepuri, and Geert Leus, "Sparsest network support estimation: A submodular approach," in *2018 IEEE Data Science Workshop*, 2018.

[36] Luigi Grippo and Marco Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.

[37] K Nodop, R Connolly, and F Girardi, "The field campaigns of the european tracer experiment (etex): Overview and results," *Atmospheric Environment*, vol. 32, no. 24, pp. 4095–4108, 1998.