Online Machine Learning for Graph Topology Identication from Multiple Time Series

Bakht Zaman

UiA University of Agder

Doctoral Dissertations at the University of Agder 298

Bakht Zaman

Online Machine Learning for Graph Topology Identification from Multiple Time Series

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty, Specialisation in ICT

University of Agder
Faculty
2020

Doctoral Dissertations at the University of Agder 298
ISSN: 1504-9272
ISBN: 978-82-8427-000-5

Printed by 07 Media
Oslo

Preface and Acknowledgments

The research work in this dissertation was carried out at the Center Intelligent Signal Processing and Wireless Networks (WISENET), Department of Information and Communication Technology (ICT), University of Agder, Grimstad, Norway, under the supervision of Prof. Baltasar Beferull Lozano. The task of completing this dissertation would not have possible without the support of many people to whom I am highly indebted and I would like to acknowledge them here.

First of all, I would like to express my heartfelt gratitude to my PhD supervisor Prof. Baltasar Beferull Lozano. I thank him for his dedication towards supervising this work, throughout mentoring me, highly motivation towards guiding me, and research discussions during this period of time. This dissertation would not have been possible without his comments, ideas, and discussions. I would also like to thank my co-supervisor Prof. Daniel Romero for the discussions in the reading groups and research meetings. I am very thankful to my co-supervisor Dr. Luis Miguel Lopez-Ramos for his constant support and being always available for meetings that helped me to progress in my research work.

I would like to thank all the WISENET Center members for making my stay at Grimstad wonderful. I am very grateful to Julia for administrative works during my stay at WISENET. I am also very indebted to Emma Horneman. She was always there to help me during this period. I am grateful to Kristine and Mrs. Katharina Pätzold. I would like to thank the former members of the WISENET, Dr. Mohamed Hamid, Dr. Cesar, and Thilina for maintaining a healthy social environment at the WISENET. I am also very grateful to the present members of the WISENET, especially, Dr. Leila, Mohamed Elnourani, Yves, and Dr. Siddarth.

I would also like to thank my friends at Grimstad. Especially, I enjoyed the company of Danial, Muaz, Waqar, Ayaz, Usman. Off course, I cannot forget to acknowledge the friends and mentors I have in Pakistan. I would like to thank Dr. Ziaul Haq for guiding me towards my PhD admission. I am also grateful to Amin Bacha, Asad Ali, Zaiwar, Fazal Muhammad, Mussawer, Khan Wali, Sadam, Ghulam Akbar, and Salman for keeping in touch with me during my stay in Norway.

I am also highly indebted to my family members. I would like to thank my mother, who always supported me. She remains a constant motivation for me during my life. I would also like to thank my brothers, Muhammad Zaman, Gul Nabi, and Akhtar Nabi, who have supported me a lot during throughout my education. I am also very thankful to my father-in-law Murad Ali for his support.

Last but the least, I am highly indebted to my wife Tayyeba for her support and my daughter Ammara. I cannot think of completing this journey without you. Thank you

Ammara for providing me moments to smile during these days.

Bakht Zaman
Grimstad
June 2020

Abstract

High dimensional time series data are observed in many complex systems. In networked data, some of the time series are influenced by other time series. Identifying these relations encoded in a graph structure or topology among the time series is of paramount interest in certain applications since the identified structure can provide insights about the underlying system and can assist in inference tasks. In practice, the underlying topology is usually sparse, that is, not all the participating time series influence each other. The goal of this dissertation pertains to study the problem of sparse topology identification under various settings.

Topology identification from time series is a challenging task. The first major challenge in topology identification is that the assumption of static topology does not hold always in practice since most of the practical systems are evolving with time. For instance, in econometrics, social networks, etc., the relations among the time series can change over time. Identifying the topologies of such dynamic networks is a major challenge.

The second major challenge is that in most practical scenarios, the data is not available at once - it is coming in a streaming fashion. Hence, batch approaches are either not applicable or they become computationally expensive since a batch algorithm is needed to be run when a new datum becomes available.

The third challenge is that the multi-dimensional time series data can contain missing values due faulty sensors, privacy and security reasons, or due to saving energy.

We address the aforementioned challenges in this dissertation by proposing online/-batch algorithms to solve the problem of time-varying topology identification. A model based on vector autoregressive (VAR) process is adopted initially. The parameters of the VAR model reveal the topology of the underlying network. First, two online algorithms are proposed for the case of streaming data. Next, using the same VAR model, two online algorithms under the framework of online optimization are presented to track the time-varying topologies. To evaluate the performance of propose online algorithms, we show that both the proposed algorithms incur a sublinear static regret. To characterize the performance theoretically in time-varying scenarios, a bound on the dynamic regret for one of the proposed algorithms (TIRSO) is derived. Next, using a structural equation model (SEM) for topology identification, an online algorithm for tracking time-varying topologies is proposed, and a bound on the dynamic regret is also derived for the proposed algorithm. Moreover, using a non-stationary VAR model, an algorithm for dynamic topology identification and breakpoint detection is also proposed, where the notion of *local structural breakpoint* is introduced to accommodate the concept of breakpoint where instead of the whole topology, only a few edges vary. Finally, the problem of tracking

VAR-based time-varying topologies with missing data is investigated. Online algorithms are proposed where the joint signal and topology estimation is carried out. Dynamic regret analysis is also presented for the proposed algorithm. For all the previously mentioned works, simulation tests about the proposed algorithms are also presented and discussed in this dissertation. The numerical results of the proposed algorithms corroborate with the theoretical analysis presented in this dissertation.

Publications

The following papers are included in this dissertation and are appended in Appendices A-E at the end of the dissertation.

Paper A **B. Zaman**, L. M. Lopez-Ramos, D. Romero, and B. Beferull-Lozano, "Online topology estimation for vector autoregressive processes in data networks," in *Proc. IEEE Int. Workshop Comput. Advan. Multi-Sensor Adapt. Process.*, Curacao, Dutch Antilles, Dec. 2017.

Paper B **B. Zaman**, L. M. Lopez-Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *Submitted to IEEE Trans. Signal Process.*, arXiv preprint arXiv:1904.01864, Apr. 2019

Paper C **B. Zaman**, L. M. Lopez-Ramos, and B. Beferull-Lozano, "Dynamic regret analysis for online tracking of time-varying structural equation model topologies," *accepted in IEEE Conf. Ind. Electron. Applicat.*, arXiv preprint arXiv:2003.08145, 2020

Paper D L. M. Lopez-Ramos, D. Romero, **B. Zaman**, and B. Beferull-Lozano, "Dynamic network identification from non-stationary vector autoregressive time series," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2018, pp. 773–777.

Paper E **B. Zaman**, L. M. Lopez-Ramos, B. Beferull-Lozano, "Online joint topology identification and signal estimation with inexact proximal online gradient descent," *Submitted to IEEE Trans. Signal Process.*

Abbreviations

ADMM Alternating Direction Method of Multipliers

AR Autoregressive

BCD Block Coordinate Descent

COMID Composite Objective Mirror Descent

EIER Edge Identification Error Rate

JSTIRSO Joint Singal and Topology Identification via Recursive Sparse Online learning

JSTISO Joint Signal and Topology Identification via Sparse Online learning

LMS Least Mean Squares

LS Least Squares

LTI Linear Time Invariant

LTV Linear Time Variant

NMSD Normalized Mean Squared Deviation

NMSE Normalized Mean Squared Error

OBCD Online Block Coordinate Descent

OSGD Online Subgradient Descent

PGD Proximal Gradient Descent

RDA Regularized Dual Averaging

RLS Recursive Least Squares

R-RLS Regularized Recursive Least Squares

ROC Receiver Operating Characteristic

SEM Structural Equation Modeling

TIRSO Topology Identification via Recursive Sparse Online learning

TISO Topology Identification via Sparse Online learning

TVAR Time-varying Vector Autoregressive Process

VAR Vector Autoregressive Process

Contents

Chapter 1

Introduction

1.1 Motivation

In many applications such as transportation networks, industrial environments equipped with sensors, social networks, stock exchanges, and meteorology, data are continuously generated and stored. Much of these data are observed merely for monitoring purposes; on the other hand, it is becoming conventional wisdom that data in sufficient amount can be considered as valuable as physical assets for many sectors of industry. State-of-the-art algorithms are continuously developed to perform inference tasks in this multiverse of data. Despite an ever-growing corpus of theoretical and applied results being developed nowadays to make these data analysis algorithms more efficient, the enormous amount of data generated in some applications (e.g. social networking) still makes it challenging to extract knowledge in real time from it. Dealing with the so-called 'big data' poses many challenges, not only in terms of memory requirements and computational complexity, but also because the systems underlying the data are continuously evolving.

Networks, broadly understood as systems where the aggregation of local effects results into global behavior, are studied not only in electrical engineering applications (e.g. wireless sensor networks) but also in other scientific and technical fields where the notion of connectivity appears both in physical and virtual ways. Data generated at multiple elements of a network (nodes) emerge naturally in a variety of applications such as wire-

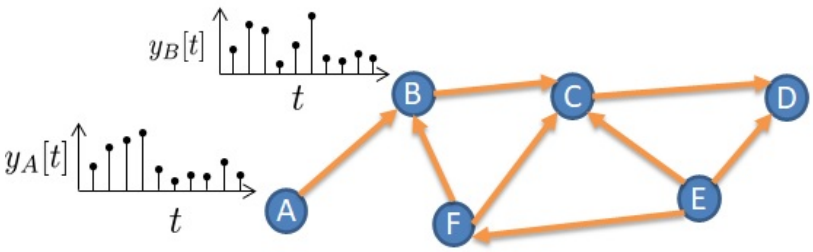


Figure 1.1: A simplified network of multiple time series. An edge in the network denotes a certain notion of causality between two time series. The direction of an edge represents the direction of causality between the two time series.

less sensor networks, transportation, social, and biological networks, to name a few. In these applications, multiple data time-series are jointly observed across network nodes. In the context of sensor networks, each time series corresponds to a network element that monitors a physical parameter of a process. A prominent task in this context is inferring causality graphs that provide the causal relations among a collection of time series (see Fig. 1.1 for a simplified network of multiple time series). Causality graphs often reveal the network structure (also termed topology) of e.g. an underlying social, biological, or brain network. Each node of the graph corresponds to a time series, and connections (edges) appear whenever one time series is inferred to influence another. Identifying their causal interactions is a central problem in many disciplines such as networked cyber-physical systems, sensor and actuator networks, neuroscience, econometrics, bio-informatics and meteorology. Revealing these interactions may offer insights about the dynamics of a given system and may facilitate data processing tasks such as forecasting [1], anomaly detection [2], signal reconstruction [3], clustering [4], [5], filtering [6], [7], sampling [8], dimensionality reduction [9], [10], to name a few.

Consider a sensor and actuator network in a real industrial environment, i.e., Lundin's offshore oil and gas (O&G) platform Edvard-Grieg[1]. A simplified diagram of the decantation system is given in Fig. 1.2, where each node corresponds to either temperature, pressure, or oil-level sensor placed in the system that separates oil, gas, and water. Given the time series data observed in this system, a topology of the network can be estimated that shows the dependencies among the involving time series. Once, the underlying topology is learned, it can be used to predict the future values of the parameters in the system. The topology of the network can also be leveraged to predict the occurrence of certain events. Moreover, the topology can provide insights about inter-relations among the process parameters that govern the dynamics of the system.

In some cases, different causality graphs can be obtained using domain knowledge, see e.g. [11, Ch. 8]; however, such prior domain knowledge is not always available, or the large dimensionality of the data makes such an approach intractable. Such situations call for data-driven approaches, which rely on models that may seem simple when connections are examined one by one, but may be surprisingly expensive when the whole set of connections is applied as a whole.

Many works in the literature related to estimating causality graphs or network topology consider batch approaches, which require the whole data before an estimate is computed. However, in most practical scenarios (weather prediction, industrial sensor data, energy networks, etc.), data are generated in real time and made available in a sequential fashion. For such streaming data, classical batch algorithms have limited applicability due to the data availability constraints or computational bottlenecks. Online learning algorithms, on the other hand, yield an estimate every time a new data sample is received. In addition, advantages such as low computational complexity and capability to deal with sequential data makes them extremely interesting as a tool to tackle streaming data.

In some other applications, the assumption of static model does not hold and the underlying causality graph can change over time. To estimate time-varying topologies

[1] https://www.lundin-petroleum.com/operations/production/norway-edvard-grieg

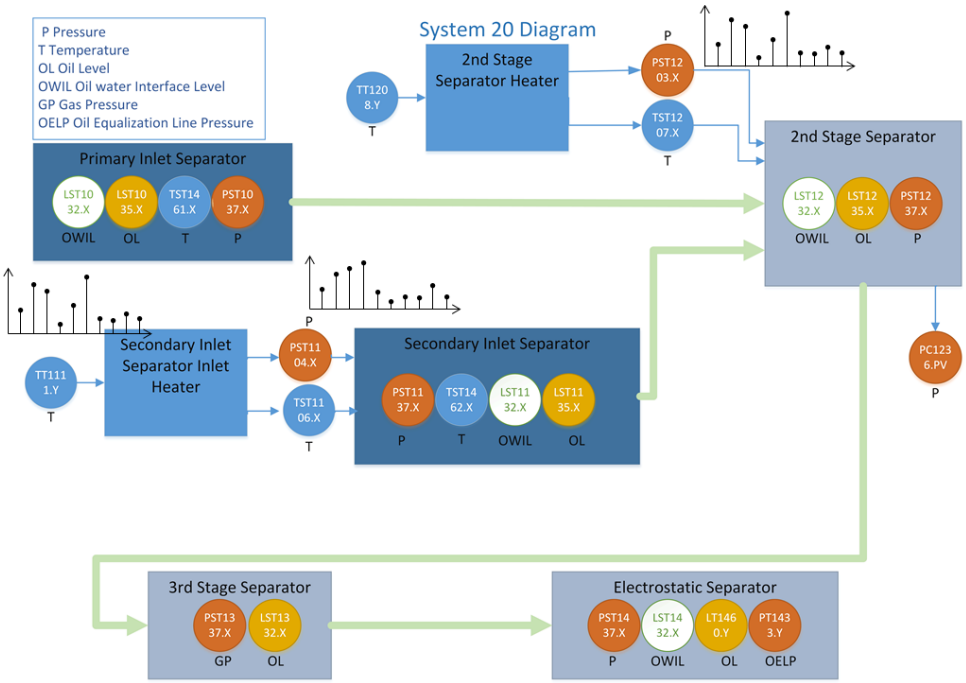


Figure 1.2: A network of separators in an Oil and Gas Platform (Diagram of Edvard-Grieg Platform, Lundin Norway). There are various sensors (labelled according to the NORSOK standard[2]) to monitor temperature, pressure, oil level. A time series is observed at each sensor. The light green thick arrows show the direction of the flow of the fluid in the system.

in a dynamic setting, online algorithms have proven their ability to track time-varying topologies [12].

When a huge amount of data are collected, the data may contain missing values due to certain reasons. For instance, in social networks, there may be missing values in the data due to security or privacy concerns. In a wireless sensor network, observations may be not recorded due to faulty sensors or the effects of lossy compression schemes aimed at saving energy or bandwidth.

To address all the aforementioned challenges, this dissertation develops algorithms and analyze their performance via theoretical results and numerical tests on synthetic and real data.

1.2 Literature Review

The problem of inferring graphs capturing dependencies among multiple time series has recently received a great attention in the literature. The simplest approach to construct a graph from the data looks for all the possible edges among the nodes by placing an edge between two nodes if the sample correlation between the associated time series exceeds a certain threshold [13]. However, such an approach cannot distinguish mediated from unmediated interactions. In order to distinguish between them, one may resort to

[1]https://www.standard.no/en/sectors/energi-og-klima/petroleum/norsok-standards/

conditional independence, partial correlations, or Markov random fields [13, 14, 15, 16, 11, 15, 17]. Recently, many approaches under the framework of graph signal processing [18] are presented; see e.g. [19, 20, 21, 22] for topology identification approaches. In [23], the problem of topology identification is divided into two parts, for which efficient solutions are known. However, these approaches cannot determine the direction of the interactions among variables.

For directed interactions, an alternative notion of interaction is adopted in the literature of structural equation modeling (SEM) [24] by incorporating the influence of exogenous variables; see e.g. [25], [26] and references therein. One may also employ Bayesian networks [11, Sec. 8.1], [27, Sec. 1.2]. However, these models do not generally capture the temporal structure present in time series and account only for memoryless interactions, i.e., they cannot accommodate memory-aware interactions where the value of a time series at a given time instant is related to the past values of other time series.

One of the foremost notions of causality to determine directed relations among multiple time series is Granger causality. Granger [28] proposed a means to infer the direction of causality by building upon the principles that the cause precedes the effect in time and that the cause has a unique information about the effect. However, it turns out that Granger causality is based on optimal prediction error, which is often difficult to compute in practice. Further approaches for topology identification include [29, 30, 31] though their batch nature cannot track temporal changes in the topology.

All methods mentioned in the previous paragraphs work in a batch mode, that is, all the data from all the sensor variables must be available before the learning stage can be started. This calls for the development of online algorithms for topology identification, and this dissertation specifically covers this gap. The goal of this PhD Dissertation is to estimate both the spatial and temporal dynamics of causal relations among time series associated with different variables under various online settings. Existing online topology identification algorithms include [32, 33] for undirected topology identification and [34, 26, 35, 36, 37] for directed topology identification.

Regarding topology identification from non-stationary data, a number of works introduce graphs to capture this notion of time-varying *direct* interactions, either relying on graphical models [38, 39, 14, 34] or structural equation models [26, 40]. Unfortunately, the aforementioned approaches can only deal with memoryless interactions, which limits their applicability to many real-world scenarios.

Only recently, approaches to identify time-varying topologies having the notion of memory into account have been developed [41, 42]. In this dissertation we have developed an approach to identifying models accounting for memory in the interactions, extending the the concept of structural breaks to express the locality of the changes in the underlying models.

Throughout this dissertation, two types of modeling approaches are taken into account: namely structural equation models (SEM) and vector autoregressive (VAR) processes. These models will be described in the next chapter, but a short description is provided here to help introducing the problem statements.

Modeling instantaneous causal relations is usually accomplished based on SEM. Static SEM has been adopted in many works for directed topology identification. However,

the underlying topology may be time-varying and the data may be coming sequentially. Using SEM to model interactions in time series is interesting when the timescale at which interactions occur is shorter than the sampling period.

The framework of VAR process has been extensively adopted to model linear dependencies among time series [43]. In a P-th order VAR model, the current data at multiple variables are assumed to be a noisy superposition of the data from the same variables at the P previous time instants.

Time-varying VAR models are used to track changes in networks where the time scale of the interactions is longer than the sampling period, but still shorter than the timescale at which the changes in the model are produced. We will consider two cases depending on how the model changes are assumed to occur. If the changes are smooth in time, then online algorithms have the potential of doing a good job in tracking the dynamic models. On the other hand, when the changes are assumed to be abrupt, the concept of structural break can be used to detect them, as it can be understood as a form of sparsity.

1.3 Problem Statement

In this dissertation, we consider four types of problems related to topology identification from time series:

- **VAR-based online topology identification.** In this type of problem, we are interested in a *memory-aware* causality graph based on vector autoregressive (VAR) processes. The framework of VAR process has been extensively adopted to model linear dependencies among time series [43]. The parameters of the VAR model reveal the topology of the causality graph (inter-dependencies among the different variables across both space and time), which motivates their estimation. The statement of the problem in the case of streaming data is, given the observation from a VAR process at time instant t and filter order P, compute an online estimate of the memory-aware causality graph.

- **Dynamic SEM-based topology identification.** An important type of instantaneous causal relations is based on SEM. Static SEM has been adopted in many works for directed topology identification. However, the underlying topology may be time-varying and the data may be coming sequentially. To this end, the problem of tracking SEM-based topologies is investigated. The problem statement is: given the sequentially available observations from the dynamic SEM, estimate the time-varying SEM-based sparse topologies in an online fashion with certain convergence guarantees.

- **Dynamic network identification using non-stationary VAR processes.** In many works, it is considered that the topology remain constant over time intervals separated by structural breakpoints, where the topology makes a transition from one model to another. However, in certain applications, e.g., in an industrial network, only a few links may change. This motivates detecting these break points where

only a few links change. The problem statement is: given the non-stationary time series observations, detect the breaking point where some of the edges are changed.

- **Online tracking VAR-based time-varying topologies in the presence of missing data.** The fourth main problem investigated in this dissertation is tracking dynamic VAR-based topologies with missing data. As discussed earlier in this chapter, the missing values in the data occur in certain scenarios and applications. The problem statement is: given the streaming noisy data with missing values, estimate the time-varying VAR-based sparse topologies.

1.4 Contributions and Outline of the Dissertation

This dissertation is organized by including five papers, which are appended at as Appendices A-E. The organization and the contributions of the dissertation are as follows:

- Chapter 2 contains the theoretical background of the work in this dissertation. Specifically, Chapter 2 details the topology identification models adopted in this dissertation. A brief discussion about online learning/optimization is also presented. To analyze the performance of online algorithms, static regret and dynamic regret are introduced.

- Chapter 3 summarizes Paper A [44]. The problem of online estimation of VAR-based causality graphs is under consideration in this paper. In [44], a static VAR model is considered. Two online algorithms are proposed to solve the problem in the case of streaming data. The contributions of the paper are a) online estimation of sparse VAR-based topologies via two algorithms with complementary benefits in terms of the computational complexity and numerical performance. b) The numerical performance of the proposed algorithms show that they converge to the batch solution.

- Chapter 4 discusses Paper B [45]. In this paper using a VAR model, two online algorithms under the framework of *online learning* [46] are derived for the problem of topology identification. The contributions of [45] are: (C1) An online algorithm, termed *Topology Identification via Sparse Online learning* (TISO), which estimates directed VAR causality graphs. Sparse and (possibly) time-varying topologies are tracked while promoting *sparse updates* with constant computational complexity and memory requirements per iteration. Although, TISO is a simple low complexity online algorithm, it has the limitations of sensitivity to noise and input variability. (C2) A second algorithm, named *Topology Identification via Recursive Sparse Online learning* (TIRSO), which improves the tracking performance of TISO and robustness to input variability by minimizing a novel estimation criterion inspired by *recursive least squares* (RLS) where the instantaneous loss function accounts for past samples. (C3) In terms of performance analysis: (i) it is established that the hindsight solution of TISO and TIRSO are asymptotically the same. (ii) The convergence of TISO and TIRSO is established by deriving sublinear static regret bounds.

(iii) A logarithmic regret bound is proved for TIRSO. (iv) For dynamic settings, a dynamic regret bound for TIRSO is derived. Moreover, the steady-state error of TIRSO in time-varying scenarios is quantified in terms of the data properties. (C4) Finally, performance is empirically validated through extensive experiments with synthetic and real data sets.

- Chapter 5 outlines Paper C [37]. In this paper, an important model for topology identification namely SEM is used. The contributions of [37] are: a) an online algorithm for tracking time-varying SEM-based topologies is proposed; b) to evaluate the performance of the proposed algorithm to track time-varying topologies, the dynamic regret analysis is presented for the proposed algorithm and c) the tracking capabilities of the algorithm have been numerically validated for a time-varying scenario under two different assumptions on the model variation.

- Chapter 6 consists of Paper D [47]. In this paper, a dynamic model based on time-varying VAR process for topology identification is considered. The concept of structural breakpoints is extended towards *local structural breakpoint*, which captures the intuitive fact that changes in the interactions are not needed to be synchronized across the system. An algorithm is proposed to identify the topology as well as detect the local breakpoint. Moreover, the simulation results show that the proposed algorithm can identify the topology and the breakpoints.

- Chapter 7 discusses Paper E [48]. In this paper, a dynamic VAR model is used to identify the time-varying topologies when the data is streaming and the noisy observations contain missing values. Online algorithms are proposed to estimate the time-varying topologies with missing values by jointly identifying the topology and estimating the signal from noisy observations. To assess the performance of the proposed online algorithm, a dynamic regret bound is derived. This dynamic regret bound depends on the parameters of the data, the error due to the missing values and noise, the path length, and the parameters of the algorithm.

- Chapter 8 concludes the thesis. The key conclusions of the works presented in this dissertation are discussed. The various possible directions for extending the work in this dissertation are also listed and discussed.

Chapter 2

Background Theory

2.1 Topology Identification Models

To address the problem of topology identification, a number of models have been adopted in the related literature. Every model has its own attributes and comes with certain advantages and limitations. To choose a specific model for a problem at hand, it mostly depends on the type of application, data, and the underlying objective of topology identification. Therefore, no single model can be embraced and be applied everywhere. This motivates the need for various topology identification models and analyzing their characteristics. Next, we describe the models used for the of topology identification in this dissertation.

2.1.1 Vector Autoregressive Processes

The main model that we adopt for topology identification in this dissertation is vector autoregressive (VAR) process. VAR process is used for topology inference in literature [49]. For instance, in neuroscience, VAR process has been used to identify causality graphs [50, 51]. Similarly, VAR processes have widely been used in econometrics [43]. Next, we present a brief description of VAR models.

Consider a collection of N time series $\{y_n[t]\}_t$, $n = 1, ..., N$, where $y_n[t]$ denotes the value of the n-th time series at time t. By defining $\boldsymbol{y}[t] \triangleq [y_1[t], \ldots, y_N[t]]^\top$, a VAR process of order P is given by

$$\boldsymbol{y}[t] = \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{y}[t-p] + \boldsymbol{u}[t], \tag{2.1}$$

where $\boldsymbol{A}_p \in \mathbb{R}^{N \times N}, p = 1, \ldots, P$, are the VAR parameters and $\boldsymbol{u}[t] \triangleq [u_1[t], \ldots, u_N[t]]^\top$ is the innovation process vector. This process is generally assumed to be a white zero-mean stochastic process, i.e., $\mathbb{E}[\boldsymbol{u}[t]] = \boldsymbol{0}_N$ and $\mathbb{E}[\boldsymbol{u}[t]\boldsymbol{u}^\top[\tau]] = \boldsymbol{0}_{N \times N}$ for $t \neq \tau$. The covariance matrix $\mathbb{E}[\boldsymbol{u}[t]\boldsymbol{u}^\top[t]]$ of the innovation process is generally assumed to be nonsingular.

Stable VAR processes are analyzed in the literature since the analysis of stable VAR processes is usually tractable. A VAR process is said to be stable if $\det(\boldsymbol{I}_N - \boldsymbol{A}_1 z - \boldsymbol{A}_2 z -, \ldots, \boldsymbol{A}_P z) \neq 1$ for all $|z| < 1$. A reason to study stable VAR processes is that

stability implies stationarity [43, Prop. 2.1]. Hence, stable VAR processes are also stationary. Some of the practical dynamic systems cannot be modeled by stable/stationary processes. To study these processes, understanding stable/stationary processes provides an initial step to dig deeper towards studying unstable/non-stationary processes.

2.1.2 Structural Equation Model

SEM is one of the predominant models used for directed topology identification. One of the key features of SEM is its simplicity, which makes these models tractable. Due to its tractability and the ability to identify directed relations, SEMs have been used in across the fields such as social networks [35], genetics [52], and economics [53].

Consider a network of N nodes, and let y_i^t be the observed value at i-th node in a linear static SEM, given by

$$y_i^t = \sum_{j=1, j\neq i}^{N} a_{ij} y_j^t + b_{ii} x_i^t + e_i^t, \tag{2.2}$$

where a_{ij} are SEM parameters capturing directed relations ($a_{ij} \neq a_{ji}$), x_i is the exogenous variable for the i-th node, b_{ii} is the influence of the exogenous variable, and e_i^t denotes the un-modeled dynamics. Observe from (2.2) that SEM depends on two type of variables, i.e., endogenous and exogenous variables. The endogenous variables y_i^t are caused by other variables in the model. The endogenous variables y_i^t can cause each other. The exogenous variables x_i^t, on the other hand, are not caused by other variables in the model. The inclusion of exogenous variables makes SEM distinctive from other models. Exogenous variables perfectly model in certain scenarios, where an external input is available.

2.2 Background on Online Optimization

Consider the generic unconstrained optimization problem

$$\underset{\boldsymbol{a}}{\text{minimize}} \ \frac{1}{T_0} \sum_{t=0}^{T_0-1} h_t(\boldsymbol{a}), \tag{2.3}$$

where $h_t(\boldsymbol{a})$ is a convex function, which depends on the data received at time t and which will typically represent some loss function related to the estimation problem and possibly some regularizer. Solving (2.3) via batch approaches requires $\{h_t(\boldsymbol{a})\}_{t=0}^{T_0-1}$ to be available. This makes the batch approaches inappropriate to be used in real-time streaming data applications. Moreover, the computational complexity and memory requirements generally grows super-linearly with T_0 for batch approaches. This motivates the design of online algorithms, as we introduce here.

Let an estimate of the solution to (2.3) at time t produced by an online algorithm be denoted by $\boldsymbol{a}[t+1]$. Online algorithms compute a new estimate $\boldsymbol{a}[t+1]$ every time a new data element (a new $h_t(\boldsymbol{a})$) is processed. At time t, $\boldsymbol{a}[t+1]$ is obtained from $\boldsymbol{a}[t]$, $h_t(\boldsymbol{a})$, and possibly some additional information carried from each update to the next. The memory requirements and number of arithmetic operations per iteration must

not grow unbounded for increasing t. Thus, online algorithms are especially appealing when data vectors are received sequentially or T_0 is so large that batch solvers are not computationally affordable. Additionally, online algorithms can track changes in the underlying time-varying model.

The analytical framework of *online learning and online convex optimization* is leveraged not only for the derivation of many of the algorithms developed in the dissertation, but also for the performance analysis. Key to understand the possibilities and limitations of an online algorithm, and to be able to compare between algorithms that are intended to solve the same problem, is the concept of *regret*. In fact, two types of regret are taken into account in our analysis, namely the static regret and the dynamic regret.

2.2.1 Static Regret

To theoretically evaluate the performance of online algorithms, regret is a common metric used in the literature. Regret quantifies the cumulative loss incurred by an online algorithm relative to the cumulative loss corresponding to the optimal *batch* solution in hindsight. Mathematically, the static regret at $T_0 - 1$ is given by [46]:

$$R_s[T_0] \triangleq \sum_{t=0}^{T_0-1} [h_t(\boldsymbol{a}[t]) - h_t(\boldsymbol{a}^*[T_0])], \tag{2.4}$$

where

$$\boldsymbol{a}^*[T_0] \triangleq \arg\min_{\boldsymbol{a}} \ (1/T_0) \sum_{t=0}^{T_0-1} h_t(\boldsymbol{a}) \tag{2.5}$$

is the optimal *batch* hindsight solution, i.e., the batch solution after T_0 data vectors have been processed. To be deemed admissible, online algorithms must yield a *sublinear regret*, i.e., $R_s[T_0]/T_0 \to 0$ as $T_0 \to \infty$. A sublinear regret for an online algorithm implies that the online algorithm has a performance that is asymptotically as good as the batch solution *on average*. Online algorithms do not consider statistical assumptions on the data, which can even be generated by an "adversary" [54]. This means that online algorithms under the framework of online learning do not need the data to be independent as opposed to stochastic optimization algorithms [55].

2.2.2 Dynamic Regret

In dynamic settings where the parameters of the data generating process vary over time, $\boldsymbol{a}^*[T_0]$ in (2.5), used in static regret, may not be a suitable reference since its computation involves potentially very old data, namely $\{h_t\}_{t \ll T_0}$. In such scenarios, it is customary to compare the estimates of an online algorithm against the instantaneous minimizer $\boldsymbol{a}^\circ[t] \triangleq \arg\min_{\boldsymbol{a}} h_t(\boldsymbol{a})$ by means of the so-called dynamic regret [56, 57, 58, 59]:

$$R_d[T_0] \triangleq \sum_{t=0}^{T_0-1} [h_t(\boldsymbol{a}[t]) - h_t(\boldsymbol{a}^\circ[t])].$$

The dynamic regret has recently gained attention and is used to quantify the performance of an online algorithm in dynamic settings, where the online algorithm is required to be

adaptive in order to track the variations in a hindsight solution sequence; see e.g. [12] and references therein. Deriving a sublinear dynamic regret is always not possible for time-varying scenarios. Usually, a bound on the dynamic regret is derived in terms of the so-called *path length*, which denotes the cumulative variations in the two instantaneous minimizers.

Chapter 3

Topology Identification from Multiple Streaming Time Series

This chapter summarizes Paper A ([44]).

3.1 Motivation

Different approaches to identify the causal relations are discussed in Chapter 1. As described earlier in Chapter 1, the notion of Granger causality is very elegant, however, it is not very practical to apply in the real-world since it is based on the optimal prediction error, which is not always easy to compute [60, p. 33], [61]. Hence, alternative definitions of causality are used in practice. To this end, VAR model is adopted in many applications to model linear dependencies among time series and VAR causality graphs are frequently estimated in econometrics, bio-informatics, neuroscience, and engineering [62, 63, 64, 49]. An important aspect of the VAR process is that it provides a model to predict the time series 1-step ahead. VAR causality is further motivated by the widespread usage of VAR models to approximate the response of systems of linear partial differential equations [65]. Next, causality graphs based on VAR processes are discussed.

3.2 Introduction to VAR Causality Graphs

Consider a collection of N time series $\{y_n[t]\}_t$, $n = 1, ..., N$, where $y_n[t]$ denotes the value of the n-th time series at time t. A prominent notion of causality can be defined using VAR models. To this end, let $\boldsymbol{y}[t] \triangleq [y_1[t], \ldots, y_N[t]]^\top$ and define a VAR time series $\{\boldsymbol{y}[t]\}_t$ as a sequence generated by the order-P VAR model [43]

$$\boldsymbol{y}[t] = \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{y}[t-p] + \boldsymbol{u}[t], \tag{3.1}$$

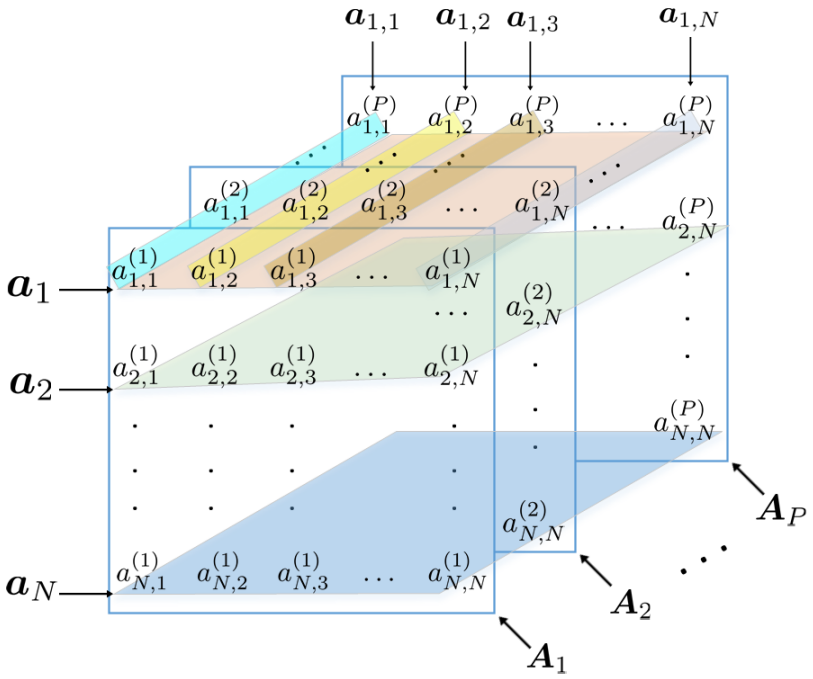


Figure 3.1: Structure of the VAR parameter matrices. The structure of $\boldsymbol{a}_{nn'}$ and $\boldsymbol{a}_n$ can be viewed clearly in terms of the VAR parameters.

where $\boldsymbol{A}_p \in \mathbb{R}^{N\times N}, p = 1,\dots,P$, are the VAR parameters and $\boldsymbol{u}[t] \triangleq [u_1[t],\dots,u_N[t]]^\top$ is the *innovation process*. This process is generally assumed to be a white zero-mean stochastic process, i.e., $\mathbb{E}[\boldsymbol{u}[t]] = \boldsymbol{0}_N$ and $\mathbb{E}[\boldsymbol{u}[t]\boldsymbol{u}^\top[\tau]] = \boldsymbol{0}_{N\times N}$ for $t \neq \tau$. A causality graph $\mathcal{G} \triangleq (\mathcal{V},\mathcal{E})$ is a collection of a vertex set $\mathcal{V}$ and a set of edges $\mathcal{E}$ where the n-th vertex in $\mathcal{V} = \{1,\dots,N\}$ is identified with the n-th time series $\{y_n[t]\}_t$ and there is an edge from n' to n (i.e. $(n,n') \in \mathcal{E}$) if and only if $\{y_{n'}[t]\}_t$ causes $\{y_n[t]\}_t$ according to a certain causality notion. Specifically, for VAR-causality: $\{y_{n'}[t]\}_t$ VAR-causes $\{y_n[t]\}_t$ whenever $n' \in \mathcal{N}(n)$, where $\mathcal{N}(n)$ is the neighborhood of the node n. Equivalently, $\{y_{n'}[t]\}_t$ VAR-causes $\{y_n[t]\}_t$ if $\boldsymbol{a}_{n,n'} \neq \boldsymbol{0}_P$, where $\boldsymbol{a}_{n,n'} \triangleq [a^{(1)}_{n,n'},\dots,a^{(P)}_{n,n'}]^\top$. The structure of the VAR parameters is presented in Fig. 3.1.

3.3 Online Topology Identification

Before considering an online approach, let us first introduce the batch problem. The problem statement is: given P and the observations $\{\boldsymbol{y}[t]\}_{t=0}^{T-1}$ generated by a VAR process of order P, estimate the VAR coefficient matrices $\{\boldsymbol{A}_p\}_{p=1}^P$. The batch problem is formulated along the lines of [66] by minimizing a least-squared criterion in (A.7) by considering group-lasso regularization to introduce sparsity in edges of the topology.

When the whole data is not available or when the computational or memory requirements constrain us to apply batch algorithms, we resort to online algorithms. Since the problem is separable across nodes, the problem for each node can be solved in separately and the solution can be computed in parallel. Hence, from now on, the n-th problem is considered. The cost function at time t for the n-th problem that an online algorithm minimizes is selected to be a time-dependent exponentially weighted least-squared functional

in [44]:

$$\hat{\boldsymbol{a}}_n[t] = \arg\min_{\boldsymbol{a}_n[t]} \sum_{\tau=P}^{t} \gamma^{t-\tau} \left(y_n[\tau] - \boldsymbol{g}^\top[\tau]\boldsymbol{a}_n[t]\right)^2 + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^{N} \|\boldsymbol{a}_{n,n'}[t]\|_2 , \tag{3.2}$$

where

$$\boldsymbol{a}_n[t] \triangleq [\boldsymbol{a}_{n,1}^\top[t], \boldsymbol{a}_{n,2}^\top[t], ..., \boldsymbol{a}_{n,N}^\top[t]]^\top \in \mathbb{R}^{NP}, \tag{3.3}$$

$$\boldsymbol{g}[\tau] \triangleq \mathrm{vec}\big(\big[\boldsymbol{y}[\tau-1], \ldots, \boldsymbol{y}[\tau-P]\big]^\top\big) \in \mathbb{R}^{NP}, \tag{3.4}$$

and $0 < \gamma \leq 1$ is a user-selected forgetting factor. This loss functional also enables an online algorithm to track the possible changes in the topology. Next, two online algorithms are proposed in [44] to solve the problem of VAR-based topology identification for streaming data.

The first algorithm is proposed by solving (3.2) using a sub-gradient based approach following a closely-related structure to recursive least squares (RLS) algorithm. The proposed approximate solution relies on the assumption that the estimated coefficients do not change abruptly between consecutive time steps. The complexity of **Algorithm 6** in [44] is dominated by step. 6, i.e., N times $\mathcal{O}(N^2P^2)$, hence the overall complexity is $\mathcal{O}(N^3P^2)$.

For a large N, the computational complexity of **Algorithm 6** becomes prohibitive. To this end, we propose an online method with quadratic complexity in N. The proposed method is based on performing a single iteration of block coordinate descent (BCD) to minimize (3.2). The objective in (3.2) can be written as:

$$\arg\min_{\boldsymbol{a}_n[t]} \boldsymbol{a}_n^\top[t]\boldsymbol{\Phi}[t]\boldsymbol{a}_n[t] - 2\boldsymbol{r}_n^\top[t]\boldsymbol{a}_n[t] + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^{N} \|\boldsymbol{a}_{n,n'}[t]\|_2$$

where

$$\boldsymbol{\Phi}[t] \triangleq \sum_{\tau=P}^{t} \gamma^{t-\tau} \boldsymbol{g}[\tau]\boldsymbol{g}^\top[\tau], \tag{3.5}$$

$$\boldsymbol{r}_n[t] \triangleq \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau]\boldsymbol{g}[\tau]. \tag{3.6}$$

For each t and n, the proposed algorithm performs N block updates: the i-th update modifies the i-th group $\boldsymbol{a}_{n,i}[t]$ whereas all other entries in $\boldsymbol{a}_n[t]$ are kept fixed. Each problem for $\boldsymbol{a}_{n,i}[t]$ is solved via Newton's algorithm detailed in **Algorithm 8** in [44]. The sample auto-correlation matrix and sample cross-correlation matrix are updated recursively. The overall algorithm is called online Block coordinate descent for topology identification and is tabulated in **Algorithm 7** in [44].

In numerical simulations based on synthetic data, a network with $N = 15$ nodes where the graph is generated via Erdős-Rényi model. A stable VAR process with $P = 5$ is generated by drawing the active coefficients of $\boldsymbol{A}_p$ from a Gaussian distribution and time series of T time instants is generated according to (3.1) with $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, 0.02\boldsymbol{I})$.

The performance of the proposed algorithms is analyzed via two error measures, namely, the normalized mean squared deviation (NMSD) defined as:

$$\text{NMSD}[t] = \frac{\mathbb{E}\left[\sum_{n=1}^{N} \|(\hat{\boldsymbol{a}}_n[t] - \boldsymbol{a}_n)\|_2^2\right]}{\mathbb{E}\left[\sum_{n=1}^{N} \|\boldsymbol{a}_n\|_2^2\right]}, \tag{3.7}$$

where $\boldsymbol{a}_n$ are the true VAR parameters corresponding to the n-th node; and the normalized mean square error (NMSE):

$$\text{NMSE}[t] = \frac{\mathbb{E}\left[\sum_{n=1}^{N} \left(y_n[t] - \boldsymbol{g}^\top[t]\hat{\boldsymbol{a}}_n[t]\right)^2\right]}{\mathbb{E}\left[\|\boldsymbol{y}[t]\|_2^2\right]}. \tag{3.8}$$

The results show that the proposed online algorithms approach the batch solution after processing a large number of samples.

The online algorithms in [44] do not necessarily yield sparse iterates before convergence. Moreover, they are not supported by theoretical convergence guarantees in the online scenarios. To address the aforementioned challenges, we propose two online algorithms in [45] under the framework of online convex optimization, which are presented in the next chapter.

3.4 Summary of the Chapter

- This chapter summarizes [44], where two online algorithms are proposed for VAR-based topology identification for sequential data.
- Both algorithms have complementary benefits in terms of computational complexity and NMSD/NMSE performance.
- The numerical results show that the proposed algorithms converge to the batch solution of the problem.
- Further investigation is required to propose online algorithms supported by convergence guarantees, which yield sparse topologies at each time instant.

Chapter 4

Online Topology Identification in Vector Autoregressive Processes with Performance Guarantees

This chapter summarizes Paper B ([45]).

4.1 Motivation

The problem of online topology identification in vector autoregressive processes is considered in [44], where two online algorithms are proposed. However, the algorithms do not yield sparse iterates and their computational complexity is not suitable for big data scenarios. Moreover, the algorithms are not supported by convergence guarantees. To tackle these challenges, the work in was [45] initiated and accomplished. This chapter describes a brief summary of the work in [45].

4.2 Topology Identification via Online Optimization

To solve the problem of online topology identification, two online algorithms in [45] are presented under the framework of online optimization. To propose the online algorithms based on the online machine learning framework, we set $h_t(\boldsymbol{a}_n)$ introduced in Chapter 2 as:

$$h_t(\boldsymbol{a}_n) = f_t^{(n)}(\boldsymbol{a}_n) + \Omega^{(n)}(\boldsymbol{a}_n), \tag{4.1}$$

where $f_t^{(n)}(\boldsymbol{a}_n)$ is a convex loss function for the n-th node since the loss function is separable across nodes and $\Omega^{(n)}(\boldsymbol{a}_n)$ is a convex regularizer for the n-th node. To propose the first online algorithm, we set

$$f_t^{(n)}(\boldsymbol{a}_n) = \ell_{t+P}^{(n)}(\boldsymbol{a}_n), \tag{4.2}$$

where

$$\ell_t^{(n)}(\boldsymbol{a}_n) \triangleq \frac{1}{2}\left(y_n[t] - \boldsymbol{g}^\top[t]\boldsymbol{a}_n\right)^2 \tag{4.3}$$

and

$$\Omega^{(n)}(\boldsymbol{a}_n) = \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \,. \tag{4.4}$$

The update at time t of the resultant online algorithm is obtained after solving the following objective under the framework of online learning by adopting the composite objective mirror descent (COMID) method [67]:

$$\boldsymbol{a}_n[t+1] \;=\; \arg\min_{\boldsymbol{a}_n} \big[\alpha_t \tilde{\nabla} f_t^{(n)\top}(\boldsymbol{a}_n[t])\,(\boldsymbol{a}_n - \boldsymbol{a}_n[t]) + \frac{1}{2\alpha_t}\,\|\boldsymbol{a}_n - \boldsymbol{a}_n[t]\|_2^2 + \alpha_t \Omega^{(n)}(\boldsymbol{a}_n)\big], \tag{4.5}$$

where $\tilde{\nabla} f_t^{(n)}(\boldsymbol{a}_n[t])$ is a subgradient of $f_t^{(n)}$ at point $\boldsymbol{a}_n[t]$, i.e., $\tilde{\nabla} f_t^{(n)}(\boldsymbol{a}[t]) \in \partial f_t^{(n)}(\boldsymbol{a}_n[t])$, and $\alpha_t > 0$ is a step size. Given $\tilde{\nabla} f_t^{(n)}(\boldsymbol{a}_n[t])$, the problem in (4.5) can be solved in parallel and the update expression for each sub-problem is given by (B.19). It can be observed from (B.19) that the online update yields sparse topology estimates. We name the resulting algorithm as *Topology Identification via Sparse Online learning* (TISO) and is presented here in **Algorithm 1**.

TISO only requires $\mathcal{O}(N^2P)$ memory entries to store the last P data vectors and the last estimate. The computational complexity of TISO is $\mathcal{O}(N^2P)$, which is in the same order as the number of parameters to be estimated. Thus, TISO can arguably be deemed as a low-complexity algorithm.

Each update of TISO depends on the data through the *instantaneous* loss $\ell_t^{(n)}(\boldsymbol{a}_n[t])$, which is based on one data vector. This renders TISO a computationally efficient algorithm for online topology identification and it also increases sensitivity to noise and input variability. To this end, we propose an alternative approach at the expense of a moderate increase in computational complexity and memory requirements. From an algorithmic point of view, observe that TISO can be seen as a generalization of the least mean squared (LMS) algorithm. To reduce the output variability and to speed up convergence of LMS, it is customary to adopt recursive least squares (RLS), which uses the previous data allowing to control the influence of each data vector on future estimates through forgetting factors. Following the same track, the trick is to replace the *instantaneous* loss $\ell_t^{(n)}(\boldsymbol{a}_n)$ in (4.3) with a *running average* loss. Specifically, $f_t^{(n)}(\boldsymbol{a}_n) = \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)$ in (4.1) with

$$\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) \triangleq \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \ell_\tau^{(n)}(\boldsymbol{a}_n), \tag{4.6}$$

where $\gamma \in (0,1)$ is the user-selected forgetting factor and $\mu = 1-\gamma$ is set to normalize the exponential weighting window, i.e., $\mu \sum_{\tau=0}^{\infty} \gamma^\tau = 1$. To derive an algorithm with constant computational and memory complexity, we use the structure of (4.6). To this end, expand

Algorithm 1 Topology Identification via Sparse Online optimization (TISO)

Input: $\lambda, \{\alpha_t\}_t, \{\boldsymbol{y}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\boldsymbol{a}_n[\tau]\}_{n=1}^{N}, \tau = P+1, \ldots$
Initialization: $\boldsymbol{a}_n[P] = \boldsymbol{0}_{NP}, n = 1, \ldots, N$

1: **for** $t = P, P+1, \ldots$ **do**
2: Receive data vector $\boldsymbol{y}[t]$
3: $\boldsymbol{g}[t] = \mathrm{vec}\big([\boldsymbol{y}[t-1], \ldots, \boldsymbol{y}[t-P]]^\top\big)$
4: **for** $n = 1, 2, \ldots, N$ **do**
5: $\boldsymbol{v}_n[t] = (\boldsymbol{g}^\top[t]\,\boldsymbol{a}_n[t] - y_n[t])\boldsymbol{g}[t]$
6: **for** $n' = 1, 2, \ldots, N$ **do**
7: $\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] = \boldsymbol{a}_{n,n'}[t] - \alpha_t \boldsymbol{v}_{n,n'}[t]$
8: $\boldsymbol{a}_{n,n'}[t+1] = \boldsymbol{a}^{\mathrm{f}}_{n,n'}[t]\left[1 - \frac{\alpha_t \lambda\ \mathbb{1}\{n \neq n'\}}{\left\|\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t]\right\|_2}\right]_+$
9: **end for**
10: **end for**
11: **end for**

and rewrite (4.6) to obtain

$$\begin{aligned}\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) &= \frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}\big(y_n^2[\tau] + \boldsymbol{a}_n^\top \boldsymbol{g}[\tau]\boldsymbol{g}^\top[\tau]\boldsymbol{a}_n - 2y_n[\tau]\boldsymbol{g}^\top[\tau]\boldsymbol{a}_n\big) \\ &= \frac{1}{2}\boldsymbol{a}_n^\top \boldsymbol{\Phi}[t]\boldsymbol{a}_n - \boldsymbol{r}_n^\top[t]\boldsymbol{a}_n + \frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[\tau], \end{aligned} \tag{4.7}$$

where

$$\boldsymbol{\Phi}[t] \triangleq \mu \sum_{\tau=P}^{t} \gamma^{t-\tau}\boldsymbol{g}[\tau]\,\boldsymbol{g}^\top[\tau], \tag{4.8a}$$

$$\boldsymbol{r}_n[t] \triangleq \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau]\,\boldsymbol{g}[\tau]. \tag{4.8b}$$

The variables $\boldsymbol{\Phi}[t]$ and $\boldsymbol{r}_n[t]$ can be updated recursively as

$$\boldsymbol{\Phi}[t] = \gamma\,\boldsymbol{\Phi}[t-1] + \mu\,\boldsymbol{g}[t]\,\boldsymbol{g}^\top[t], \tag{4.9a}$$

$$\boldsymbol{r}_n[t] = \gamma\,\boldsymbol{r}_n[t-1] + \mu\,y_n[t]\,\boldsymbol{g}[t]. \tag{4.9b}$$

We name this algorithm as *Topology Identification via Recursive Sparse Online optimization* (TIRSO) and tabulated as **Algorithm 2**.

The computational complexity is dominated by step 7 of **Algorithm 2**, which is $\mathcal{O}(N^3P^2)$ operations per t. However, exploiting the group-sparse structure of $\tilde{\boldsymbol{a}}_n[t]$ may reduce the computation by disregarding the columns of $\boldsymbol{\Phi}[t]$ corresponding to the zero entries of $\tilde{\boldsymbol{a}}_n[t]$. If, for instance, the number of edges is $\mathcal{O}(N)$, then the complexity of TIRSO becomes $\mathcal{O}(N^2P^2)$ per t. Regarding memory complexity, TIRSO requires N^2P^2 memory positions to store $\boldsymbol{\Phi}[t]$ and N^2P positions to store $\{\boldsymbol{r}_n[t]\}_{n=1}^{N}$.

To support the decision of setting $f_t^{(n)}(\boldsymbol{a}_n) = \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)$ to develop TIRSO for solving a batch objective in an online fashion, we establish that the batch problems that TISO

Algorithm 2 Topology Identification via Recursive Sparse Online optimization (TIRSO)

Input: $\gamma, \mu, P, \lambda, \sigma^2, \{\alpha_t\}_t, \{\boldsymbol{y}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\tilde{\boldsymbol{a}}_n[t]\}_{n=1}^N, t = P+1, ...$
Initialization: $\tilde{\boldsymbol{a}}_n[P] = \mathbf{0}_{NP}, n = 1, ..., N$, $\boldsymbol{\Phi}[P-1] = \sigma^2 \boldsymbol{I}_{NP}$
$\boldsymbol{r}_n[t] = \mathbf{0}_{NP}, n = 1, ..., N$

for $t = P, P+1, \ldots$ **do**
  Receive data vector $\boldsymbol{y}[t]$
  $\boldsymbol{g}[t] = \text{vec}\big([\boldsymbol{y}[t-1], \ldots, \boldsymbol{y}[t-P]]^\top\big)$
  $\boldsymbol{\Phi}[t] = \gamma\, \boldsymbol{\Phi}[t-1] + \mu\, \boldsymbol{g}[t]\, \boldsymbol{g}^\top[t]$
  for $n = 1, \ldots, N$ **do**
    $\boldsymbol{r}_n[t] = \gamma\, \boldsymbol{r}_n[t-1] + \mu\, y_n[t]\, \boldsymbol{g}[t]$
    $\tilde{\boldsymbol{v}}_n[t] = \boldsymbol{\Phi}[t]\, \tilde{\boldsymbol{a}}_n[t] - \boldsymbol{r}_n[t]$
    for $n' = 1, 2, \ldots, N$ **do**
      $\tilde{\boldsymbol{a}}^{\text{f}}_{n,n'}[t] = \tilde{\boldsymbol{a}}_{n,n'}[t] - \alpha_t \tilde{\boldsymbol{v}}_{n,n'}[t]$
      $\tilde{\boldsymbol{a}}_{n,n'}[t+1] = \tilde{\boldsymbol{a}}^{\text{f}}_{n,n'}[t] \left[1 - \frac{\alpha_t \lambda\, \mathbb{1}\{n \neq n'\}}{\left\|\tilde{\boldsymbol{a}}^{\text{f}}_{n,n'}[t]\right\|_2}\right]_+$
    end for
  end for
end for

and TIRSO implicitly solve become asymptotically equivalent as $T \to \infty$. Theorem 1 in [45] essentially establishes not only that the TISO and TIRSO hindsight objectives are asymptotically the same but also that their minima and minimizers asymptotically coincide. Since the TISO hindsight objective equals the batch objective considered, it follows that the TIRSO hindsight objective asymptotically approaches the batch objective. This observation is very important since the regret analysis in the next section will establish that the TISO and TIRSO estimates asymptotically match their hindsight counterparts.

4.3 Static and Dynamic Regret Analysis

We characterize the performance of TISO and TIRSO analytically by establishing that the sequences of estimates produced by these algorithms yield a sublinear static regret, which is a basic requirement in online optimization. This property means that, on average and asymptotically, the online estimates perform as well as their hindsight counterparts. The upcoming results will make use of one or more of the following assumptions:

A1. *Bounded samples:* There exists $B_y > 0$ such that $|y_n[t]|^2 \leq B_y \; \forall\, n, t$.

A2. *Bounded minimum eigenvalue of* $\boldsymbol{\Phi}[t]$*:* There exists $\beta_{\tilde{\ell}} > 0$ such that $\lambda_{\min}(\boldsymbol{\Phi}[t]) \geq \beta_{\tilde{\ell}},\; \forall\, t \geq P$.

A3. *Bounded maximum eigenvalue of* $\boldsymbol{\Phi}[t]$*:* There exists $L > 0$ such that $\lambda_{\max}(\boldsymbol{\Phi}[t]) \leq L,\; \forall\, t \geq P$.

A4. *Asymptotically invertible sample covariance:* There exists T_m and β such that

$$\lambda_{\min}\left(\frac{1}{t-P}\sum_{\tau=P}^{t}\boldsymbol{g}[\tau]\,\boldsymbol{g}^{\top}[\tau]\right)\geq\beta\quad\forall\,t\geq T_m. \tag{4.10}$$

Note that A1 entails no loss of generality in real-world applications, where data are bounded and thus B_y always exists. A2 usually holds in practice unless the data is redundant, meaning that some time series can be obtained as a linear combination of the others. A3 will also hold in practice since it can be shown that it is implied by A1. In particular, if A1 holds, then A3 holds with $L = PNB_y$. Similarly, A4 will also generally hold since it is a weaker version of A2.

Theorem 2 in [45] establishes that under the assumptions A1 and A4, the regret for the n-th subproblem is given by

$$R_s^{(n)}[T]=\mathcal{O}\left(PNB_y\left(1/\beta(B_y\sqrt{PN}+\sqrt{B_y^2PN+\beta B_y})\right)^2\sqrt{T}\right). \tag{4.11}$$

Similarly, the corresponding static regret for TIRSO is bounded under the assumptions A1-3 in Theorem 3 in [45] as:

$$R_s^{(n)}[T]=\mathcal{O}\left(L\left(1/\beta_{\tilde{\ell}}(B_y\sqrt{PN}+\sqrt{B_y^2PN+\beta_{\tilde{\ell}}B_y})\right)^2\sqrt{T}\right). \tag{4.12}$$

Using the strong convexity of the data-fitting function of TIRSO, a logarithmic bound is provided in Theorem (4) in [45] for a diminishing step size $\alpha_t = 1/(\beta_{\tilde{\ell}}t)$:

$$\tilde{R}_s^{(n)}[T]\leq\frac{G_{\tilde{\ell}}^2}{2\beta_{\tilde{\ell}}}\left(\log(T-P+1)+1\right)+\frac{1}{2\alpha_{P-1}}\left(1/\beta_{\tilde{\ell}}(B_y\sqrt{PN}+\sqrt{B_y^2PN+\beta_{\tilde{\ell}}B_y})\right)^2, \tag{4.13}$$

where $G_{\tilde{\ell}}\triangleq(1+\kappa_{\boldsymbol{\Phi}})\sqrt{PN}B_y$ with $\kappa_{\boldsymbol{\Phi}}=L/\beta_{\tilde{\ell}}$.

Next, the results about the dynamic regret are presented. For the n-th subproblem, the dynamic regret is defined as:

$$\tilde{R}_d^{(n)}[T]\triangleq\sum_{t=P}^{T}\left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])-\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t])\right], \tag{4.14}$$

where $\tilde{\boldsymbol{a}}_n[t]$ is the TIRSO estimate and $\tilde{\boldsymbol{a}}_n^{\circ}[t] = \arg\min_{\tilde{\boldsymbol{a}}_n}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n)$. It can be easily shown that the static regret is upper-bounded by the dynamic regret. Attaining a low dynamic regret is therefore more challenging because the estimator under consideration is compared with a *time-varying* reference. This implies that a sublinear dynamic regret may not be attained if this time-varying reference changes too rapidly, which generally occurs when the tracked parameters vary too quickly. To this end, the dynamic regret is commonly upper-bounded in terms of the cumulative distance between two consecutive instantaneous optimal solutions, known as *path length*:

$$W^{(n)}[T]\triangleq\sum_{t=P+1}^{T}\left\|\tilde{\boldsymbol{a}}_n^{\circ}[t]-\tilde{\boldsymbol{a}}_n^{\circ}[t-1]\right\|_2. \tag{4.15}$$

Theorem 5 in [45] presents a bound on the dynamic regret of TIRSO as a function of the path length under assumptions A1-3:

$$\tilde{R}_d^{(n)}[T] \leq \frac{1}{\alpha\beta_{\tilde{\ell}}} \left((1 + L/\beta_{\tilde{\ell}}) \sqrt{PN} B_y + \lambda N \right) \left(\|\tilde{\boldsymbol{a}}_n^\circ[P]\|_2 + W^{(n)}[T] \right), \tag{4.16}$$

where $\alpha \in (0, 1/L]$ and $\|\tilde{\boldsymbol{a}}_n^\circ[t] - \tilde{\boldsymbol{a}}_n^\circ[t-1]\|_2 \leq \sigma,\ \forall\, t \geq P+1$. If the path length $W^{(n)}[T]$ is sublinear in T, then the dynamic regret is also sublinear in T. Moreover, the bound on the dynamic regret is a function of the path length, the parameters of the data, and the parameters of the algorithm such as initial value of the estimate and the step size.

4.4 Summary of the Chapter

- Two online algorithms are presented in [45] to estimate the VAR-based topologies under the framework of online learning.
- Sub-linear static regret bounds are established for both the proposed online algorithms. A logarithmic regret bound is also derived for TIRSO.
- A dynamic regret bound is also presented for TIRSO. This means that TIRSO can track the changes in time-varying topologies. This bound depends on path length, which characterizes how fast the changes occur in the topologies.
- The proposed algorithms are compared with the current benchmarks and extensive numerical tests are presented.
- The algorithms are tested over both synthetic data and real data.

Chapter 5

Topology Identification in Dynamic Structural Equation Modeling

This chapter summarizes Paper C ([37]).

5.1 Motivation

Spatio-temporal data coming from many complex systems often reflect the dynamics of an underlying physical, information, or technological network of the underlying observed systems. Identifying the structure or topology of this network is a well-motivated problem. This chapter considers the problem of identifying a directed topology of an underlying network under the framework of structural equation modeling (SEM) [24]. This is a powerful model due to: a) accommodation of the exogenous variables (the variables which are not influenced by the endogenous variables) in the model, allowing to represent different possible applications, and b) its tractability. The static SEM model has been used to investigate the problem of directed topology identification in several applications [52, 68]. However, in many cases, the underlying topology can be time-varying. To this end, dynamic SEM can be used [69]. In this chapter, we investigate the estimation of sparse dynamic SEM based topologies.

5.2 Model and Problem Formulation

The dynamic linear structural equation model (SEM) for a network with N nodes and C *contagions* is given by [35]:

$$y_{ic}^t = \sum_{j=1, j\neq i}^{N} a_{ij}^t y_{jc}^t + b_{ii}^t x_{ic} + e_{ic}^t, \tag{5.1}$$

where y_{ic}^t denotes the intensity of the c-th contagion in node i at time t and x_{ic} represents the susceptibility of node i to external influence by contagion c. The coefficients a_{ij}^t are

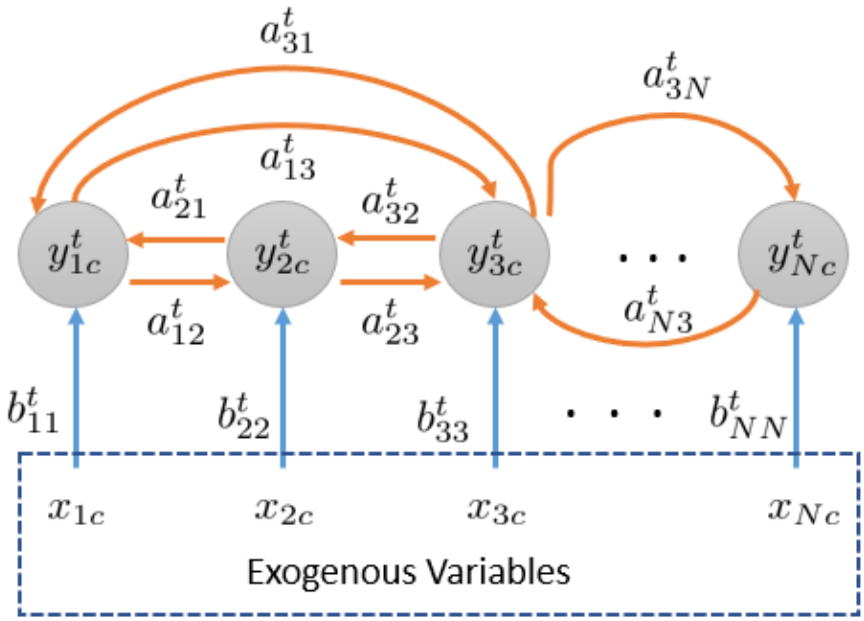


Figure 5.1: A digram of the dynamic SEM.

the time-varying SEM parameters that encode the topology of the network, b_{ii}^t quantifies the level of influence of external sources on node i, and e_{ic} denotes the measurement errors and un-modeled dynamics. A pictorial representation of the dynamic SEM for the t-th time instant is presented in Fig. 5.1.

By defining $\boldsymbol{y}_c^t = [y_{1c}^t, \ldots, y_{Nc}^t]^\top \in \mathbb{R}^N$, $\boldsymbol{x}_c = [x_{1c}, \ldots, x_{Nc}]^\top \in \mathbb{R}^N$, $\boldsymbol{B}^t = \text{diag}(\boldsymbol{b}^t) \in \mathbb{R}^{N\times N}$ with $\boldsymbol{b}^t = [b_{11}^t, \ldots, b_{NN}^t]^\top$, and $\boldsymbol{e}_c^t = [e_{1c}^t, \ldots, e_{Nc}^t]^\top \in \mathbb{R}^N$, the model in (5.1) can also be written in a compact form as:

$$\boldsymbol{y}_c^t = \boldsymbol{A}^t \boldsymbol{y}_c^t + \boldsymbol{B}^t \boldsymbol{x}_c + \boldsymbol{e}_c^t, \qquad c = 1, \ldots, C. \tag{5.2}$$

The matrix $\boldsymbol{A}^t \in \mathbb{R}^{N\times N}$ can be seen as a time-varying adjacency matrix for a SEM-based network. Note that $\boldsymbol{A}^t$ and $\boldsymbol{B}^t$ are the same for all the contagions. The observations for all contagions can be collected in a matrix by defining $\boldsymbol{Y}^t = [\boldsymbol{y}_1^t, \ldots, \boldsymbol{y}_C^t] \in \mathbb{R}^{N\times C}$, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C] \in \mathbb{R}^{N\times C}$, and $\boldsymbol{E}^t = [\boldsymbol{e}_1^t, \ldots, \boldsymbol{e}_C^t] \in \mathbb{R}^{N\times C}$. The dynamic SEM takes the following form:

$$\boldsymbol{Y}^t = \boldsymbol{A}^t \boldsymbol{Y}^t + \boldsymbol{B}^t \boldsymbol{X} + \boldsymbol{E}^t. \tag{5.3}$$

The goal is to track the time-varying SEM-based topologies. The problem statement is: given the observations $\{\boldsymbol{Y}^t\}_{t=1}^T$ and $\boldsymbol{X}$, find $\{\boldsymbol{A}^t\}_{t=1}^T$ and $\{\boldsymbol{B}^t\}_{t=1}^T$. We formulate the estimation problem as

$$\{\hat{\boldsymbol{A}}^t, \hat{\boldsymbol{B}}^t\} = \underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min}\ f_t(\boldsymbol{A}, \boldsymbol{B}) + \Omega(\boldsymbol{A}) \tag{5.4a}$$

$$\text{s.t. } a_{ii} = 0, \forall i \tag{5.4b}$$

$$b_{ij} = 0, \forall i \neq j. \tag{5.4c}$$

The first term in the above criterion is a data-fitting function while the second term is a sparsity-promoting regularization term. The constraints ensure that a valid $\boldsymbol{A}$ and

$\boldsymbol{B}$ are estimated. Specifically, the constraint $a_{ii} = 0$ eliminates any component of the trivial solution $\boldsymbol{A} = \boldsymbol{I}$. The constraint $b_{ij} = 0$ guarantees a diagonal $\boldsymbol{B}$, meaning that external sources for a certain node i do not affect any other node $j \neq i$. Following [35], we consider the exponentially-weighted least-squares criterion, i.e., $f_t(\boldsymbol{A}, \boldsymbol{B}) \triangleq \frac{1}{2}\sum_{\tau=1}^{t} \gamma^{t-\tau} \left\|\boldsymbol{Y}^{\tau} - \boldsymbol{A}\boldsymbol{Y}^{\tau} - \boldsymbol{B}\boldsymbol{X}\right\|_F^2$ and the regularizer $\Omega(\boldsymbol{A}) \triangleq \lambda \left\|\mathrm{vec}(\boldsymbol{A})\right\|_1$. The parameter $\gamma \in (0, 1]$ is a forgetting factor that regulates how much past information influences the solution at time t, and λ is the sparsity-promoting regularization parameter.

5.3 The Proposed Algorithm and its Dynamic Regret Analysis

The work in [37] proposes an online algorithm to track the time-varying SEM-based sparse topologies. An exponentially-weighted least-square data-fitting function is considered along with a regularization term to enforce sparse topologies. The resulting problem is solved by proposing an online algorithm based on proximal online gradient descent under the framework of online convex optimization. The proximal online gradient algorithm in [70] is adopted to solve (5.4) leveraging the separability of the problem. The proposed algorithm is tabulated in **Algorithm 3**.

Algorithm 3 Online algorithm for tracking dynamic SEM-based Topologies

Input: $\gamma, \lambda, \alpha \in (0, 1/L_f], \{\boldsymbol{Y}^t\}_{t=1}^T, \boldsymbol{X}$
Output: $\{\boldsymbol{A}[t]\}_{t=1}^T$, $\{\boldsymbol{B}[t]\}_{t=1}^T$
Initialization:
$\boldsymbol{v}_i[1] = \boldsymbol{0}_{N\times 1}, \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^0 = \boldsymbol{0}_{N\times N}, \boldsymbol{r}_i^0 = \boldsymbol{0}_{N\times 1},\ i = 1, \ldots, N$

for $t = 1, 2, \ldots, T$ **do**
  Receive data $\boldsymbol{Y}^t$
  for $i = 1, 2, \ldots, N$ **do**
    $\boldsymbol{Z}_i^t = [(\boldsymbol{Y}_{-i}^t)^\top\ (\boldsymbol{x}_i^\top)^\top]^\top$
    $\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t = \gamma\, \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^{t-1} + \boldsymbol{Z}_i(\boldsymbol{Z}_i)^\top$
    $\boldsymbol{r}_i^t = \gamma\, \boldsymbol{r}_i^{t-1} + \boldsymbol{Z}_i^t(\boldsymbol{y}_i^{t\top})^\top$
    $\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) = \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t \boldsymbol{v}_i[t] - \boldsymbol{r}_i^t$
    $\boldsymbol{v}_i^{\mathrm{f}}[t] = \boldsymbol{v}_i[t] - \alpha \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t])$
    $\boldsymbol{v}_i[t+1] = \mathbf{prox}_{\Omega^i}^{\alpha}\left(\boldsymbol{v}_i^{\mathrm{f}}[t]\right)$
  end for
  Form $\boldsymbol{A}[t]$ and $\boldsymbol{B}[t]$ from $\boldsymbol{v}_i[t], i = 1, ..., N$
end for

To characterize the tracking performance of the proposed online algorithm, we analyze its dynamic regret [56]. It is established in Theorem 1 of [37] that the dynamic regret is a function of the path length and parameters of data and algorithm. If the path length is sublinear, the upper bound on the dynamic regret becomes sublinear.

The numerical results presented in the paper show that the proposed algorithm can track the changes in the time-varying SEM-based topologies. Two different models are

shown to characterize the variations in the topologies. Specifically, the normalized mean squared error (NMSE) and the dynamic regret of the proposed algorithm is analyzed for a smooth-transition model and a non-smooth transition model.

5.4 Summary of the Chapter

- Tracking dynamic SEM-based topologies for streaming data is an important and a well-motivated problem.
- Our paper [37] proposes an online algorithm to estimate the time-varying topologies in dynamic environments.
- A dynamic regret analysis for the proposed algorithm is presented in [37]. A bound on the dynamic regret is derived. This bound depends on parameters related to the data, the algorithm, and the path length.
- Numerical results are presented that show that the proposed online algorithm can track the time-varying topologies in the case of streaming data.

Chapter 6

Dynamic Topology and Breakpoint Identification in Non-stationary Vector Autoregressive Processes

This chapter summarizes Paper D ([47]).

6.1 Motivation

To deal with time-varying topologies, graphical models [38, 39, 14, 34] or structural equation models [26, 40] can be used. However, the above models only capture memoryless interactions, which limits their applicability to many real-world scenarios. To this end, topology inference from multiple time series is usually addressed via vector autoregressive (VAR) models [43]. To cope with non-stationarity, VAR coefficients are assumed to evolve smoothly over time [71, 72, 73, 45], to vary according to a hidden Markov model [74], or to remain constant over time intervals separated by *structural breakpoints* [75, 76, 77, 78, 79, 42, 80]. However, these methods cannot handle rapid variations in the topology. To this end, we propose an algorithm in [47] that can detect breakpoints for VAR-based topologies.

6.2 Dynamic Topology Identification

A customary model for multivariate time series generated by non-stationary dynamic systems is the so-called P-th order TVAR model [43, Ch. 1]:

$$\boldsymbol{y}[t] = \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)} \boldsymbol{y}[t-p] + \boldsymbol{u}_t, \tag{6.1}$$

Algorithm 4 ADMM solver for dynamic network identification

Input: λ, γ, data $\{\boldsymbol{y}_t\}_{t=1}^T$

Output: matrix $\mathbf{B}$ containing VAR coefficients

```
for k = 1, ... until convergence do
    Update B_t via (D.9a)
    for t ∈ [L+1, T] do
        for (i, j) ∈ [1, P]^2 do
            Update c_{ij,t}, θ_{ij,t-1} via (D.9b,D.9c)
        end for
    end for
    Update U, V via (D.9d,D.9e)
end for
```

where the matrix entries $\{a_{ij,t}^{(\ell)}\}_{i,j\in[1,P],t\in[1,T]}$ are the model coefficients and $\boldsymbol{u}_{i,t}$ form the innovation process. Throughout, the notation $[m,n]$ with m and n integers satisfying $m \leq n$ will stand for $\{m, m+1, \ldots, n\}$. A time-invariant VAR model is a special case of (6.1) where $a_{ij,t}^{(\ell)} = a_{ij,t'}^{(\ell)}\ \forall(t,t')$.

An insightful interpretation of time-varying VAR models stems from expressing (6.1) as

$$y_{i,t} = \textstyle\sum_{\ell=1}^{L}\sum_{j=1}^{P} a_{ij,t}^{(\ell)} y_{j,t-\ell} + \boldsymbol{u}_{i,t} \tag{6.2a}$$

$$= \sum_{j=1}^{P} [y_{j,t-1}, y_{j,t-2}, \ldots, y_{j,t-L}]\ \boldsymbol{a}_{ij,t} + \boldsymbol{u}_{i,t} \tag{6.2b}$$

where $\boldsymbol{a}_{ij,t} := [a_{ij,t}^{(1)}, a_{ij,t}^{(2)}, \ldots, a_{ij,t}^{(L)}]^\top$. From (6.2a), the i-th sequence $\{y_{i,t}\}_{t=1}^T$ equals the innovation plus the sum of all sequences $\{\{y_{p,t}\}_{t=1}^T\}_{p=1}^P$ after being filtered with a *linear time-varying* (LTV) filter with coefficients $\{a_{ij,t}^{(l)}\}_{l=1}^L$.

As described in the introduction of this Dissertation, interactions between time series are generally indirect (unmediated), which translates into many of these LTV filters being identically zero. To mathematically capture this interaction pattern, previous works consider the notion of graph associated with a time-invariant VAR process (see e.g. [66]), which is generalized next to *time-varying* VAR models (6.1). To this end, identify the i-th time series with the i-th vertex (or node) in the vertex set $\mathcal{V} := [1,P]$ and define the time-varying edge set as $\mathcal{E}_t := \{(i,j) \in \mathcal{V}\times\mathcal{V} : \boldsymbol{a}_{ij,t} \neq \mathbf{0}\}$. Thus, each edge of this time-varying graph can be thought of as an LTV filter, as depicted in Fig. 6.1.

The main goal of this paper is to estimate $\{\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P\}_{t=P+1}^T$ given $\{\boldsymbol{y}[t]\}_{t=1}^T$. To cope with the issue of identifiability, we impose certain structure usually found in real-world dynamic systems. The proposed estimation criterion is given by

$$\min_{\{\boldsymbol{A}_p^{(t)}\}} \sum_{t=L+1}^{T} \left\| \boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)} \boldsymbol{y}[t-p] \right\|_2^2 \tag{6.3}$$
$$+ \sum_{(i,j)} \left(\lambda \sum_{t=P+1}^{T} \|\boldsymbol{a}_{ij,t}\|_2 + \gamma \sum_{t=P+2}^{T} \|\boldsymbol{a}_{ij,t} - \boldsymbol{a}_{ij,t-1}\|_2 \right).$$

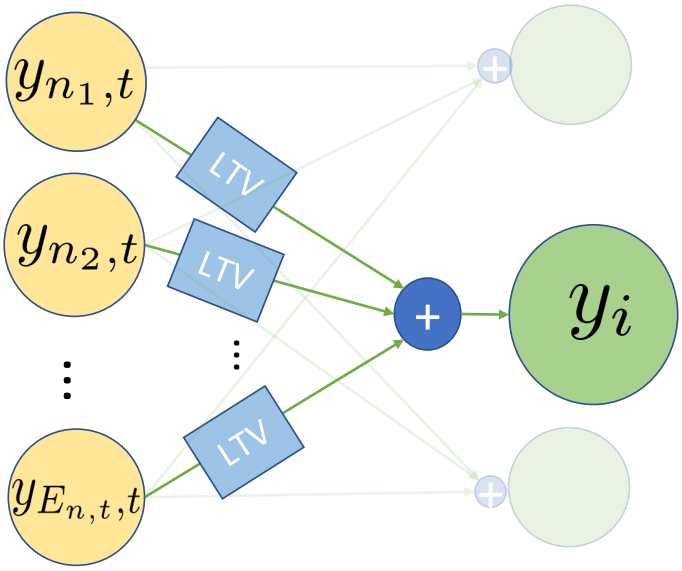


Figure 6.1: Graph associated with a TVAR model.

The regularization parameters $\lambda > 0$ and $\gamma > 0$ can be selected through cross-validation to balance the relative weight of data and prior information. The first regularizer is a group-lasso penalty that promotes edge sparsity. This corresponds to the intuitive notion that most interactions in a complex network are indirect and therefore nodes are connected only with a small fraction of other nodes. This regularizer generalizes the one in [66] to time-varying graphs.

The second regularizer promotes estimates where the edges remain constant over time except for a relatively small number of time instants $\mathcal{T}_{i,j} := \{t : a_{ij,t}^{(p)} \neq a_{ij,t-1}^{(p)} \text{ for some } p\}$ denoted as *local breakpoints*. This variant of total-variation regularizer, together with the notion of local breakpoints, constitutes one of the major novelties of this work and contrasts with the notion of structural (or global) breakpoints, defined as $\mathcal{T} := \{t : \boldsymbol{A}_p^{(t)} \neq \boldsymbol{A}_p^{(t-1)} \text{ for some } p\}$ and adopted in the literature [79, 42, 75, 76, 78]. These works promote solutions with few global breakpoints, and therefore all the edges estimates change simultaneously at the same time for all nodes. In contrast, this work advocates promoting solutions with a few *local* breakpoints, since it is expected that changes in the underlying dynamic system take place locally.

In practice, the time series are expected to evolve at a faster time scale than the underlying system that generates them. The sampling rate needs to be increased if this does not hold. Therefore, it is beneficial to assume that $\boldsymbol{A}_p^{(t)}$ remain constant within a certain window since this would decrease the number of coefficients to estimate and therefore would improve estimation performance.

Next, an ADMM based algorithm is proposed to solve (6.3). The algorithm is detailed here in **Algorithm 4**. The numerical results in the paper show that the algorithm can

detect the local breakpoint more accurately as compared to a competing algorithm.

6.3 Summary of the Chapter

- The problem of breakpoint detection is considered in non-stationary VAR-based topology identification.
- For piece-wise stationary VAR-based topologies, the concept of local break-point is introduced, motivated by the fact that only a few edges of the dynamic topology can change.
- A low computational complexity algorithm is proposed, which can detect local breakpoint and can identify sparse topologies.
- Numerical test based on synthetic data demonstrate that the proposed algorithm outperforms the existing competing state-of-the-art methods.

Chapter 7

Online Joint Topology Identification and Signal Estimation with Inexact Proximal Online Gradient Descent

This chapter summarizes Paper E ([48]).

7.1 Introduction

In the previous chapters, we have discussed the problem of topology identification problems in the settings where the data has no missing values. However, in many scenarios, we have noisy observations with missing values [81]. For instance, the missing values may occur due to faulty sensors in sensor networks or due to burst sensing with some periods where sensors take measurements and other periods where there are no measurements [82]. In social networks, the users may be reluctant to share certain information. The source of the observation noise may be due to the accuracy of sensors. A simplified diagram with missing values is presented in Fig. 7.1.

A batch approach to the problem of undirected topology identification under an incomplete data scenario is considered in [83]. For directed graphs, the works in [84] and [85] address the batch estimation of the VAR parameters in the presence of missing and noisy data. In online approaches, online time series prediction for missing data is considered in [86]. A similar problem is presented in [87]. However, these works adopt a one-dimensional autoregressive (AR) process. An approach to jointly estimate the signal and topology with missing data is presented in [88]. Different batch and online algorithms are proposed. However, the proposed online algorithm is not supported by convergence guarantees for online scenarios such as the dynamic regret analysis.

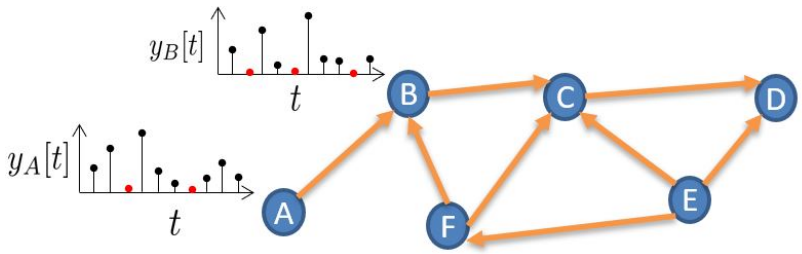


Figure 7.1: A simplified network with missing values in time series. The red dots represents the missing values in the time series.

To address the problem of topology of identification in time-varying scenarios with missing data, we propose two novel online algorithms, i.e., Joint Signal and Topology Identification via Recursive Sparse Online optimization (JSTIRSO) and Joint Signal and Topology Identification via Sparse Online optimization (JSTISO) in [48]. To analyze theoretically the tracking performance of JSTIRSO time-varying topology settings, a bound on the dynamic regret is derived.

7.2 Problem Formulation

A time-varying model given in (6.1) is used in this work. In order to provide the optimization framework, let us first consider the batch version of the optimization problem. The problem statement in the case of batch estimation is: given the observations $\boldsymbol{y}[t]$, $t = 0, \ldots, T-1$ and the VAR process order P, find the time-varying VAR coefficients $\{\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\}_{t=P}^{T-1}$ such that it yield sparse topology at each time instant. To formulate the problem of estimating the causality graphs with missing values and noise in the observation vector, consider a subset of $\mathcal{V}$ where the signal is observed, given by $\mathcal{M}_t \subseteq \mathcal{V}$. The (random) pattern of missing values is collected in the N-by-N diagonal matrix $\boldsymbol{M}_t$ where $M_{nn}[t], n = 1, \ldots, N$, are i.i.d. Bernoulli random variables taking value 1 with probability ρ and zero with probability $1-\rho$. $\boldsymbol{M}_t$ is a diagonal matrix with the n-th diagonal entry being zero whenever the value at the n-th node is missing, otherwise one. Let $\tilde{\boldsymbol{y}}[t]$ be the observation obtained at time t, given by

$$\tilde{\boldsymbol{y}}[t] = \boldsymbol{M}_t \boldsymbol{y}[t] + \boldsymbol{M}_t \boldsymbol{\epsilon}[t], \tag{7.1}$$

where $\boldsymbol{\epsilon}[t]$ is the observation noise vector. In the batch setting, the problem of estimating time-varying topologies with missing values is: given the noisy observations $\{\tilde{\boldsymbol{y}}[t]\}_{t=0}^{T-1}$ with missing values and the VAR process order P, find the coefficients $\{\{\hat{\mathbf{A}}_p^{(t)}\}_{p=1}^{P}\}_{t=P}^{T-1}$ such that it yield a sparse topology. However, it is easier to estimate the topology from the observation vector directly if the missing values are reconstructed (imputed), and the topology (VAR parameters) helps in such reconstruction. Thus, a natural approach is to jointly estimate the signal and the topology.
In batch setting, a common approach is to solve the following problem which includes

joint estimation of the signal and the VAR coefficients:

$$\left\{\hat{\boldsymbol{y}}[t], \left\{\hat{\mathbf{A}}_p^{(t)}\right\}_{p=1}^{P}\right\}_{t=P}^{T-1} = \underset{\left\{\boldsymbol{y}[t], \{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\right\}_{t=P}^{T-1}}{\arg\min} \frac{1}{2}\sum_{t=P}^{T-1}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P}\boldsymbol{A}_p^{(t)}\,\boldsymbol{y}[t-p]\right\|_2^2$$
$$+\frac{\nu}{2|\mathcal{M}_t|}\sum_{t=P}^{T-1}\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t\boldsymbol{y}[t]\|_2^2 + \lambda\sum_{t=P}^{T-1}\sum_{n=1}^{N}\sum_{n'=1}^{N}\mathbb{1}\{n' \neq n\}\left\|\boldsymbol{a}_{n,n'}^{(t)}\right\|_2 + \beta\sum_{t=P}^{T-1}\sum_{p=1}^{P}\|\boldsymbol{A}_p^{(t)} - \boldsymbol{A}_p^{(t-1)}\|_{\mathrm{F}}^2, \tag{7.2}$$

where the first term is a least-squares (LS) fitting error for all time instants (where the t-th term in the summation fits the signal based on the P previous observations and the VAR coefficients at time t), the second term penalizes the mismatch between the observation vector and the reconstructed signal (recall that $|\mathcal{M}_t|$ is the number of nodes where the signal is observed), the third term is a regularization function that promotes sparsity in the edges, and the fourth term limits the variations in the coefficients. The parameter $\nu > 0$ is a constant to control the trade-off between the prediction error based on the VAR coefficients and the mismatch between the measured samples and the signal reported after the reconstruction. The parameter λ controls the sparsity in the edges while β controls the magnitude of the cumulative norm of the difference between consecutive coefficients. The resulting problem in (7.2) is (separately) convex in $\{\boldsymbol{y}[t]\}_{t=P}^{T-1}$ and in $\{\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\}_{t=P}^{T-1}\}$, but not jointly convex. The problem in (7.2) can be solved via alternating minimization. Each problem in alternating minimization can be solved via proximal gradient descent.

7.3 Proposed Online Solutions

The batch formulation in (7.2) uses information from all time instants to produce a sequence of reconstructed signal values and VAR parameter (topology) estimates. On the other hand, an online formulation should allow us to produce such a sequence with minimum delay and with fixed complexity (at the price of lower accuracy). Specifically, here we are interested in an algorithm that, at each time instant t, produces an estimate of $\boldsymbol{y}[t]$ and$\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}$ as soon as the partial observation $\tilde{\boldsymbol{y}[t]}$ is received.

The problem of estimating time-varying topology with missing data in the online setting is posed as follows: at each time instant t, given the noisy observations $\tilde{\boldsymbol{y}}[t]$ with missing values, the previous estimate $\{\hat{\mathbf{A}}_p^{(t)}\}_{p=1}^{P}$, and the VAR process order P, find the coefficients $\{\hat{\mathbf{A}}_p^{(t+1)}\}_{p=1}^{P}$ such that it yields a sparse topology. However, as opposed to the approaches in the previous Chapters, we cannot only estimate the topology from the observation vector directly since it may have missing values. A natural approach is to jointly estimate signal and the topology.

To this end, we design an online criterion such that its sum over time matches the

batch objective in (7.2). Consider the following dynamic cost function:

$$c_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}, \left\{\boldsymbol{A}_p^{(t)}, \boldsymbol{A}_p^{(t-1)}\right\}_{p=1}^{P}\right) \triangleq \ell_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}, \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \beta \sum_{t=P}^{T-1}\sum_{p=1}^{P} \|\boldsymbol{A}_p^{(t)} - \boldsymbol{A}_p^{(t-1)}\|_{\mathrm{F}}^2, \quad (7.3)$$

where

$$\ell_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}, \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \triangleq \frac{1}{2}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \boldsymbol{y}[t-p]\right\|_2^2 + \frac{\nu}{2|\mathcal{M}_t|}\, \|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t \boldsymbol{y}[t]\|_2^2, \quad (7.4)$$

and

$$\Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \triangleq \lambda \sum_{n=1}^{N}\sum_{n'=1}^{N} \mathbb{1}\{n' \neq n\} \left\|\boldsymbol{a}_{n,n'}^{(t)}\right\|_2, \quad (7.5)$$

where $\boldsymbol{a}_{n,n'}^{(t)}$ has the same structure of $\boldsymbol{a}_{n,n'}$ with time-varying VAR parameters. With these definitions, the objective function in (7.2) can be rewritten as $\sum_t c_t(\dots)$. It becomes clear that producing an estimate of $\boldsymbol{y}[t]$ and $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}$ does not only have an impact on $c_t(\cdot)$, but also on $\{c_\tau(\cdot)\}_{\tau=t}^{t+P}$. Such a coupling in time is taken into account in the framework of dynamic programming (or reinforcement learning), where the goal is to find a policy π of the form

$$\begin{aligned} \pi &: \mathbb{R}^{PN} \times \mathbb{R}^{N^2P} \times \mathbb{R}^{N} \times \mathbb{R}^{N^2} && \to \quad \mathbb{R}^{N} \times \mathbb{R}^{N^2P} \\ \pi &\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \left\{\hat{\boldsymbol{A}}_p^{(t-1)}\right\}_{p=1}^{P}, \tilde{\boldsymbol{y}}[t], \boldsymbol{M}_t\right) && \rightsquigarrow \quad \hat{\boldsymbol{y}}[t], \left\{\hat{\boldsymbol{A}}_p^{(t)}\right\}_{p=1}^{P} \end{aligned} \quad (7.6)$$

such that the cumulative cost is minimized in expectation. Learning such a policy (via, e.g., deep reinforcement learning) would require a high amount of computation, and it is out of the scope of the present work. Instead, we propose to approximate such a policy using the much more tractable framework of online convex optimization. Fortunately enough, the structure of (7.3) resembles that of the composite problems that can be efficiently dealt with via proximal online gradient descent (OGD). In the next paragraph, we will explain the approximations we take in order to be able to apply a variant of proximal OGD to the online problem at hand.

Our approach consists in treating, at time t, the P previous reconstructed samples, $\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}$, as random variables. Although those variables are dependent of the estimated VAR parameters, we adopt the simplifying approximation of assuming that they are *independent*. After doing so, the deterministic function $c_t(\cdot)$ is replaced with a *random* function

$$C_t\left(\boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) = \ell_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \beta \sum_{p=1}^{P} \|\boldsymbol{A}_p^{(t)} - \hat{\boldsymbol{A}}_p^{(t-1)}\|_{\mathrm{F}}^2, \quad (7.7)$$

which is jointly convex in its arguments. Notice that, if $\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}$ and $\tilde{\boldsymbol{y}}[t]$ were equal to the true (unobservable) signals $\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}$, this setting would be the same that is dealt

with in [45], by direct application of proximal OGD. Since the aforementioned signal estimates are inexact versions of the true signals, in the present work we will use the inexact proximal OGD framework discussed in [70] to analyze the regret of the resulting algorithm.

Before proceeding to the formulation of the online algorithm, one more remark is in order. Notice that the cost function has as inputs the signal estimate and the VAR parameters. It is assumed that the VAR parameters change smoothly with time, but we cannot assume that the signals vary smoothly with time. Recall that in each proximal OGD iteration, a minimization is solved involving a first-order approximation of the loss ℓ_t, the (non-linearized) regularizer Ω, and a proximal term that ensures that the variable estimated at time t is close in norm to its previous estimate at time $t-1$. This proximal term should involve $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$, but not $\boldsymbol{y}[t]$.

Fortunately, the joint optimization over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$ and $\boldsymbol{y}[t]$ can be reformulated into an optimization only over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$ as follows. Since C_t is jointly convex in both of its arguments, minimizing it can be split into first minimizing over $\boldsymbol{y}$ and then over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$. Then, we can write

$$\mathcal{L}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right) \triangleq \min_{\boldsymbol{y}[t]} \ell_t\left(\{\hat{\boldsymbol{g}}[t], \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right), \tag{7.8}$$

where $\hat{\boldsymbol{g}}[t] \triangleq \text{vec}([\hat{\boldsymbol{y}}[t-1], \ldots, \hat{\boldsymbol{y}}[t-P]]^\top)$, and the minimization can be solved analytically. Once a closed form is available for $\mathcal{L}$, a composite objective online optimization algorithm (specifically proximal OGD) can be applied.

The signal reconstruction problem is solved as follows: Given the current data vector $\tilde{\boldsymbol{y}}[t]$, the masking matrix $\boldsymbol{M}_t$, the estimates of the previous P data vectors $\{\hat{\boldsymbol{y}}[t-p]\}_{p=1}^P$, and the VAR coefficients estimated at the previous time-instant $\{\hat{\mathbf{A}}_p^{(t)}\}_{p=1}^P$, the problem of estimating the signal $\boldsymbol{y}[t]$ becomes [88] :

$$\hat{\boldsymbol{y}}[t] = \arg\min_{\boldsymbol{y}[t]} \frac{1}{2}\left\|\boldsymbol{y}[t] - \sum_{p=1}^P \boldsymbol{A}_p^{(t)}\,\hat{\boldsymbol{y}}[t-p]\right\|_2^2 + \frac{\nu}{2|\mathcal{M}_t|}\left\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t\boldsymbol{y}[t]\right\|_2^2, \tag{7.9}$$

which is solved in closed-form by computing the gradient of its cost function with respect to $\boldsymbol{y}[t]$ and setting it to zero. The resulting solution for n-th value of $\boldsymbol{y}[t]$is given by:

$$\hat{y}_n[t] = (1 - U_n[t])\,\hat{\boldsymbol{g}}[t]^\top \boldsymbol{a}_n[t] + U_n[t]\tilde{y}_n[t], \tag{7.10}$$

where

$$U_n[t] \triangleq \frac{\nu M_{nn}[t]}{|\mathcal{M}_t| + \nu M_{nn}[t]}. \tag{7.11}$$

Observe that $U_n[t]$ is zeros when $y_n[t]$ is missing, otherwise $U_n[t]$ is $\nu/(|\mathcal{M}_t| + \nu)$. When $y_n[t]$ is present, $U_n[t]$ is always less than 1. The overall computational complexity for estimating $\hat{\boldsymbol{y}}[t]$ is $O(N^2P)$. This complexity can be reduced depending on the sparse structure of $\{\boldsymbol{a}_n[t]\}_{n=1}^N$. If, for instance, the number of edges is $\mathcal{O}(N)$, then the computational complexity for estimating $\hat{\boldsymbol{y}}[t]$ becomes $\mathcal{O}(NP)$ per t.

The loss function in (7.8) is derived in closed-form by substituting the solution from (7.10). Once the loss function is computed, we can apply inexact proximal OGD since

due to noisy observations with missing values, the loss function is inexact and so is the gradient. The proposed JSTISO algorithm is detailed in **Algorithm 13** in [48].

The loss function in JSTISO is an instantaneous loss, i.e., based only on the current sample. While this keeps the complexity of the iterations very low, and may be sufficient for online estimation of a static VAR model, it is sensitive to noise and input variability, and thus it is expected to perform poorly when attempting at tracking a time-varying model. In [45], a running average loss function is designed drawing inspiration from the relation between least mean squares (LMS) and recursive least squares (RLS) to improve the tracking capabilities of the algorithm that is derived based on an instantaneous loss function. In this work, we follow similar steps to propose a second approach, where a running average loss function is adopted, which depends on the past received signal values. In this second approach, we set the loss function as

$$\tilde{\ell}_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) = \frac{1}{2}\Bigg(\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[t-p]\right\|_2^2 + \sum_{\tau=P}^{t-1} \gamma^{t-\tau} \left\|\hat{\boldsymbol{y}}[\tau] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[\tau-p]\right\|_2^2\Bigg) + \frac{\nu}{2|\mathcal{M}_t|}\left\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t \boldsymbol{y}[t]\right\|_2^2, \quad (7.12)$$

where γ is a user-selected forgetting factor which controls the weight of past (reconstructed) samples of $\boldsymbol{y}[t]$. The procedure in the previous section (treating the previously reconstructed samples as a random variable, and minimizing over $\boldsymbol{y}[t]$) is applied to the alternative deterministic loss $\tilde{\ell}_t$, so we can write the random loss function $\tilde{\mathcal{L}}_t$ as

$$\tilde{\mathcal{L}}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \triangleq \min_{\boldsymbol{y}[t]} \tilde{\ell}_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right). \quad (7.13)$$

Next, we follow the same steps as in the previous case for the signal reconstruction. It turns out that the signal reconstruction in this case coincides with the reconstruction problem in (7.9). Therefore, to derive the closed-form solution for the loss function, we substitute the solution of $\hat{\boldsymbol{y}}[t]$ given by (7.10) into (7.13). Next, inexact proximal OGD is adopted to derive JSTIRSO, tabulated in **Algorithm 5**.

7.4 Dynamic Regret Analysis

The dynamic regret analysis is presented for the proposed algorithm JSTIRSO. In Theorem 7 of [48], the following bound on the dynamic regret is derived for the proposed algorithm:

$$\tilde{R}_d^{(n)}[T] \leq \mathcal{O}\left(1 + W^{(n)}[T] + E^{(n)}[T]\right), \quad (7.14)$$

where

$$E^{(n)}[T] \triangleq \sum_{t=P}^{T} \left[\|\boldsymbol{e}^{(n)}[t]\|_2\right]. \quad (7.15)$$

The dynamic regret is a function the path length and the cumulative error in the gradient. If these two are sublinear, then the dynamic regret becomes sublinear.

Algorithm 5 Tracking time-varying topologies with missing data via JSTIRSO

Input: $\nu, \gamma, P, \lambda, \sigma^2, \alpha, \{\hat{\boldsymbol{y}}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\tilde{\boldsymbol{a}}_n[t]\}_{n=1}^{N}$
Initialization: $\tilde{\boldsymbol{a}}_n[P] = \mathbf{0}, n = 1, \ldots, N, \hat{\boldsymbol{\Phi}}[P-1] = \sigma^2 \boldsymbol{I}$,
$\hat{\boldsymbol{r}}_n[t] = \mathbf{0}, n = 1, \ldots, N$

for $t = P, P+1, \ldots$ **do**
  Receive noisy data vector with missing values $\tilde{\boldsymbol{y}}[t]$
  $\hat{\boldsymbol{g}}[t] = \mathrm{vec}\Big(\big[\hat{\boldsymbol{y}}[t-1], \ldots, \hat{\boldsymbol{y}}[t-P]\big]^\top\Big)$
  $\hat{\boldsymbol{\Phi}}[t] = \gamma\, \hat{\boldsymbol{\Phi}}[t-1] + \hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]$
  for $n = 1, \ldots, N$ **do**
    Compute $\hat{y}_n[t]$ from $\tilde{y}_n[t]$ via (7.10)
    $\hat{\boldsymbol{r}}_n[t] = \gamma\, \hat{\boldsymbol{r}}_n[t-1] + \tilde{y}_n[t]\, \hat{\boldsymbol{g}}[t]$
    $U_n[t] = \frac{\nu M_{nn}[t]}{|\mathcal{M}_t| + \nu M_{nn}[t]}$
    $\hat{\boldsymbol{v}}_n[t] = U_n[t]\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\tilde{\boldsymbol{a}}_n[t] - \tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right) + \hat{\boldsymbol{\Phi}}[t-1]\tilde{\boldsymbol{a}}_n[t] - \hat{\boldsymbol{r}}_n[t-1]$
    for $n' = 1, 2, \ldots, N$ **do**
      $\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t] = \tilde{\boldsymbol{a}}_{n,n'}[t] - \alpha \hat{\boldsymbol{v}}_{n,n'}[t]$
      $\tilde{\boldsymbol{a}}_{n,n'}[t+1] = \tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t] \left[1 - \frac{\alpha\lambda\ \mathbb{1}\{n \neq n'\}}{\left\|\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\right\|_2}\right]_+$
    end for
    $\tilde{\boldsymbol{a}}_n[t+1] = \big[\tilde{\boldsymbol{a}}^\top_{n,1}[t+1], \ldots, \tilde{\boldsymbol{a}}^\top_{n,N}[t+1]\big]^\top$
  end for
  Output $\{\tilde{\boldsymbol{a}}_n[t+1]\}_{n=1}^{N}$
end for

7.5 Summary of the Chapter

- This chapter summarizes [48], where the problem of online tracking time-varying topologies with missing data is considered.
- Two online algorithms are proposed in [48], where the problem is solved via a joint signal and topology estimation approach. At each step of the online algorithms, the signal is estimated from the noisy observations with missing values. Then, the time-varying topology is estimated. The resulting algorithms are called JSTISO and JSTIRSO.
- The dynamic regret bound for the JSTIRSO is derived. The bound is a function of the path length, error in the gradient, parameters of the data, and parameters of the algorithm.
- Simulations results show that both the proposed algorithms can track the changes in time-varying topologies in the presence of missing values in the data.

Chapter 8

Concluding Remarks

8.1 Conclusions

This dissertation addresses the problem of topology identification using online machine learning algorithms. Different practical settings for topology identification are envisaged under different models and online problems of topology estimation are formulated. To solve these problems, various online approaches are proposed and their performance is evaluated theoretically by deriving convergence guarantees and numerically by simulation tests in this dissertation. Mainly time-varying environments, where the data is sequentially available, are studied and online algorithms under the framework of online convex optimization are proposed. The bounds on the static and the dynamic regret of the proposed algorithms are derived. The following are the some of the worth-mentioning conclusions of this dissertation:

- Online algorithms for sparse topology identification are proposed for static settings when the data is coming sequentially. The numerical results confirm that the proposed algorithms converges to the batch solution.

- When the underlying topologies are static or slowly time-varying and the data is coming is a streaming fashion, two algorithms namely TISO and TIRSO under the framework of online optimization are proposed to identify the sparse topologies in an online fashion. Regret analysis is presented for both the algorithms. First, both the algorithms are proved to incur sublinear regret. Second, due to the strongly convex loss function considered in TIRSO, a logarithmic regret bound is derived. Finally, it is shown that TIRSO can also work in time-varying scenarios by deriving the dynamic regret bound, which depends on path length and the parameters of the data and the algorithm.

- To address the problem of tracking SEM-based topologies, an online algorithm for tracking the time-varying SEM-based sparse topologies is proposed. The performance guarantee in the form of dynamic regret is presented. Specifically, a dynamic regret bound for the proposed algorithm is derived.

- When the topologies are sparse and time-varying and the variations are not smooth,

all the edges of the graph may not change at once, therefore a concept of local structural breakpoint is introduced. An algorithm is proposed which is capable of detecting the breakpoint as well as the weights of the edges.

- The problem of dynamic topology identification in streaming signals when the noisy observations contain missing values is studied. Online algorithms, i.e., JSTISO and JSTIRSO are proposed for the problem. The dynamic regret bound for JSTIRSO is derived. The regret bound is a function of the parameters of the data, the path length, the error in the gradient, and the parameters of the algorithm such initial estimates and the step size.

8.2 Future Work

A number of different problems and settings have been considered in this dissertation. There are various directions to extend the work in this dissertation. Among them, the following are some of the possible potential future work directions:

- Nonlinear models for online topology identification preferably with constant memory and constant computational complexity can be a possible future research direction for this work.

- Considering other time-varying VAR models explicitly modeling the variations in the VAR coefficients, possibly along the lines of [89, 90, 91] and deriving the online algorithms can be a potential extension of this work.

- An interesting future work direction is to study the problem of identifying topologies whose adjacency matrix has a low-rank plus sparse structure along the lines of [92] to account for clusters in the nodes.

- The proposed algorithm in [47] is a batch method. Although challenging but a potential extension would be to derive an online algorithm to detect the breakpoints when the data is streaming.

- The algorithms in this dissertation are centralized. A possible extension would be to consider the problem of topology identification under the framework of distributed time-varying online optimization, e.g., along the lines of [93].

- Hyper-parameter tuning is a challenge in online settings. Hyper-parameter free online learning based approaches for topology identification can be studied along the lines of e.g., [94], [95].

- There are some works based on deep learning in the area of forecasting time series. A possible future work is to consider applying deep learning approaches to predict the time series similar to [96].

Appendices

Appendix A

Paper A

Title:	Online Topology Estimation for Vector Autoregressive Processes in Data Networks
Authors:	**Bakht Zaman**, Luis M. Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano
Affiliation:	Center Intelligent Signal Processing and Wireless Networks (WISENET) Department of ICT, University of Agder, Grimstad, Norway
Conference:	Proc. IEEE Int. Workshop Comput. Advances MultiSensor Adaptive Process., Curacao, NL, Dec. 2017.

Online Topology Estimation for Vector Autoregressive Processes in Data Networks

Bakht Zaman, Luis M. Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano

***Abstract*— An important problem in data sciences pertains to inferring causal interactions among a collection of time series. Upon modeling these as a vector autoregressive (VAR) process, this paper deals with estimating the model parameters to identify the underlying causality graph. To exploit the sparse connectivity of causality graphs, the proposed estimators minimize a group-Lasso regularized functional. To cope with real-time applications, big data setups, and possibly time-varying topologies, two online algorithms are presented to recover the sparse coefficients when observations are received sequentially. The proposed algorithms are inspired by the classic recursive least squares (RLS) algorithm and offer complementary benefits in terms of computational efficiency. Numerical results showcase the merits of the proposed schemes in both estimation and prediction tasks.**

A.1 Introduction

Network data analysis emerges naturally in a plethora of applications such as wireless sensor networks, transportation, social, and biological networks, to name a few. A prominent task in this context is inferring graphs that provide the causal relations among a collection of time series such as those encountered in econometrics and sensor data analysis. Identifying these causal interactions is a central problem in many disciplines such as neuroscience, econometrics, bio-informatics, meteorology. Revealing these interactions facilitates tasks such as prediction of time series and data completion.

The problem of inferring graphs capturing dependencies among variables has recently received a great attention in the literature. The simplest approach is to place an edge between two vertices if the sample correlation between the associated variables exceeds a threshold [13]. However, such an approach cannot distinguish mediated from unmediated interactions, thus motivating the methods of partial correlations [13], [14]. Since these methods are still unable to determine directionality in the dependencies, Granger proposed a means to infer the direction of causation by building upon the principle that the cause precedes the effect [28]. An alternative notion of interaction is adopted in the literature of structural equation models by incorporating the influence of exogenous variables; see e.g. [25],[26] and references therein. Unfortunately, these models do not generally capture the temporal structure present in time series. Further approaches for topology identification include [29, 30, 31] though their batch nature cannot track temporal changes in the topology.

The goal of this paper is to track the temporal dynamics of causal relations among time series associated with different variables. To this end, the framework of vector autoregressive (VAR) processes is invoked. These processes are extensively adopted to model linear dependencies among time series [43]. In a P-th order VAR model, the

current data are a noisy superposition of the data at the P previous time instants. The parameters of the VAR model reveal the topology of the causality graph, which motivates their estimation. An estimator based on minimizing a convex criterion regularized by a group-Lasso penalty is presented in [66] to estimate VAR parameters and hence the graph topology. This approach relies on the assumption that the connectivity is sparse, in the sense that the number of edges is small.

When the samples of the time series become available one by one, or when the size of the data challenges the available processing and memory capabilities, online estimation of the model parameters offers a great advantage compared to batch approaches as presented in [97], [98]. Online estimation is also advantageous when the data model is time-varying. Some authors have addressed estimation of time-varying AR models [89, 90, 91]. However, to the best of our knowledge, no online approach for tracking VAR parameters, and thus the associated network topology, has been considered in the literature.

This paper proposes two online estimators for the parameters of a VAR signal model to track the topology of the causality graph. Sparse estimates, where each time series is influenced by a small number of other time series, are enforced by means of a group-Lasso regularized objective in [66]. The first algorithm applies the approximate recursive least squares (RLS) approach in [99], whereas the second solves an optimization problem at each time instant by means of block coordinate descent (BCD). The complexity of these algorithms grows at different rates with the problem size (P and the number of time series), so that their benefits are complementary depending on the specific problem setting.

The contribution of this paper is twofold:

- Online estimation of the group-sparse parameters of VAR process by means of two algorithms with different orders of computational complexity.
- A performance comparison of the different approaches through numerical simulations.

The remainder of this paper is structured as follows: Section A.2 introduces the model and formulates the problem, and the proposed algorithms are presented in Section B.2.2. Section A.4 provides numerical tests and wraps up the paper.

A.2 Model and problem formulation

Consider a collection of N time series, where $f_n[t]$, $t = 0, 1, \ldots, T-1$, denotes the value of the n-th time series at time t. The goal is to determine a directed graph $\mathcal{G} \triangleq (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} = \{1, \ldots, N\}$ is the vertex set and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the edge set capturing the causation relations among time series. Specifically, $(n, n') \in \mathscr{E}$ iff $f_{n'}[t]$ causes $f_n[t+\tau]$ for some $\tau \in [1, P]$. To this end, the VAR model is adopted, which prescribes that

$$\boldsymbol{f}[t] \triangleq [f_1[t], \ldots, f_N[t]]^\top = \boldsymbol{u}[t] + \sum_{p=1}^{P} \mathbf{A}_p \boldsymbol{f}[t-p], \tag{A.1}$$

where $\boldsymbol{u}[t] \triangleq \left[u_1[t], u_2[t], ..., u_N[t]\right]^\top$ denotes noise and $\boldsymbol{A}_p \in \mathbb{R}^{N\times N}, p = 1, \ldots, P$, are the VAR parameters. From this expression, it follows that

$$f_n[t] = u_n[t] + \sum_{n'=1}^{N}\sum_{p=1}^{P} a_{n,n'}^{(p)} f_{n'}[t-p], \quad n = 1...N. \tag{A.2}$$

Then, if $f_{n'}[t]$ does not cause $f_n[t+\tau]$ for any $\tau \in [1, P]$, it holds that $a_{n,n'}^{(p)} = 0\ \forall p$, where $a_{n,n'}^{(p)}$ stands for the (n, n')-th entry of $\boldsymbol{A}_p$. This implies that (A.2) can be equivalently expressed as

$$f_n[t] = u_n[t] + \sum_{n':(n,n')\in\mathscr{E}}\sum_{p=1}^{P} a_{n,n'}^{(p)} f_{n'}[t-p], n = 1\ldots N. \tag{A.3}$$

Therefore, one can trivially obtain $\mathscr{E}$, and consequently $\mathcal{G}$, if $\{\boldsymbol{A}_p\}_{p=1}^P$ are known. From (E.2), it follows that $f_n[t]$ is the result of filtering the neighboring time series through a linear time-invariant (LTI) filter and adding these filtered signals together with noise. One can therefore interpret a sparse VAR model in terms of a graph whose edges correspond to an LTI filter between the adjacent vertices. The problem of topology identification reduces therefore to estimating $\{\boldsymbol{A}_p\}_{p=1}^P$ given $\{\boldsymbol{f}[t]\}_{t=0}^{T-1}$.

A.3 Online topology identification

After presenting the estimation criterion in Sec. A.3.0.1, this section describes the proposed algorithms in Secs. A.3.1 and A.3.2.

A.3.0.1 Estimation criterion

A natural approach to estimate $\{\boldsymbol{A}_p\}_{p=1}^P$ is to minimize the following objective:

$$\underset{\{\boldsymbol{A}_p\}_{p=1}^P}{\arg\min}\ \mathcal{L}\left(\{\boldsymbol{A}_p\}_{p=1}^P\right) + \lambda \sum_{n=1}^{N}\sum_{\substack{n'=1\\ n'\neq n}}^{N} \mathbb{1}\Big\{\sum_{p=1}^{P}|a_{n,n'}^{(p)}|\Big\}, \tag{A.4}$$

where $\mathcal{L}(\cdot)$ is given by

$$\begin{aligned}\mathcal{L}\left(\{\boldsymbol{A}_p\}_{p=1}^P\right) &\triangleq \sum_{\tau=P}^{T-1}\left\|\boldsymbol{f}[\tau] - \sum_{p=1}^{P}\boldsymbol{A}_p\boldsymbol{f}[\tau-p]\right\|_2^2 \\ &= \sum_{n=1}^{N}\sum_{\tau=P}^{T-1}\left(f_n[\tau] - \sum_{n'=1}^{N}\sum_{p=1}^{P} a_{n,n'}^{(p)} f_{n'}[\tau-p]\right)^2\end{aligned}$$

and it is a quadratic empirical loss function promoting data fit; and $\mathbb{1}$ is an indicator function satisfying $\mathbb{1}\{x\} = 0$ if $x = 0$ and $\mathbb{1}\{x\} = 1$ if $x \neq 0$. The second term in (A.4) equals the cardinality of $\mathscr{E}$, i.e., the number of edges, times the regularization parameter $\lambda > 0$; and therefore promotes a group-sparse structure in $\{\boldsymbol{A}_p\}_{p=1}^P$ to exploit thus the prior information that the number of edges in $\mathscr{E}$ is small. Self-connections are

not regularized. The parameter λ controls the tradeoff between the data fit and sparsity, and can be adjusted e.g. via cross-validation [11].

For notational convenience, let us introduce the variables $\boldsymbol{a}_{n,n'} \triangleq [a_{n,n'}^{(1)}, a_{n,n'}^{(2)}, ..., a_{n,n'}^{(P)}]^\top \in \mathbb{R}^P$, $\boldsymbol{a}_n \triangleq [\boldsymbol{a}_{n,1}^\top, \boldsymbol{a}_{n,2}^\top, ..., \boldsymbol{a}_{n,N}^\top]^\top \in \mathbb{R}^{NP}$, as well as

$$\boldsymbol{g}[\tau] \triangleq \mathrm{vec}\big(\big[\boldsymbol{f}[\tau-1], \ldots, \boldsymbol{f}[\tau-P]\big]^\top\big) \in \mathbb{R}^{NP}. \tag{A.5}$$

Then, $\mathcal{L}\left(\{\boldsymbol{A}_p\}_{p=1}^P\right)$ can be expressed as $\sum_{n=1}^N \mathcal{L}^{(n)}(\boldsymbol{a}_n)$, where

$$\mathcal{L}^{(n)}(\boldsymbol{a}_n) \triangleq \sum_{\tau=P}^{T} \left(f_n[\tau] - \boldsymbol{g}^\top[\tau]\boldsymbol{a}_n\right)^2. \tag{A.6}$$

With this notation, (A.4) can be expressed as

$$\{\hat{\boldsymbol{a}}_n\}_{n=1}^N = \underset{\{\boldsymbol{a}_n\}_{n=1}^N}{\arg\min} \sum_{n=1}^N \left[\mathcal{L}^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^N \mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2\}\right].$$

Since the above problem is non-convex, [66] proposed recovering sparse coefficients by minimizing the following group-Lasso regularized functional

$$\{\hat{\boldsymbol{a}}_n\}_{n=1}^N = \underset{\{\boldsymbol{a}_n\}_{n=1}^N}{\arg\min} \sum_{n=1}^N \left[\mathcal{L}^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^N \|\boldsymbol{a}_{n,n'}\|_2\right] \tag{A.7}$$

which clearly separates across $\boldsymbol{a}_n$ as

$$\hat{\boldsymbol{a}}_n = \underset{\boldsymbol{a}_n}{\arg\min}\ \mathcal{L}^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^N \|\boldsymbol{a}_{n,n'}\|_2. \tag{A.8}$$

The batch estimation criterion in (A.8) requires all data $\{\boldsymbol{f}[t]\}_{t=0}^{T-1}$ before an estimate can be obtained. The rest of this section proposes an online criterion that provides an estimate per each time-slot when new data is received, and furthermore enables tracking topology changes. To this end, the objective in (A.8) is replaced with a time-dependent objective as follows:

$$\hat{\boldsymbol{a}}_n[t] = \underset{\boldsymbol{a}_n[t]}{\arg\min}\ \mathcal{L}^{(n)}(\boldsymbol{a}_n[t], t) + \lambda \sum_{\substack{n'=1 \\ n'\neq n}}^N \|\boldsymbol{a}_{n,n'}[t]\|_2, \tag{A.9}$$

where $\hat{\boldsymbol{a}}_n[t]$ is the estimate of $\boldsymbol{a}_n$ at time t,

$$\boldsymbol{a}_n[t] \triangleq [\boldsymbol{a}_{n,1}^\top[t], \boldsymbol{a}_{n,2}^\top[t], ..., \boldsymbol{a}_{n,N}^\top[t]]^\top \in \mathbb{R}^{NP} \tag{A.10}$$

contains the optimization variables at time t, and

$$\mathcal{L}^{(n)}(\boldsymbol{a}_n[t], t) \triangleq \sum_{\tau=P}^{t} \gamma^{t-\tau} \left(f_n[\tau] - \boldsymbol{g}^\top[\tau]\boldsymbol{a}_n[t]\right)^2 \tag{A.11}$$

is a time-dependent version of the empirical loss function in (A.6), where $0 < \gamma \leq 1$ is a user-selected forgetting factor. The latter weights recent samples more heavily than the older ones and is introduced to facilitate tracking topology changes. Observe that (A.11), and therefore (A.9), only depend on data up to time t, and therefore $\hat{\boldsymbol{a}}_n[t]$ can be obtained right after $\{\boldsymbol{f}[\tau]\}_{\tau=0}^{t}$ have been received. The rest of this section proposes two algorithms to solve (A.9) in an online fashion.

A.3.1 Regularized RLS (R-RLS)

A solver based on RLS is proposed in this section. To this end, consider the following valid subgradient of the (non differentiable) regularization term in (A.9)

$$\boldsymbol{h}(\boldsymbol{a}_n[t]) \triangleq [\nabla^{s\top}_{\boldsymbol{a}_{n,1}[t]} \|\boldsymbol{a}_{n,1}[t]\|_2, \ldots, \nabla^{s\top}_{\boldsymbol{a}_{n,n-1}[t]} \|\boldsymbol{a}_{n,n-1}[t]\|_2, \boldsymbol{0}, \nabla^{s\top}_{\boldsymbol{a}_{n,n+1}[t]} \|\boldsymbol{a}_{n,n+1}[t]\|_2, \ldots, \nabla^{s\top}_{\boldsymbol{a}_{n,N}[t]} \|\boldsymbol{a}_{n,N}[t]\|_2]^\top, \quad \text{(A.12)}$$

where

$$\nabla^{s}_{\boldsymbol{x}} \|\boldsymbol{x}\|_2 = \begin{cases} \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}, & \boldsymbol{x} \neq \boldsymbol{0} \\ \boldsymbol{0}, & \boldsymbol{x} = \boldsymbol{0}. \end{cases} \quad \text{(A.13)}$$

On the other hand, let

$$\boldsymbol{\Phi}[t] \triangleq \sum_{\tau=P}^{t} \gamma^{t-\tau} \boldsymbol{g}[\tau]\boldsymbol{g}^\top[\tau], \quad \text{(A.14)}$$

$$\boldsymbol{r}_n[t] \triangleq \sum_{\tau=P}^{t} \gamma^{t-\tau} f_n[\tau]\boldsymbol{g}[\tau], \quad \text{(A.15)}$$

respectively denote a weighted sample auto-correlation matrix of $\boldsymbol{g}[\tau]$ and a weighted sample cross-correlation of $f_n[\tau]$ and $\boldsymbol{g}[\tau]$. Note that $\boldsymbol{\Phi}[t]$ and $\boldsymbol{r}_n[t]$ can be updated recursively as

$$\boldsymbol{\Phi}[t] = \gamma\boldsymbol{\Phi}[t-1] + \boldsymbol{g}[t]\boldsymbol{g}^\top[t], \quad \text{(A.16)}$$

$$\boldsymbol{r}_n[t] = \gamma\boldsymbol{r}_n[t-1] + f_n[t]\boldsymbol{g}[t]. \quad \text{(A.17)}$$

In view of these equations, it can be shown that the algorithm in [99] reduces to **Algorithm 6** when solving (A.9). This algorithm offers an approximate solution since it relies on the assumption that the estimated coefficients do not change abruptly between consecutive time steps.

The complexity of **Algorithm 6** is dominated by the N computations of $\boldsymbol{Q}[t]\boldsymbol{h}(\hat{\boldsymbol{a}}_n[t-1])$ (which are $\mathcal{O}(N^2P^2)$), and therefore the overall complexity is $\mathcal{O}(N^3P^2)$.

A.3.2 Online Block Coordinate Descent (OBCD)

When N is very large, the computational burden of **Algorithm 6** can become prohibitive given its cubic-order complexity with respect to N. To alleviate this limitation, this section proposes an online method with quadratic complexity in N. The proposed method is based on performing a single iteration of BCD to minimize (A.9). A related approach

Algorithm 6 Group-sparse R-RLS algorithm

Input: $\sigma, P, \lambda, \gamma, \{\boldsymbol{f}[\tau]\}_{\tau=0}^{t}$
Output: $\{\hat{\boldsymbol{a}}_n[t]\}_{n=1}^{N}$
Initialization: $\hat{\boldsymbol{a}}_n[P-1] = \boldsymbol{0}, \boldsymbol{Q}[P-1] = \sigma^{-1}\boldsymbol{I}$

1: **for** $t = P, P+1, \ldots,$ **do**
2: $\quad \boldsymbol{k}[t] = \dfrac{\boldsymbol{Q}[t-1]\boldsymbol{g}[t]}{\gamma + \boldsymbol{g}^{\top}[t]\boldsymbol{Q}[t-1]\boldsymbol{g}[t]}$
3: $\quad \boldsymbol{Q}[t] = \gamma^{-1}\boldsymbol{Q}[t-1] - \gamma^{-1}\boldsymbol{k}[t]\boldsymbol{g}^{\top}[t]\boldsymbol{Q}[t-1]$
4: $\quad$ **for** $n = 1, 2, \ldots, N$ **do**
5: $\quad\quad e_n[t] = f_n[t] - \boldsymbol{g}^{\top}[t]\hat{\boldsymbol{a}}_n[t-1]$
6: $\quad\quad \hat{\boldsymbol{a}}_n[t] = \hat{\boldsymbol{a}}_n[t-1] + e_n[t]\boldsymbol{k}[t] + \lambda(\gamma-1)\boldsymbol{Q}[t]\boldsymbol{h}(\hat{\boldsymbol{a}}_n[t-1])$
7: $\quad$ **end for**
8: **end for**

for solving batch group-Lasso problems was proposed in [100]. Note that although (A.9) can be solved directly by off-the-shelf convex optimization solvers, their complexity is high and therefore an algorithm tailored to (A.9) is preferable.

Block coordinate descent is based on iteratively minimizing a given objective with respect to a group of variables while keeping the rest of groups fixed to their values in previous iterations. Fortunately, at each minimization step the function is differentiable in all points except the zero, while the minimization step at the zero vector is simple to be performed.

The right-hand side of (A.9) can be rewritten in terms of the recursively computed $\boldsymbol{\Phi}[t]$ and $\boldsymbol{r}_n[t]$ as

$$\arg\min_{\boldsymbol{a}_n[t]} \boldsymbol{a}_n^{\top}[t]\boldsymbol{\Phi}[t]\boldsymbol{a}_n[t] - 2\mathbf{r}_n^{\top}[t]\boldsymbol{a}_n[t] + \lambda\sum_{\substack{n'=1\\ n'\neq n}}^{N} \|\boldsymbol{a}_{n,n'}[t]\|_2$$

For each t and n, the proposed algorithm performs N block updates: the i-th update modifies the i-th group $\boldsymbol{a}_{n,i}[t]$ whereas all other entries in $\boldsymbol{a}_n[t]$ are kept fixed.

Upon appropriately permuting the entries of $\boldsymbol{a}_n[t], \boldsymbol{\Phi}[t]$, and $\boldsymbol{r}_n[t]$, the minimization of the above objective with respect to the i-th group can be expressed as

$$\hat{\boldsymbol{a}}_{n,i}[t] = \arg\min_{\boldsymbol{a}_{n,i}[t]} \begin{bmatrix}\boldsymbol{a}_{n,\bar{i}}^{\top}[t] & \boldsymbol{a}_{n,i}^{\top}[t]\end{bmatrix} \begin{bmatrix}\boldsymbol{\Phi}_{\bar{i}\bar{i}}[t] & \boldsymbol{\Phi}_{\bar{i}i}[t]\\ \boldsymbol{\Phi}_{i\bar{i}}[t] & \boldsymbol{\Phi}_{ii}[t]\end{bmatrix} \begin{bmatrix}\boldsymbol{a}_{n,\bar{i}}[t]\\ \boldsymbol{a}_{n,i}[t]\end{bmatrix} - 2\begin{bmatrix}\mathbf{r}_{n,i}^{\top}[t] & \mathbf{r}_{n,\bar{i}}^{\top}[t]\end{bmatrix} \begin{bmatrix}\boldsymbol{a}_{n,\bar{i}}[t]\\ \boldsymbol{a}_{n,i}[t]\end{bmatrix} + \lambda\Big(\sum_{n'=1}^{N} \|\boldsymbol{a}_{n,n'}[t]\|_2\Big)\mathbb{1}\{i-n\} \quad \text{(A.18)}$$

where $\hat{\boldsymbol{a}}_{n,i}[t]$ collects the entries of the i-th group in $\hat{\boldsymbol{a}}_n[t]$; $\boldsymbol{a}_{n,\bar{i}}[t]$ collects the entries in the complementary set of i-th group; and similar definitions apply for $\mathbf{r}_{n,\bar{i}}[t]$, $\boldsymbol{\Phi}_{\bar{i}\bar{i}}[t]$, and $\boldsymbol{\Phi}_{i\bar{i}}[t]$ $(= \boldsymbol{\Phi}_{\bar{i}i}^{\top}[t]$ because of symmetry). Ignoring the constant terms, the right-hand side of (A.18) can be rewritten as

$$\hat{\boldsymbol{a}}_{n,i}[t] = \arg\min_{\boldsymbol{a}_{n,i}[t]} \boldsymbol{a}_{n,i}^{\top}[t]\boldsymbol{\Phi}_{ii}[t]\boldsymbol{a}_{n,i}[t] + 2(\boldsymbol{\Phi}_{i\bar{i}}[t]\boldsymbol{a}_{n,\bar{i}}[t] - \mathbf{r}_{n,i}[t])^{\top}\boldsymbol{a}_{n,i}[t] + \lambda\|\boldsymbol{a}_{n,i}[t]\|_2 \quad \text{(A.19)}$$

When $i = n$, the last term is zero and therefore (A.19) constitutes a conventional least-squares equation and its solution is $\hat{\boldsymbol{a}}_{n,n}[t] = \boldsymbol{\Phi}_{nn}^{\dagger}\boldsymbol{p}_n$, where $\boldsymbol{M}_i \triangleq \boldsymbol{\Phi}_{ii}$ and $\boldsymbol{p}_i \triangleq \boldsymbol{\Phi}_{i\bar{i}}\boldsymbol{a}_{n,\bar{i}} - \mathbf{r}_{n,i}^{\top}[t]$.

Conversely, when $i \neq n$, we solve the optimization using Newton's method. This requires the cost function to be twice differentiable at every point. In our case, this holds at every point except the zero vector; fortunately, when solving (A.19) this case can be circumvented. Note first that it can be proven [100] that $\mathbf{0}$ will be an optimizer of (A.19) iff $\|\mathbf{p}\|_2 \leq \lambda$. Second, if Newton's method is initialized at an $\boldsymbol{a}^{(0)}$ that yields a negative objective and every iteration effectively reduces the objective, then the optimization is done over a sub-level set where the gradient and the Hessian are always well defined.

Consequently, the proposed solver first checks if $\mathbf{0}$ is the optimal solution to (A.19), and if it is not, $\boldsymbol{a}^{(0)}$ is initialized as $\boldsymbol{a}^{(0)} = \left((\lambda \|\mathbf{p}\|_2 - \|\mathbf{p}\|_2^2)/(\mathbf{p}^{\top}\mathbf{M}\mathbf{p})\right)\mathbf{p}$ which is the solution to a line search over the half line that starts at $\mathbf{0}$ in the steepest descent direction. Afterwards, standard Newton iterations are performed until convergence as detailed in **Algorithm 8**.

Algorithm 7 further generalizes [101, Algorithm 3] which can only accommodate groups of size 1 (regular Lasso). Regarding complexity, **Algorithm 8** is called $N(N-1)$ times and its complexity is dominated by the inversion of the $P \times P$ Hessian. Consequently, OBCD entails a complexity of $\mathcal{O}(N^2P^3)$ per time instant.

Algorithm 7 Online Block Coordinate Descent

Input: $\lambda, \gamma, \sigma, \{\boldsymbol{f}[\tau]\}_{\tau=0}^{t}$,
Output: $\{\hat{\boldsymbol{a}}_n[t]\}_{n=1}^{N}$
Initialization: $\hat{\boldsymbol{a}}_n[P-1] = \mathbf{0}, \boldsymbol{\Phi}[P-1] = \sigma^2\boldsymbol{I}, \mathbf{r}_n[P-1] = \mathbf{0}$, and $\mathbf{g}[P-1]$ as in (B.4)

for $t = P, P+1, \ldots,$ **do**
  Obtain $\boldsymbol{\Phi}[t]$ as in (A.16)
  for $n = 1, 2, \ldots, N$ **do**
    for $i = 1, 2, \ldots, N$ **do**
      Obtain $\mathbf{r}_n[t]$ as in (A.17)
      Set $\boldsymbol{a}_{n,j}[t] = \boldsymbol{a}_{n,j}[t-1] \forall j \neq i$
      Update $\boldsymbol{a}_{n,i}[t]$ via (A.19)
    end for
  end for
end for
end for

A.4 Numerical Experiments

The performance of the proposed online algorithms is compared with the batch group-Lasso approach by numerical tests in this section. A network is simulated by a random graph with $N = 15$ nodes, and an edge set randomly generated by an Erdos-Renyi model with edge probability p_e constant for every pair of nodes except for self-loops, which have

Algorithm 8 Solve (A.19) via Newton's method

Input: $\boldsymbol{\Phi}_{ii}[t], \boldsymbol{\Phi}_{i\bar{i}}[t], \lambda, \mathbf{r}_{n,i}[t], \boldsymbol{a}_{n,\bar{i}}[t]$ **Output:** $\hat{\boldsymbol{a}}_{n,i}[t]$

$\mathbf{M} = \boldsymbol{\Phi}_{ii}[t]$; $\mathbf{p} = \boldsymbol{\Phi}_{i\bar{i}}[t]\boldsymbol{a}_{n,\bar{i}} - \mathbf{r}_{n,i}[t]$

if $\|\mathbf{p}\|_2 \leq \lambda$ **then return** $\hat{\boldsymbol{a}}_n[t] = \mathbf{0}$

else $\boldsymbol{a}^{(0)} = \left((\lambda \|\mathbf{p}\|_2 - \|\mathbf{p}\|_2^2)/(\mathbf{p}^\top \mathbf{M}\mathbf{p})\right)\mathbf{p}$

for $k = 0, 1, \ldots$ until convergence **do**

$$\mathbf{H} = \mathbf{M} + \lambda \left(\frac{\mathbf{I}}{\|\boldsymbol{a}^{(k)}\|_2} - \frac{\boldsymbol{a}^{(k)}\boldsymbol{a}^{\top(k)}}{\|\boldsymbol{a}^{(k)}\|_2^3} \right)$$

$$\mathbf{g} = \mathbf{M}\boldsymbol{a}^{(k)} + \mathbf{p} + \frac{\lambda \boldsymbol{a}^{(k)}}{\|\boldsymbol{a}^{(k)}\|_2}$$

$\boldsymbol{a}^{(k+1)} = \boldsymbol{a}^{(k)} - \mathbf{H}^\dagger \mathbf{g}$

end forreturn $\hat{\boldsymbol{a}}_{n,i}[t] = \boldsymbol{a}^{(k)}$

edge probability one. A VAR process with order $P = 5$ is generated by drawing the active coefficients of $\boldsymbol{A}_p$ from a Gaussian distribution, setting the rest of the coefficients to zero, and normalizing the result so that the largest-magnitude eigenvalue of $\boldsymbol{A}_p$ is less than $1/P$, thus guaranteeing a stable VAR process. A time series of T time instants is generated according to (E.1) with $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, 0.02\boldsymbol{I})$. The regularization parameter is chosen as $\lambda = 0.02$.

Two error measures are used to compare the performance of the developed methods. In the first case, the estimated VAR coefficients $\{\hat{\boldsymbol{a}}_n[t]\}$ are directly compared to the true coefficients and the evolution of the normalized mean squared deviation (NMSD) defined as $\mathbb{E}[\|\sum_n(\hat{\boldsymbol{a}}_n[t] - \boldsymbol{a}_n)\|_2^2]/\mathbb{E}[\sum_n\|\boldsymbol{a}_n\|_2^2]$ is represented in the top pane of Fig. A.1. In the second case, the coefficients are used to predict the process in the next time instant and the normalized mean square error (NMSE) is depicted in the bottom pane. To reduce computational burden, the two error measures for the batch approach are evaluated for $T = 50, 100, 150, \ldots, 650$, considering all available data up to time T. The dashed line is added to improve visualization. These results suggest that both algorithms have similar convergence rates and their estimate approaches the batch solution after processing a large number of samples. Although OBCD shows a slight advantage over R-RLS, a main factor to choose one approach or the other is the computational efficiency. As a short wrap-up, recall that OBCD has $\mathcal{O}(N^2P^3)$ computation, and R-RLS has $\mathcal{O}(N^3P^2)$. This makes the former more suitable for large networks, whereas the latter enjoys fast performance for large filter order.

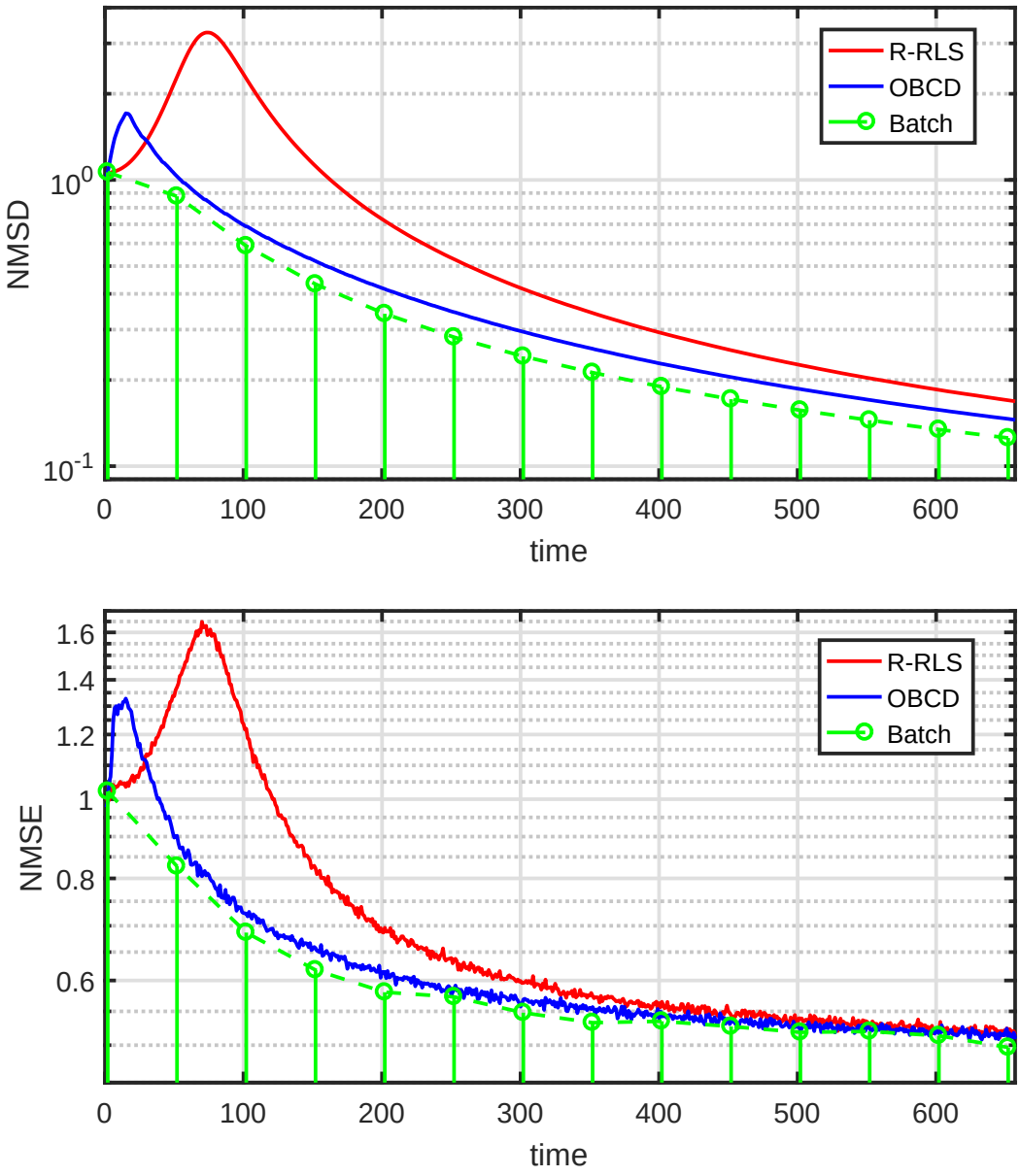


Figure A.1: Normalized Mean Squared Deviation (top) and Normalized Mean Squared Error (bottom).

Appendix B

Paper B

Title:	Online Topology Identification from Vector Autoregressive Time Series
Authors:	**Bakht Zaman**, Luis M. Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano
Affiliation:	Center Intelligent Signal Processing and Wireless Networks (WISENET) Department of ICT, University of Agder, Grimstad, Norway
Journal:	Submitted to IEEE Trans. Signal Process.

Online Topology Identification from Vector Autoregressive Time Series

Bakht Zaman, Luis M. Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano

***Abstract*— Causality graphs are routinely estimated in social sciences, natural sciences, and engineering due to their capacity to efficiently represent the spatiotemporal structure of multi-variate data sets in a format amenable for human interpretation, forecasting, and anomaly detection. A popular approach to mathematically formalize causality is based on vector autoregressive (VAR) models and constitutes an alternative to the well-known, yet usually intractable, Granger causality. Relying on such a VAR causality notion, this paper develops two algorithms with complementary benefits to track time-varying causality graphs in an online fashion. Their constant complexity per update also renders these algorithms appealing for big-data scenarios. Despite using data sequentially, both algorithms are shown to asymptotically attain the same average performance as a batch estimator which uses the entire data set at once. To this end, sublinear (static) regret bounds are established. Performance is also characterized in time-varying setups by means of dynamic regret analysis. Numerical results with real and synthetic data further support the merits of the proposed algorithms in static and dynamic scenarios.**

B.1 Introduction

Inferring causal relations among time series finds countless applications in social sciences, natural sciences, and engineering. These relations are typically encoded as the edges of a causality graph, where each node corresponds to a time series, and oftentimes reveal the topology of e.g. an underlying social, biological, or brain network [13]. Causality graphs may also offer valuable insights into the spatio-temporal structure of time series and assist data processing tasks such as forecasting [102], signal reconstruction [3], anomaly detection [2], and dimensionality reduction [9]. In some applications, graphs capturing different forms of causality can be constructed based on domain knowledge; see e.g. [11, Ch. 8]. However, this approach is often impractical in the aforementioned applications due to the large dimension of the data or because such prior knowledge is unavailable. Instead, causality graphs need to be inferred from data in these situations. This paper accomplishes this task in an online fashion.

Identifying graphs capturing the spatiotemporal "interactions" among time series has attracted great attention [13, 22]. Some approaches focus on instantaneous interactions, i.e., they disregard the temporal structure. The simplest one is to connect two nodes if the sample correlation between the associated time series exceeds a certain threshold [13]. To distinguish mediated from unmediated interactions [13, Sec. 7.3.2], one may resort to conditional independence, partial correlations, Markov random fields, or other approaches in graph signal processing; see e.g. [14, 16, 15, 11, 19, 21]. For directed interactions, one

may employ structural equation models (SEM) [24] (see also [26] and references therein) or Bayesian networks [11, Sec. 8.1]. However, these methods account only for *memoryless* interactions, i.e., they cannot accommodate delayed interactions where the value of a time series at a given time instant is related to the past values of other time series.

The earliest effort to formalize the notion of causality among time series is due to Granger [28] and relies on the rationale that *the cause precedes the effect*. A time series is said to be Granger-caused by another if the optimal prediction error of the former is decreased when the past of the latter is taken into account. Albeit elegant, this definition is generally impractical since the *optimal* prediction error is difficult to determine [60, p. 33], [61]. Thus, alternative causality definitions based on vector autoregressive (VAR) models are typically preferred [103, 31, 104]. VAR causality is determined from the support of VAR matrix parameters and is equivalent to Granger causality [43, Chap. 2] in certain cases (yet sometimes treated as equivalent [31, 104]). VAR causality is further motivated by the widespread usage of VAR models to approximate the response of systems of linear partial differential equations [65] and, more generally, in disciplines such as econometrics, bio-informatics, neuroscience, and engineering [62, 63, 64]. VAR topologies are estimated assuming Gaussianity and stationarity in [30, 29] and assuming sparsity in [66, 105, 20, 106]. All these approaches assume that the graph does not change over time. Since this is not the case in many applications, approaches have been devised to identify undirected time-varying topologies [38, 107] and directed piecewise-constant time-varying topologies [47].

The complexity of all previously discussed approaches becomes prohibitive for long observation windows since they process the entire data set at once and cannot accommodate data arriving sequentially. The modern approach to tackle these issues is *online* optimization, where an estimate is refined with every new data instance. Existing online topology identification algorithms include [34, 26],[35, 36, 37], and [32], but they only account for memoryless interactions.

The present work is the first to propose online algorithms to estimate the *memory-aware* causality graphs associated with a collection of time series.[1] The specific contributions include: (C1) An online algorithm, termed *Topology Identification via Sparse Online learning* (TISO), which estimates directed VAR causality graphs and therefore captures memory-based interactions. Sparse and (possibly) time-varying topologies are tracked by a composite-objective iteration [67] that minimizes a sequential version of the criterion in [66] while promoting *sparse updates*. In addition, computational complexity and memory requirements per iteration of the algorithm remain constant, which renders it suitable for sequential and big-data scenarios. (C2) A second algorithm, named *Topology Identification via Recursive Sparse Online learning* (TIRSO), which improves the tracking performance of TISO and robustness to input variability by minimizing a novel estimation criterion inspired by *recursive least squares* (RLS) where the instantaneous loss function accounts for past samples. TIRSO inherits certain benefits of TISO but incurs a moder-

[1]The conference version [44] of this work presents two online algorithms different from the algorithms presented here. One is based on a subgradient approximation and the other one is based on block coordinate minimization via Newton's method. In addition, no convergence guarantees were provided. The related work in [108] was run in parallel and published subsequently.

ate increase in computational complexity, which is still constant per iteration. (C3) In terms of performance analysis: (i) it is established that the hindsight solution of TISO and TIRSO are asymptotically the same. (ii) The convergence of TISO and TIRSO is established by deriving sublinear static regret bounds. Hence, in the long run, these algorithms perform as well as the best (batch) predictor in hindsight, which supports their adoption for online topology identification. Remarkably, the performance (regret) analysis does not require probabilistic assumptions, which endows the developed approaches with high generality. (iii) A logarithmic regret bound is proved for TIRSO. (iv) To analyze the performance of TIRSO when the topology is time-varying, a dynamic regret bound is derived. Moreover, the steady-state error of TIRSO in time-varying scenarios is quantified in terms of the data properties. (C4) Finally, performance is empirically validated through extensive experiments with synthetic and real data sets.

The rest of the paper is organized as follows: Sec. B.2 presents the model, a batch estimation criterion, and background on online optimization. Sec. B.3 develops TISO and TIRSO. Sec. B.4 Sec. B.5 respectively assess performance analytically and via simulations, whereas Sec. B.6 concludes the paper. All code will be made public at the authors' websites.

Notation. Bold lowercase (uppercase) letters denote column vectors (matrices). Operators $\mathbb{E}[\cdot]$, ∇, $\tilde{\nabla}$, ∂, $(\cdot)^\top$, $\text{vec}(\cdot)$, $\lambda_{\max}(\cdot)$, $\mathcal{R}(\cdot)$, $(\cdot)^\dagger$, and $\text{diag}(\cdot)$ respectively denote expectation, gradient, subgradient, sub-differential, matrix transpose, vectorization, maximum eigenvalue, range or column space, pseudo-inverse, and diagonal of a matrix. Symbols $\mathbf{0}_N$, $\mathbf{1}_N$, $\mathbf{0}_{N\times N}$, and $\boldsymbol{I}_N$ respectively represent the all-zero vector of size N, the all-ones vector of size N, the all-zero matrix of size $N \times N$, and the size-N identity matrix. Also, $[\cdot]_+ = \max(\cdot, 0)$. For functions $f(x)$ and $g(x)$, the notation $f(x) \propto g(x)$ means $\exists a > 0, b : f(x) = ag(x) + b$. Finally, $\mathbb{1}$ is the indicator satisfying $\mathbb{1}\{x\} = 1$ if x is true and $\mathbb{1}\{x\} = 0$ otherwise.

B.2 Preliminaries

After outlining the notion of directed causality graphs, this section reviews how these graphs can be identified in a batch fashion. Later, the basics of online optimization are described.

B.2.1 Directed Causality Graphs

Consider a collection of N time series $\{y_n[t]\}_t$, $n = 1, ..., N$, where $y_n[t]$ denotes the value of the n-th time series at time t. A causality graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ is a graph where the n-th vertex in $\mathcal{V} = \{1, \ldots, N\}$ is identified with the n-th time series $\{y_n[t]\}_t$ and there is an edge (or arc) from n' to n (i.e. $(n, n') \in \mathcal{E}$) if and only if (iff) $\{y_{n'}[t]\}_t$ *causes* $\{y_n[t]\}_t$ according to a certain causality notion. For the reasons outlined in Sec. E.1, a prominent notion of causality described later in this section can be defined using VAR models. To this end, let $\boldsymbol{y}[t] \triangleq [y_1[t], \ldots, y_N[t]]^\top$ and define a VAR time series $\{\boldsymbol{y}[t]\}_t$ as a sequence

generated by the order-P VAR model[43]

$$\boldsymbol{y}[t] = \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{y}[t-p] + \boldsymbol{u}[t], \tag{B.1}$$

where $\boldsymbol{A}_p \in \mathbb{R}^{N\times N}, p = 1, \ldots, P$, are the VAR parameters[2] and $\boldsymbol{u}[t] \triangleq [u_1[t], \ldots, u_N[t]]^\top$ is the *innovation process*. This process is generally assumed to be a white zero-mean stochastic process, i.e., $\mathbb{E}[\boldsymbol{u}[t]] = \boldsymbol{0}_N$ and $\mathbb{E}[\boldsymbol{u}[t]\boldsymbol{u}^\top[\tau]] = \boldsymbol{0}_{N\times N}$ for $t \neq \tau$. Yet, the present work does not even need to assume that $\boldsymbol{u}[t]$ is random; see the remark at the end of Sec. B.4. With $a_{n,n'}^{(p)}$ the n, n'-th entry of $\boldsymbol{A}_p$, expression (E.1) becomes

$$\begin{aligned} y_n[t] &= \sum_{n'=1}^{N} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p] + u_n[t] \\ &= \sum_{n'\in\mathcal{N}(n)} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p] + u_n[t] \end{aligned} \tag{B.2}$$

for $n = 1, \ldots, N$, where $\mathcal{N}(n) \triangleq \{n' : \boldsymbol{a}_{n,n'} \neq \boldsymbol{0}_P\}$ and $\boldsymbol{a}_{n,n'} \triangleq [a_{n,n'}^{(1)}, \ldots, a_{n,n'}^{(P)}]^T$. Recognizing the convolution operation in the right-hand side enables one to express (E.2) as $y_n[t] = \sum_{n'\in\mathcal{N}(n)} a_{n,n'}^{(t)} * y_{n'}[t] + u_n[t]$ in signal processing notation. Thus, in a VAR model, $y_n[t]$ equals the sum of noise and the output of $|\mathcal{N}(n)|$ linear time-invariant filters where the n, n'-th filter has input $\{y_{n'}[t]\}_t$ and coefficients $\{a_{n,n'}^{(p)}\}_{p=1}^P$.

When $\boldsymbol{u}[t]$ is a zero-mean and temporally white stochastic process, the term $\hat{y}_n[t] \triangleq \sum_{n'\in\mathcal{N}(n)} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p]$ in (E.2) is the *minimum mean square error estimator* of $y_n[t]$ given the previous values of all time series $\{y_{n'}[\tau], n' = 1, ..., N, \tau < t\}$; see e.g. [61, Sec. 12.7]. The set $\mathcal{N}(n)$ therefore collects the indices of those time series that participate in this optimal predictor of $y_n[t]$ or, alternatively, the information provided by time series $\{y_{n'}[\tau]\}_{\tau<t}$ with $n' \notin \mathcal{N}(n)$ is not informative to predict $y_n[t]$. This motivates the following definition of causality: $\{y_{n'}[t]\}_t$ *VAR-causes* $\{y_n[t]\}_t$ whenever $n' \in \mathcal{N}(n)$. Equivalently, $\{y_{n'}[t]\}_t$ *VAR-causes* $\{y_n[t]\}_t$ if $\boldsymbol{a}_{n,n'} \neq \boldsymbol{0}_P$. VAR causality[3] relations among the N time series can be represented using a causality graph where $\mathcal{E} \triangleq \{(n, n') : \boldsymbol{a}_{n,n'} \neq \boldsymbol{0}_P\}$. Clearly, in such a graph, $\mathcal{N}(n)$ is the in-neighborhood of node n. To quantify the strength of these causality relations, a weighted graph can be constructed by assigning e.g. the weight $\|\boldsymbol{a}_{n,n'}\|_2$ to the edge (n, n').

With these definitions, the *batch* problem of identifying a VAR causality graph reduces to estimating the VAR coefficient matrices $\{\boldsymbol{A}_p\}_{p=1}^P$ given P and the observations $\{\boldsymbol{y}[t]\}_{t=0}^{T-1}$. To simplify notation, form the tensor $\mathcal{A}$ by stacking the matrices $\{\boldsymbol{A}_p\}_{p=1}^P$ along the third dimension as shown in Fig. B.1.

[2]For the sake of clarity, matrices $\{\boldsymbol{A}_p\}_{p=1}^P$ are deemed constant throughout this section. However, all the notions explained here can be easy generalized to time-varying scenarios, as detailed in subsequent sections.

[3]A detailed comparison with Granger causality lies out of scope, yet it is worth mentioning that the main distinction lies in the prediction horizon: whereas VAR causality just pertains to prediction 1 time instant ahead, Granger causality involves prediction of all future samples $y_n[t']$, $t' \geq t$, given the ones up to a certain time instant $\{y_{n'}[\tau],\ n' = 1, \ldots, N,\ \tau < t\}$. Therefore VAR causality implies Granger causality, but the converse is false. See [43, Sec. 2.3.1] for a more detailed comparison.

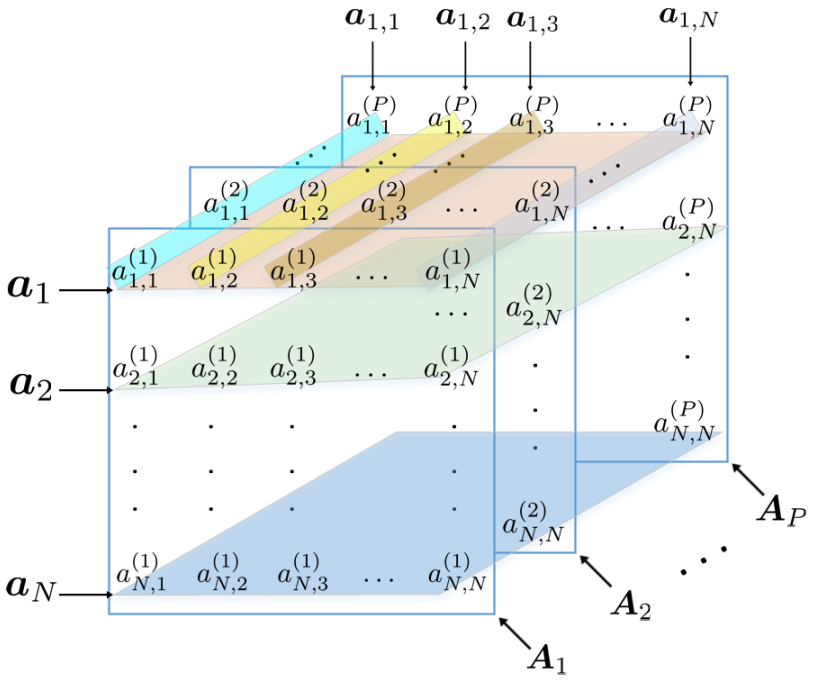


Figure B.1: Tensor $\mathcal{A}$ collecting the VAR parameter matrices.

B.2.2 Batch Estimation Criterion for Topology Identification

This section presents an estimation criterion to address the batch problem formulated in Sec. B.2.1. A natural estimate could be pursued through *least-squares* by minimizing [43]

$$\begin{aligned}\mathcal{L}(\mathcal{A}) &\triangleq \frac{1}{2(T-P)}\sum_{\tau=P}^{T-1}\Big\|\boldsymbol{y}[\tau]-\sum_{p=1}^{P}\boldsymbol{A}_p\,\boldsymbol{y}[\tau-p]\Big\|_2^2 \\ &= \frac{1}{2\,(T-P)}\sum_{n=1}^{N}\sum_{\tau=P}^{T-1}\Big[y_n[\tau]-\sum_{n'=1}^{N}\sum_{p=1}^{P}a_{n,n'}^{(p)}\,y_{n'}[\tau-p]\Big]^2.\end{aligned}$$

This estimation task becomes underdetermined unless the number NT of available data samples meaningfully exceeds the number of unknowns PN^2. To circumvent this limitation, the following criterion has been proposed in [66]:

$$\arg\min_{\mathcal{A}}\ \mathcal{L}(\mathcal{A})+\lambda\sum_{n=1}^{N}\sum_{\substack{n'=1\\ n'\neq n}}^{N}\|\boldsymbol{a}_{n,n'}\|_2\,, \tag{B.3}$$

where $\lambda>0$ is a regularization parameter[4] that can be adjusted e.g. via cross-validation [11, Ch. 1]. The second term in (B.3) is conventionally referred to as a *group-lasso* regularizer and the solution to (B.3) as a *group-lasso* estimate [109]. This promotes a *group-sparse structure* in $\{\boldsymbol{A}_p\}_{p=1}^P$ to exploit the information that the number of edges in

[4]As seen in (B.3), λ is the same for all candidate edges (n,n'). This can be readily replaced with an edge-dependent regularization parameter $\lambda_{n,n'}$ without any complexity increase to exploit possibly available prior-information about edges.

$\mathcal{E}$ is typically small. Self-connections ($\boldsymbol{a}_{n,n}$, $n = 1, ..., N$) are excluded from the regularization term so that the inferred causal relations pertain to the component of each time series that cannot be predicted using its own past [66].

Remarkably, (B.3) separates along n. To see this, let $\boldsymbol{a}_n \triangleq [\boldsymbol{a}_{n,1}^\top, \boldsymbol{a}_{n,2}^\top, ..., \boldsymbol{a}_{n,N}^\top]^\top \in \mathbb{R}^{NP}$ and

$$\boldsymbol{g}[t] \triangleq \mathrm{vec}\big([\boldsymbol{y}[t-1], \ldots, \boldsymbol{y}[t-P]]^\top \big) \in \mathbb{R}^{NP}, \tag{B.4}$$

and express $\mathcal{L}(\mathcal{A})$ as $\mathcal{L}(\mathcal{A}) = \sum_{n=1}^N \ell^{(n)}(\boldsymbol{a}_n)$, where $\ell^{(n)}(\boldsymbol{a}_n) \triangleq 1/(T-P) \sum_{t=P}^{T-1} \ell_t^{(n)}(\boldsymbol{a}_n)$ and $\ell_t^{(n)}(\boldsymbol{a}_n) \triangleq 1/2(y_n[t] - \boldsymbol{g}^\top[t]\boldsymbol{a}_n)^2$. Then, (B.3) becomes

$$\{\boldsymbol{a}_n^*\}_{n=1}^N = \underset{\{\boldsymbol{a}_n\}_{n=1}^N}{\arg\min} \sum_{n=1}^N [\ell^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^N \|\boldsymbol{a}_{n,n'}\|_2], \tag{B.5}$$

with

$$\boldsymbol{a}_n^* = \underset{\boldsymbol{a}_n}{\arg\min}\ \ell^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^N \|\boldsymbol{a}_{n,n'}\|_2 \tag{B.6}$$

for $n = 1, \ldots, N$. Thus, the VAR causality graph can be identified by separately estimating the VAR coefficients, and hence incoming edge weights, for each node.

The batch estimation criterion in (B.6) requires all data $\{\boldsymbol{y}[t]\}_{t=0}^{T-1}$ before an estimate can be obtained and cannot track changes. Furthermore, solving (B.6) eventually becomes prohibitively complex for sufficiently large T. To address these challenges, this paper adopts the framework of online optimization, which is reviewed in the following subsection.

B.2.3 Background on Online Optimization

This section reviews the fundamental notions of online optimization from a general perspective, not necessarily applied to the problem of topology identification. To this end, consider the generic unconstrained optimization problem

$$\underset{\boldsymbol{a}}{\text{minimize}}\ \frac{1}{T_0} \sum_{t=0}^{T_0-1}\ h_t(\boldsymbol{a}), \tag{B.7}$$

where $h_t(\boldsymbol{a})$ is a convex function, which in many applications depends on the data received at time t. For example, in least squares $h_t(\boldsymbol{a}) = \|\boldsymbol{X}[t]\boldsymbol{a} - \boldsymbol{y}[t]\|_2^2$, where $\boldsymbol{y}[t]$ and $\boldsymbol{X}[t]$ are the data vector and matrix made available at time t. To solve (B.7), it is necessary that all $\{h_t(\boldsymbol{a})\}_{t=0}^{T_0-1}$ be available. Approaches that process all data at once are termed *batch* and, hence, suffer from potentially long waiting times, which generally render them inappropriate for real-time operation. Besides, computational complexity and memory generally grow super-linearly with T_0, which eventually becomes prohibitive.

Online algorithms alleviate these limitations. Let $\boldsymbol{a}[t+1]$ denote an estimate of the solution to (B.7) at time t produced by an online algorithm. Online algorithms compute a new $\boldsymbol{a}[t+1]$ every time a new $(\boldsymbol{X}[t], \boldsymbol{y}[t])$ data element (or, more generally, a new $h_t(\boldsymbol{a})$) is processed. At every iteration, also known as *update*, $\boldsymbol{a}[t+1]$ is obtained from $\boldsymbol{a}[t]$, $\boldsymbol{y}[t]$,

$\boldsymbol{X}[t]$, and possibly some additional information carried from each update to the next. The memory requirements and number of arithmetic operations per iteration must not grow unbounded for increasing t. This requirement rules out approaches involving solving (B.7) as a batch problem per update or carrying all the past data $\{(\boldsymbol{X}[\tau], \boldsymbol{y}[\tau])\}_{\tau=0}^{t-1}$ from the $(t-1)$-th update to the t-th update. Thus, online algorithms are especially appealing when data vectors are received sequentially or T_0 is so large that batch solvers are not computationally affordable. Additionally, online algorithms can track changes in the underlying model.

The most common performance metric to evaluate online algorithms is the *regret*, which quantifies the cumulative loss incurred by an online algorithm relative to the loss corresponding to the optimal constant solution in hindsight. Formally, the (static) regret[5] at iteration $T_0 - 1$ is given by [46]:

$$R_s[T_0] \triangleq \sum_{t=0}^{T_0-1} [h_t(\boldsymbol{a}[t]) - h_t(\boldsymbol{a}^*[T_0])], \tag{B.8}$$

where $\boldsymbol{a}^*[T_0] \triangleq \arg\min_{\boldsymbol{a}} (1/T_0) \sum_{t=0}^{T_0-1} h_t(\boldsymbol{a})$ is the optimal *constant* hindsight solution, i.e., the batch solution after T_0 data vectors have been processed. To be deemed admissible, online algorithms must yield a *sublinear regret*, i.e., $R_s[T_0]/T_0 \to 0$ as $T_0 \to \infty$. Thus, online algorithm with sublinear regret perform asymptotically as well as the batch solution *on average*. It is worth noting that the online learning framework does not involve statistical assumptions on the data, which can even be generated by an "adversary" [54].

In dynamic settings where the parameters of the data generating process vary over time, $\boldsymbol{a}^*[T_0]$ may not be a suitable reference since its computation involves potentially very old data, namely $\{h_t\}_{t \ll T_0}$, which is informative about old values of the true parameters but not about the new values. In those cases, it is customary to compare against the instantaneous minimizer $\boldsymbol{a}^\circ[t] \triangleq \arg\min_{\boldsymbol{a}} h_t(\boldsymbol{a})$ by means of the so-called dynamic regret [56], [57]:

$$R_d[T_0] \triangleq \sum_{t=0}^{T_0-1} \left[h_t(\boldsymbol{a}[t]) - h_t(\boldsymbol{a}^\circ[t])\right].$$

More details about the dynamic regret are given in Sec. C.3.

B.3 Online Topology Identification

This section develops online algorithms for the considered problem of topology identification from time series. To this end, cast (B.6) for the n-th node in the form (B.7) by setting

$$h_t(\boldsymbol{a}_n) = \ell_{t+P}^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 , \tag{B.9}$$

[5]The static regret is known simply as regret in earlier works, e.g. [46], and different types of regret were formalized later, see e.g. [56].

for $t = 0, ..., T - P - 1$. The most immediate approach to solve (B.7) would be applying *online subgradient descent* (OSGD), whose updates are given by $\boldsymbol{a}_n[t+1] = \boldsymbol{a}_n[t] - \alpha_t \tilde{\mathbf{w}}_n[t]$ with $\tilde{\mathbf{w}}_n[t]$ a subgradient of h_t at $\boldsymbol{a}_n[t]$ and α_t the step size at time t. From (B.9), $\tilde{\mathbf{w}}_n[t]$ equals $\nabla \ell_{t+P}^{(n)}(\boldsymbol{a}_n[t])$ plus λ times a valid subgradient of the form

$$\tilde{\nabla}_{\boldsymbol{a}_n} \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \triangleq [\tilde{\nabla}_{\boldsymbol{a}_{n,1}}^{\top} \|\boldsymbol{a}_{n,1}\|_2, \ldots, \tilde{\nabla}_{\boldsymbol{a}_{n,n-1}}^{\top} \|\boldsymbol{a}_{n,n-1}\|_2, \mathbf{0}_P, \tilde{\nabla}_{\boldsymbol{a}_{n,n+1}}^{\top} \|\boldsymbol{a}_{n,n+1}\|_2, \ldots, \tilde{\nabla}_{\boldsymbol{a}_{n,N}}^{\top} \|\boldsymbol{a}_{n,N}\|_2]^{\top},$$

evaluated at $\boldsymbol{a}_n[t]$. For example, for $\boldsymbol{x} \in \mathbb{R}^P$, set $\tilde{\nabla}_{\boldsymbol{x}} \|\boldsymbol{x}\|_2 = \boldsymbol{x}/\|\boldsymbol{x}\|_2$ for $\boldsymbol{x} \neq \mathbf{0}_P$ and $\tilde{\nabla}_{\boldsymbol{x}} \|\boldsymbol{x}\|_2 = \mathbf{0}_P$ for $\boldsymbol{x} = \mathbf{0}_P$. It is easy to see that the resulting iterates $\boldsymbol{a}_n[t]$ are not necessarily sparse; see also [67]. Since the solution to the batch problem is indeed sparse, alternative approaches are required.

To this end, note that OSGD fails to provide sparse iterates because it implicitly linearizes the instantaneous objective $h_t(\boldsymbol{a}_n)$. Since the regularizer (last term in (B.9)) is not differentiable, it is not well approximated by a linear function and, as a result, it fails to promote sparsity. To address this issue, *composite* algorithms decompose $h_t(\boldsymbol{a}_n)$ as $h_t(\boldsymbol{a}_n) = f_t^{(n)}(\boldsymbol{a}_n) + \Omega^{(n)}(\boldsymbol{a}_n)$, where $f_t^{(n)}(\boldsymbol{a}_n)$ is a convex loss function and $\Omega^{(n)}(\boldsymbol{a}_n)$ is a convex regularizer, and linearize only $f_t^{(n)}(\boldsymbol{a}_n)$. Algorithms of this family, which include *regularized dual averaging* (RDA) [110] and *composite objective mirror descent* (COMID) [67], solve the generic problem

$$\underset{\boldsymbol{a}_n}{\text{minimize}} \; \frac{1}{T_0} \sum_{t=0}^{T_0-1} \left[f_t^{(n)}(\boldsymbol{a}_n) + \Omega^{(n)}(\boldsymbol{a}_n) \right], \tag{B.10}$$

by linearizing $f_t^{(n)}(\boldsymbol{a}_n)$ but not $\Omega^{(n)}(\boldsymbol{a}_n)$. For instance, in COMID[6]

$$\boldsymbol{a}_n[t+1] = \arg\min_{\boldsymbol{a}_n} \left[\alpha_t \tilde{\nabla} f_t^{(n)^T}(\boldsymbol{a}_n[t]) (\boldsymbol{a}_n - \boldsymbol{a}_n[t]) + B_\psi(\boldsymbol{a}_n, \boldsymbol{a}_n[t]) + \alpha_t \Omega^{(n)}(\boldsymbol{a}_n) \right], \tag{B.11}$$

where $\tilde{\nabla} f_t^{(n)}(\boldsymbol{a}_n[t])$ is a subgradient of $f_t^{(n)}$ at point $\boldsymbol{a}_n[t]$ (that is, $\tilde{\nabla} f_t^{(n)}(\boldsymbol{a}[t]) \in \partial f_t^{(n)}(\boldsymbol{a}_n[t])$), $\alpha_t > 0$ is a step size, and $B_\psi(\boldsymbol{w}, \boldsymbol{v}) \triangleq \psi(\boldsymbol{w}) - \psi(\boldsymbol{v}) - \nabla \psi^T(\boldsymbol{v})(\boldsymbol{w} - \boldsymbol{v})$ is the so-called Bregman divergence associated with a ζ-strongly convex and continuously differentiable function ψ. The strong convexity condition means that $B_\psi(\boldsymbol{w}, \boldsymbol{v}) \geq (\zeta/2)\|\boldsymbol{w} - \boldsymbol{v}\|^2$, which motivates using $B_\psi(\boldsymbol{w}, \boldsymbol{v})$ as a surrogate of a distance between $\boldsymbol{w}$ and $\boldsymbol{v}$. Thus, the Bregman divergence in (B.11) penalizes updates $\boldsymbol{a}_n[t+1]$ lying far from the previous one $\boldsymbol{a}_n[t]$, which essentially smoothes the sequence of iterates.

Relative to each term in (B.10), the loss $f_t^{(n)}$ in (B.11) has been linearized but the regularizer $\Omega^{(n)}(\boldsymbol{a}_n)$ has been kept intact. When $\Omega^{(n)}(\boldsymbol{a}_n)$ is a sparsity-promoting regularizer, then the online estimate $\boldsymbol{a}[t+1]$ is therefore expected to be sparse.

In view of these appealing features, the algorithm proposed in Sec. B.3.1 builds upon COMID to address the problem of online causality graph identification from time series.

[6] This work focuses on COMID since, unlike RDA, there exist bounds for its regret for constant step size when the regularizer is not strongly convex.

B.3.1 Topology Identification via Sparse Online optimization

This section proposes *topology identification via sparse online optimization* (TISO), an online algorithm for the problem in Sec. B.2.2 that provides a causality graph estimate every time a new $\boldsymbol{y}[t]$ is processed. The key idea of this first algorithm is to refine the previous topology estimate with the information provided by the new data vector by means of a COMID update.

To this end, express h_t in (B.9) in the form $h_t(\boldsymbol{a}_n) = f_t^{(n)}(\boldsymbol{a}_n) + \Omega^{(n)}(\boldsymbol{a}_n)$ by setting

$$f_t^{(n)}(\boldsymbol{a}_n) = \ell_{t+P}^{(n)}(\boldsymbol{a}_n), \tag{B.12a}$$

$$\Omega^{(n)}(\boldsymbol{a}_n) = \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \,, \tag{B.12b}$$

for $t = 0, ..., T-P-1$. To choose $B_\psi(\boldsymbol{w}, \boldsymbol{v})$, note that (B.11) with $f_t^{(n)}(\boldsymbol{a}_n)$ and $\Omega^{(n)}(\boldsymbol{a}_n)$ given by (B.12) can be solved in closed form when $\psi(\cdot) = 1/2\|\cdot\|_2^2$. In that case, $B_\psi(\boldsymbol{w}, \boldsymbol{v}) = 1/2\|\boldsymbol{w} - \boldsymbol{v}\|_2^2$ and $\boldsymbol{a}_n[t+1]$ can be found via a modified *group soft-thresholding* operator, as detailed next. With these expressions, the TISO update after processing $\{\boldsymbol{y}[\tau]\}_{\tau=0}^{t}$ is

$$\boldsymbol{a}_n[t+1] = \arg\min_{\boldsymbol{a}_n} J_t^{(n)}(\boldsymbol{a}_n), \tag{B.13}$$

where

$$J_t^{(n)}(\boldsymbol{a}_n) \triangleq \boldsymbol{v}_n^\top[t](\boldsymbol{a}_n - \boldsymbol{a}_n[t]) + \frac{1}{2\alpha_t}\|\boldsymbol{a}_n - \boldsymbol{a}_n[t]\|_2^2 + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \tag{B.14}$$

and

$$\boldsymbol{v}_n[t] \triangleq \nabla \ell_t^{(n)}(\boldsymbol{a}_n[t]) = \boldsymbol{g}[t]\,(\boldsymbol{g}^\top[t]\,\boldsymbol{a}_n[t] - y_n[t]). \tag{B.15}$$

To solve (B.13) in closed form, expand the squared norm in (B.14) to obtain

$$\begin{aligned} J_t^{(n)}(\boldsymbol{a}_n) &\propto \frac{\|\boldsymbol{a}_n\|_2^2}{2\alpha_t} + \boldsymbol{a}_n^\top\big(\boldsymbol{v}_n[t] - \frac{1}{\alpha_t}\boldsymbol{a}_n[t]\big) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \\ &= \sum_{n'=1}^{N} \Big[\frac{1}{2\alpha_t}\|\boldsymbol{a}_{n,n'}\|_2^2 + \boldsymbol{a}_{n,n'}^\top\big(\boldsymbol{v}_{n,n'}[t] - \frac{1}{\alpha_t}\,\boldsymbol{a}_{n,n'}[t]\big) \quad + \lambda\,\|\boldsymbol{a}_{n,n'}\|_2\,\mathbb{1}\{n' \neq n\}\Big], \end{aligned} \tag{B.16}$$

where $\boldsymbol{v}_n[t] \triangleq [\,\boldsymbol{v}_{n,1}^\top[t], ..., \boldsymbol{v}_{n,N}^\top[t]\,]^\top$ and $\boldsymbol{v}_{n,n'}[t] \in \mathbb{R}^P \ \forall n'$. From (B.16), it can be observed that the updates in (B.13) can be computed separately for each group $n' = 1, ..., N$.

For $n' \neq n$, the n'-th subvector of $\boldsymbol{a}_n[t+1]$ (or n'-th *group*) can be expressed in terms of the so-called multidimensional shrinkage-thresholding operator [111] as:

$$\boldsymbol{a}_{n,n'}[t+1] = \boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\left[1 - \frac{\alpha_t\,\lambda}{\big\|\boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\big\|_2}\right]_+, \tag{B.17}$$

where $\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] \triangleq \boldsymbol{a}_{n,n'}[t] - \alpha_t \boldsymbol{v}_{n,n'}[t]$. Expression (B.17) is composed of two terms: whereas $\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t]$ is the result of performing a gradient-descent step in a direction that decreases the instantaneous loss $\ell_t^{(n)}(\boldsymbol{a}_n)$, the second term promotes *group sparsity* by setting $\boldsymbol{a}_{n,n'}[t+1] = \mathbf{0}_P$ for those groups n' with $\|\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t]\|_2 \leq \alpha_t \lambda$. Recalling that each vector $\boldsymbol{a}_{n,n'}$ corresponds to an edge in the estimated causality graph (see Sec. B.2.1), expression (B.17) indicates that only the relatively strong edges (i.e. causality relations) survive. In view of such a shrinkage operation, a larger λ will result in sparser estimates.

On the other hand, when $n' = n$, the n'-th subvector of $\boldsymbol{a}_n[t+1]$ in (B.13) is given by:

$$\boldsymbol{a}_{n,n'}[t+1] = \boldsymbol{a}_{n,n'}[t] - \alpha_t \boldsymbol{v}_{n,n'}[t] = \boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] \tag{B.18}$$

and, as intended, no sparsity is promoted on self-connections; see Sec. B.2.2. Combining (B.17) and (B.18), the estimate of the n'-th group at time $t+1$ is given by:

$$\boldsymbol{a}_{n,n'}[t+1] = \boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] \left[1 - \frac{\alpha_t \lambda \, \mathbb{1}\{n \neq n'\}}{\left\| \boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] \right\|_2} \right]_+ . \tag{B.19}$$

The performance of TISO depends on the choice of the step-size sequence $\{\alpha_t\}_t$, as discussed in Sec. B.4. The overall TISO algorithm is listed as **Algorithm 9**. It only requires $\mathcal{O}(N^2 P)$ memory entries to store the last P data vectors and the last estimate. On the other hand, each update requires $\mathcal{O}(N^2 P)$ arithmetic operations, which is in the same order as the number of parameters to be estimated. Thus, TISO can arguably be deemed a low-complexity algorithm.

Algorithm 9 Topology Identification via Sparse Online optimization (TISO)

Input: $\lambda, \{\alpha_t\}_t, \{\boldsymbol{y}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\boldsymbol{a}_n[\tau]\}_{n=1}^{N}, \tau = P+1, \ldots$
Initialization: $\boldsymbol{a}_n[P] = \mathbf{0}_{NP}, n = 1, \ldots, N$

for $t = P, P+1, \ldots$ **do**
  Receive data vector $\boldsymbol{y}[t]$
  Form $\boldsymbol{g}[t]$ via (B.4)
  for $n = 1, 2, \ldots, N$ **do**
    $\boldsymbol{v}_n[t] = (\boldsymbol{g}^\top[t]\, \boldsymbol{a}_n[t] - y_n[t])\boldsymbol{g}[t]$
    for $n' = 1, 2, \ldots, N$ **do**
      $\boldsymbol{a}^{\mathrm{f}}_{n,n'}[t] = \boldsymbol{a}_{n,n'}[t] - \alpha_t \boldsymbol{v}_{n,n'}[t]$
      Compute $\boldsymbol{a}_{n,n'}[t+1]$ via (B.19)
    end for
  end for
end for

The next section will build upon TISO to develop an algorithm with increased robustness to input variability.

B.3.2 Topology Identification via Recursive Sparse Online optimization

As seen in Sec. B.3.1, each update of TISO depends on the data through the *instantaneous* loss $\ell_t^{(n)}(\boldsymbol{a}_n[t])$, which quantifies the prediction error of the newly received vector $\boldsymbol{y}[t]$ when the VAR parameters $\mathcal{A}$ are given by the previous estimate $\boldsymbol{a}_n[t]$. Thus, the residual of predicting each data vector is used only in a single TISO update. Although this renders TISO a computationally efficient algorithm for online topology identification, it also increases sensitivity to noise and input variability. To this end, this section pursues an alternative approach at the expense of a moderate increase in computational complexity and memory requirements.

It is clear from (B.13) that $\boldsymbol{a}_n[t+1]$ is determined by $\boldsymbol{a}_n[t]$ and $\boldsymbol{v}_n[t]$. The latter incorporates the residual only at time t. The step size α_t controls how much variability in the input data propagates to the estimates $\{\boldsymbol{a}_n[t]\}_t$. When a diminishing step-size sequence is adopted, the influence of each new $\boldsymbol{y}[t]$ on the estimate becomes arbitrarily small, and the variability of the estimates fades away. However, decreasing sequences cannot be utilized when the application at hand demands tracking changes in the coefficients $\mathcal{A}$. In these settings, a constant step size $\alpha_t = \alpha$ is preferable. In such a scenario, a desire to reduce output variability would therefore force one to adopt a small α, but this would hinder TISO from tracking changes in the topology.

An approach to reduce output variability without sacrificing tracking capability will be developed next by drawing inspiration from the connections between TISO, the *least mean squares* (LMS) algorithm, and the *recursive least squares* (RLS) algorithm [112]. Indeed, observe that TISO generalizes LMS, which is recovered for $\lambda = 0$. To speed up convergence and reduce variability in the output of LMS, it is customary to resort to RLS, which accommodates the received data in a more sophisticated fashion, allowing to control the influence of each data vector on future estimates through forgetting factors.

Along these lines, the trick is to replace the *instantaneous* loss $\ell_t^{(n)}(\boldsymbol{a}_n)$ in (B.12) with a *running average* loss. To maintain tracking capabilities, a heavier weight is assigned to recent data using the exponential window customarily adopted by RLS. Specifically, consider setting $f_t^{(n)}(\boldsymbol{a}_n) = \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)$ in (B.12) with

$$\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) \triangleq \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \ell_\tau^{(n)}(\boldsymbol{a}_n), \tag{B.20}$$

where $\gamma \in (0,1)$ is the user-selected forgetting factor and $\mu = 1-\gamma$ is set to normalize the exponential weighting window, i.e., $\mu \sum_{\tau=0}^{\infty} \gamma^\tau = 1$.

Having specified a loss function, the next step is to derive the update equation. In a direct application of COMID to solve (B.10) with $f_t^{(n)}(\boldsymbol{a}_n) = \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)$, each iteration would involve the evaluation of the gradient of the $t-P+1$ terms of $\tilde{\ell}_t^{(n)}$. The computational complexity per iteration would grow with t and, therefore, the resulting updates would not make up a truly online algorithm according to the requirements expressed in Sec. B.2.3. To remedy this issue, the structure of (B.20) will be exploited next to develop an algorithm with constant memory and complexity per iteration. To this end, expand and rewrite

(B.20) to obtain

$$\begin{aligned}\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) =& \frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}\big(y_n^2[\tau]+\boldsymbol{a}_n^\top\boldsymbol{g}[\tau]\boldsymbol{g}^\top[\tau]\boldsymbol{a}_n-2y_n[\tau]\boldsymbol{g}^\top[\tau]\boldsymbol{a}_n\big)\\ =& \frac{1}{2}\boldsymbol{a}_n^\top\boldsymbol{\Phi}[t]\boldsymbol{a}_n-\boldsymbol{r}_n^\top[t]\boldsymbol{a}_n+\frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[\tau],\end{aligned}\tag{B.21}$$

where

$$\boldsymbol{\Phi}[t]\triangleq\mu\sum_{\tau=P}^{t}\gamma^{t-\tau}\boldsymbol{g}[\tau]\,\boldsymbol{g}^\top[\tau],\tag{B.22a}$$

$$\boldsymbol{r}_n[t]\triangleq\mu\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n[\tau]\,\boldsymbol{g}[\tau].\tag{B.22b}$$

The variables $\boldsymbol{\Phi}[t]$ and $\boldsymbol{r}_n[t]$ can be respectively thought of as a weighted sample autocorrelation matrix and a weighted sample cross-correlation vector. The key observation here is that, as occurs in RLS, these quantities can be updated recursively as $\boldsymbol{\Phi}[t]=\gamma\,\boldsymbol{\Phi}[t-1]+\mu\,\boldsymbol{g}[t]\,\boldsymbol{g}^\top[t]$ and $\boldsymbol{r}_n[t]=\gamma\,\boldsymbol{r}_n[t-1]+\mu\,y_n[t]\,\boldsymbol{g}[t]$. Noting that

$$\nabla\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)=\boldsymbol{\Phi}[t]\boldsymbol{a}_n-\boldsymbol{r}_n[t],\tag{B.23}$$

and letting $\tilde{\boldsymbol{v}}_n[t]\triangleq[\tilde{\boldsymbol{v}}_{n,1}^\top[t],\ldots,\tilde{\boldsymbol{v}}_{n,N}^\top[t]]^\top\triangleq\nabla\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n[t])$, the estimate $\tilde{\boldsymbol{a}}_n[t+1]$ after receiving $\{\boldsymbol{y}[\tau]\}_{\tau=0}^t$ becomes

$$\tilde{\boldsymbol{a}}_n[t+1]=\arg\min_{\tilde{\boldsymbol{a}}_n}\ \tilde{J}_t^{(n)}(\tilde{\boldsymbol{a}}_n),\tag{B.24}$$

where

$$\tilde{J}_t^{(n)}(\tilde{\boldsymbol{a}}_n)\triangleq\tilde{\boldsymbol{v}}_n^\top[t](\tilde{\boldsymbol{a}}_n-\tilde{\boldsymbol{a}}_n[t])+\frac{1}{2\alpha_t}\left\|\tilde{\boldsymbol{a}}_n-\tilde{\boldsymbol{a}}_n[t]\right\|_2^2+\lambda\sum_{\substack{n'=1\\n'\neq n}}^{N}\left\|\tilde{\boldsymbol{a}}_{n,n'}\right\|_2.\tag{B.25}$$

Proceeding similarly to Sec. B.3.1 yields the update

$$\tilde{\boldsymbol{a}}_{n,n'}[t+1]=\tilde{\boldsymbol{a}}_{n,n'}^{\mathrm{f}}[t]\left[1-\frac{\alpha_t\lambda\,\mathbb{1}\{n\neq n'\}}{\left\|\tilde{\boldsymbol{a}}_{n,n'}^{\mathrm{f}}[t]\right\|_2}\right]_+,\tag{B.26}$$

where $\tilde{\boldsymbol{a}}_{n,n'}^{\mathrm{f}}[t]\triangleq\tilde{\boldsymbol{a}}_{n,n'}[t]-\alpha_t\tilde{\boldsymbol{v}}_{n,n'}[t]$. Due to the recursive nature of the updates for $\boldsymbol{\Phi}[t]$ and $\boldsymbol{r}_n[t]$, the resulting algorithm is termed *Topology Identification via Recursive Sparse Online optimization* (TIRSO) and tabulated as **Algorithm 10**.

The choice of the step size affects the convergence properties of TIRSO, as analyzed in Sec. B.4. Regarding step size selection, natural choices include (i) constant step size, which is convenient in dynamic setups where changes in the coefficients $\mathcal{A}$ need to be tracked over time (see Theorem 5) but also gives rise to performance guarantees in static scenarios; (ii) diminishing step size, commonly in the form of $\mathcal{O}(1/\sqrt{t})$ or $\mathcal{O}(1/t)$ (see Theorem 4); or (iii) an adaptive step size that depends on the data, as discussed at the end of Sec. C.3.

Algorithm 10 Topology Identification via Recursive Sparse Online optimization (TIRSO)

Input: $\gamma, \mu, P, \lambda, \sigma^2, \{\alpha_t\}_t, \{\boldsymbol{y}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\tilde{\boldsymbol{a}}_n[t]\}_{n=1}^N, t = P+1, ...$
Initialization: $\tilde{\boldsymbol{a}}_n[P] = \mathbf{0}_{NP}, n = 1, ..., N$, $\boldsymbol{\Phi}[P-1] = \sigma^2 \boldsymbol{I}_{NP}$
$\boldsymbol{r}_n[t] = \mathbf{0}_{NP}, n = 1, ..., N$

1: **for** $t = P, P+1, \ldots$ **do**
2: Receive data vector $\boldsymbol{y}[t]$
3: Form $\boldsymbol{g}[t]$ via (B.4)
4: $\boldsymbol{\Phi}[t] = \gamma\, \boldsymbol{\Phi}[t-1] + \mu\, \boldsymbol{g}[t]\, \boldsymbol{g}^\top[t]$
5: **for** $n = 1, \ldots, N$ **do**
6: $\boldsymbol{r}_n[t] = \gamma\, \boldsymbol{r}_n[t-1] + \mu\, y_n[t]\, \boldsymbol{g}[t]$
7: $\tilde{\boldsymbol{v}}_n[t] = \boldsymbol{\Phi}[t]\, \tilde{\boldsymbol{a}}_n[t] - \boldsymbol{r}_n[t]$
8: **for** $n' = 1, 2, \ldots, N$ **do**
9: $\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t] = \tilde{\boldsymbol{a}}_{n,n'}[t] - \alpha_t \tilde{\boldsymbol{v}}_{n,n'}[t]$
10: Compute $\tilde{\boldsymbol{a}}_{n,n'}[t+1]$ via (B.26)
11: **end for**
12: **end for**
13: **end for**

Observe that $\boldsymbol{\Phi}[t]$ only needs to be updated once per observed sample t, whereas the vector $\boldsymbol{r}_n[t]$ need to be updated for each n at every t. The computational complexity is dominated by step 7, which is $\mathcal{O}(N^3P^2)$ operations per t. However, exploiting the group-sparse structure of $\tilde{\boldsymbol{a}}_n[t]$ may reduce the computation by disregarding the columns of $\boldsymbol{\Phi}[t]$ corresponding to the zero entries of $\tilde{\boldsymbol{a}}_n[t]$. If, for instance, the number of edges is $\mathcal{O}(N)$, then the complexity of TIRSO becomes $\mathcal{O}(N^2P^2)$ per t. Regarding memory complexity, TIRSO requires N^2P^2 memory positions to store $\boldsymbol{\Phi}[t]$ and N^2P positions to store $\{\boldsymbol{r}_n[t]\}_{n=1}^N$.

B.4 Theoretical Results

In this section, the performance of TISO and TIRSO is analyzed. The upcoming results will make use of one or more of the following assumptions:

A1. *Bounded samples:* There exists $B_y > 0$ such that $|y_n[t]|^2 \leq B_y\ \forall\, n, t$.

A2. *Bounded minimum eigenvalue of* $\boldsymbol{\Phi}[t]$*:* There exists $\beta_{\tilde{\ell}} > 0$ such that $\lambda_{\min}(\boldsymbol{\Phi}[t]) \geq \beta_{\tilde{\ell}},\ \forall\, t \geq P$.

A3. *Bounded maximum eigenvalue of* $\boldsymbol{\Phi}[t]$*:* There exists $L > 0$ such that $\lambda_{\max}(\boldsymbol{\Phi}[t]) \leq L,\ \forall\, t \geq P$.

A4. *Asymptotically invertible sample covariance:* There exists T_m and β such that

$$\lambda_{\min}\left(\frac{1}{t-P}\sum_{\tau=P}^{t} \boldsymbol{g}[\tau]\, \boldsymbol{g}^\top[\tau]\right) \geq \beta \quad \forall\, t \geq T_m. \tag{B.27}$$

Note that A1 entails no loss of generality in real-world applications, where data are bounded and thus B_y necessarily exists. A2 usually holds in practice unless the data is redundant, meaning that some time series can be obtained as a linear combination of the others. In general, the latter will not be the case e.g. if the data $\{\boldsymbol{y}[t]\}_t$ adheres to a continuous probability distribution, in which case $\boldsymbol{\Phi}[t]$ is positive definite for all $t \geq P$ with probability 1. A3 will also hold in practice since it can be shown that it is implied by A1. In particular, if A1 holds, then A3 holds with $L = PNB_y$. Similarly, A4 will also generally hold since it is a weaker version of A2.

Next, the asymptotic equivalence of the batch solutions for TISO and TIRSO is established.

B.4.1 Asymptotic Equivalence between TISO and TIRSO

To complement the arguments given in Sec. B.3.2 to support the decision of setting $f_t^{(n)}(\boldsymbol{a}_n) = \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n)$, which laid the grounds to develop TIRSO, we establish that the batch problems that TISO and TIRSO implicitly solve become asymptotically equivalent as $T \to \infty$. To this end, let $\boldsymbol{a}_n^*[T]$ denote the hindsight solution for TISO, which is given by

$$\boldsymbol{a}_n^*[T] = \arg\min_{\boldsymbol{a}_n} C_T(\boldsymbol{a}_n), \tag{B.28}$$

where

$$C_T(\boldsymbol{a}_n) \triangleq \frac{1}{T-P} \sum_{t=P}^{T-1} \Big[\ell_t^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \Big]. \tag{B.29}$$

Observe that (B.29) is identical to the objective in the batch criterion (B.6). Likewise, let $\tilde{\boldsymbol{a}}_n^*[T]$ denote the hindsight solution of TIRSO, which is given by

$$\tilde{\boldsymbol{a}}_n^*[T] = \arg\min_{\boldsymbol{a}_n} \ \tilde{C}_T(\boldsymbol{a}_n) \tag{B.30}$$

with

$$\tilde{C}_T(\boldsymbol{a}_n) \triangleq \frac{1}{T-P} \sum_{t=P}^{T-1} \Big[\tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) + \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{a}_{n,n'}\|_2 \Big]. \tag{B.31}$$

In this case, (B.31) no longer coincides with the objective in (B.6). Therefore, one can argue that the TIRSO algorithm is not pursuing the estimates that minimize the batch criterion (B.6). This idea is dispelled next by establishing the asymptotic equivalence between minimizing $\tilde{C}_T(\boldsymbol{a}_n)$ and minimizing $C_T(\boldsymbol{a}_n)$, since the latter is identical to (B.6).

Theorem 1. *Under assumption A1:*

1. *It holds for all* $\boldsymbol{a}_n$ *that* $\lim_{T\to\infty} |C_T(\boldsymbol{a}_n) - \tilde{C}_T(\boldsymbol{a}_n)| = 0$.

2. *It holds that* $\lim_{T\to\infty} \left| \inf_{\boldsymbol{a}_n} C_T(\boldsymbol{a}_n) - \inf_{\boldsymbol{a}_n} \tilde{C}_T(\boldsymbol{a}_n) \right| = 0.$

3. *If, additionally, assumption A2 holds, then* $\lim_{T\to\infty} \|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\|_2 = 0.$

Proof. See Appendix B.7 in the supplementary material. □

Theorem 1 essentially establishes not only that the TISO and TIRSO hindsight objectives are asymptotically the same but also that their minima and minimizers asymptotically coincide. Since the TISO hindsight objective equals the batch objective (B.6), it follows that the TIRSO hindsight objective asymptotically approaches the batch objective (B.6). This observation is very important since the regret analysis from Sec. B.4.2 will establish that the TISO and TIRSO estimates asymptotically match their hindsight counterparts.

B.4.2 Static Regret Analysis

This section characterizes the performance of TISO and TIRSO analytically. Specifically, it is shown that the sequences of estimates produced by these algorithms yield a sublinear static regret, which is a basic requirement in online optimization; see Sec. B.2.3. Broadly speaking, this property means that, on average and asymptotically, the online estimates perform as well as their hindsight counterparts.

A general definition of the regret metric is given in (B.8). Since the problem at hand is separable across nodes, it is natural to separately quantify the regret for each node. The total regret will be the sum of the regret for all nodes. Applying this idea and shifting the time index to simplify notation, one can replace $R_s[T_0]$ in (B.8) with $R_s^{(n)}[T_0+P-1]$, function h_t with $h_{t+P}^{(n)}$, and T_0 with $T-P+1$ to write the regret of TISO for the n-th node at time T as

$$R_s^{(n)}[T] \triangleq \sum_{t=P}^{T} \left[h_t^{(n)}(\boldsymbol{a}_n[t]) - h_t^{(n)}(\boldsymbol{a}_n^*[T]) \right], \tag{B.32}$$

where $h_t^{(n)}(\cdot) = \ell_t^{(n)}(\cdot) + \Omega^{(n)}(\cdot)$ and $\boldsymbol{a}_n^*[T]$ is defined in (B.28). For TIRSO, the regret for the n-th node is given by

$$\tilde{R}_s^{(n)}[T] \triangleq \sum_{t=P}^{T} \left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) \right], \tag{B.33}$$

where $\tilde{h}_t^{(n)}(\cdot) = \tilde{\ell}_t^{(n)}(\cdot) + \Omega^{(n)}(\cdot)$ and $\tilde{\boldsymbol{a}}_n^*[T]$ is defined in (B.30).

Since constant step size sequences allow tracking time-varying topologies, one could think of seeking a sublinear bound for the regret. However, it is easy to see (cf. (B.18) and (B.19) in the case of TISO) that the sequences of estimates in this case are generally noisy, unless the innovation process $\boldsymbol{u}[t]$ in (E.1) is $\mathbf{0}_N$. For this reason, a sublinear regret bound cannot be obtained for a constant α_t. However, it is possible to establish sublinear regret when the step size is "asymptotically constant," as described next.

The idea is to run the selected algorithm in time windows of exponentially increasing length with a step size that differs across windows but is constant within each one. Specifically, let the $(m+1)$-th window, $m = 1, \ldots, M$, comprise the time indices

$t_0 2^{m-1} < t \leq t_0 2^m$ for some user-selected $t_0 \geq P$. Set $\alpha_t = \alpha_{[m]}$ for those t satisfying $t_0 2^{m-1} < t \leq t_0 2^m$. The following result proves sublinear regret for TISO.

Theorem 2. *Let $\{\boldsymbol{a}_n[t]\}_{t=P}^T$ be generated by applying TISO (**Algorithm 9**) with step size $\alpha_t = \alpha_{[m]} = \mathcal{O}(1/\sqrt{t_0 2^{m-1}})$ in the window $t_0 2^{m-1} < t \leq t_0 2^m$, $m = 1, 2, \ldots$ Then, the regret of TISO under assumptions A1 and A4 is*

$$R_s^{(n)}[T] = \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \sqrt{T}\right), \tag{B.34}$$

where $B_{\boldsymbol{a}} = 1/\beta(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta B_y})$.

Proof. See Appendix B.8 in the supplementary material. □

Similarly, the regret of TIRSO is characterized as follows:

Theorem 3. *Let $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^T$ be generated by applying TIRSO (**Algorithm 10**) with step size $\alpha_t = \alpha_{[m]} = \mathcal{O}(1/\sqrt{t_0 2^{m-1}})$ in the window $t_0 2^{m-1} < t \leq t_0 2^m$, $m = 1, 2, \ldots$ Then, the regret of TIRSO under assumptions A1, A2, and A3, is*

$$R_s^{(n)}[T] = \mathcal{O}\left(LB_{\tilde{\boldsymbol{a}}}^2 \sqrt{T}\right), \tag{B.35}$$

where $B_{\tilde{\boldsymbol{a}}} \triangleq 1/\beta_{\tilde{\ell}}(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta_{\tilde{\ell}} B_y})$.

Proof. See Appendix B.10 in the supplementary material. □

Theorem 3 has the same form as Theorem 2 with the exception of (B.78), where the constant term multiplying $\sqrt{T}$ differs from the one in (B.34). However, it can be readily shown that $L \leq PNB_y$, which implies that TIRSO also satisfies (B.34).

To sum up, both TISO and TIRSO behave asymptotically in the same fashion and provide, on average, the same performance as the hindsight solution of TISO, which coincides with the batch solution in (B.6). The difference between TISO and TIRSO is, therefore, in the non-asymptotic regime, where TIRSO can track changes in the estimated graph more swiftly than TISO. This is at the expense of a slight increase in the number of arithmetic operations and required memory. Note, however, that TIRSO offers an additional degree of freedom through the selection of the forgetting factor γ. This enables the user to select the desired point in the trade-off between adaptability to changes and low variability in the estimates.

As demonstrated next, tighter regret bounds can be obtained when a diminishing step size sequence is adopted. Such sequences are of special interest when the VAR coefficients do not change over time. Even in this scenario, the application of online algorithms such as TISO or TIRSO is well-motivated when the number or dimension of the data vectors is prohibitively large to tackle with a batch algorithm.

Theorem 4. *Under assumptions A1, A2, and A3, let $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^T$ be generated by TIRSO (**Algorithm 10**) with $\alpha_t = 1/(\beta_{\tilde{\ell}} t)$. Then, the static regret of TIRSO satisfies*

$$\tilde{R}_s^{(n)}[T] \leq \frac{G_{\tilde{\ell}}^2}{2\beta_{\tilde{\ell}}} \left(\log(T - P + 1) + 1\right) + \frac{1}{2\alpha_{P-1}} B_{\tilde{\boldsymbol{a}}}^2, \tag{B.36}$$

where $G_{\tilde{\ell}} \triangleq (1 + \kappa_{\boldsymbol{\Phi}})\sqrt{PN} B_y$ with $\kappa_{\boldsymbol{\Phi}} = L/\beta_{\tilde{\ell}}$ and $B_{\tilde{\boldsymbol{a}}}$ is defined in Theorem 3.

Proof. See Appendix B.12 in the supplementary material. □

Next, we analyze the performance of TIRSO in dynamic environments.

B.4.3 Dynamic Regret Analysis of TIRSO

In this section, the performance of TIRSO is analyzed in dynamic settings. Specifically, a dynamic regret bound is derived for TIRSO, and its steady-state tracking error in dynamic scenarios is also discussed. To characterize the performance of TIRSO in dynamic setups, the dynamic regret is defined as:

$$\tilde{R}_d^{(n)}[T] \triangleq \sum_{t=P}^{T} \big[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t])\big], \tag{B.37}$$

where $\tilde{\boldsymbol{a}}_n[t]$ is the TIRSO estimate and $\tilde{\boldsymbol{a}}_n^{\circ}[t] = \arg\min_{\tilde{\boldsymbol{a}}_n} \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n)$. The dynamic regret in (E.12) compares the estimate $\tilde{\boldsymbol{a}}_n[t]$ with $\tilde{\boldsymbol{a}}_n^{\circ}[t]$ in terms of the metric $\tilde{h}_t^{(n)}(\cdot)$. As opposed to $\tilde{\boldsymbol{a}}_n^{\circ}[t]$, estimate $\tilde{\boldsymbol{a}}_n[t]$ does not "know" $\tilde{h}_t^{(n)}(\cdot)$ since $\tilde{\boldsymbol{a}}_n[t]$ is obtained from $\{\boldsymbol{y}[\tau]\}_{\tau<t}$ whereas $\tilde{h}_t^{(n)}(\cdot)$ depends on both $\{\boldsymbol{y}[\tau]\}_{\tau<t}$ and $\boldsymbol{y}[t]$. This means that the dynamic regret captures the ability of an algorithm to attain small *future* residuals. Furthermore, note that comparing with $\tilde{\boldsymbol{a}}_n^{\circ}[t]$ is highly meaningful in the present case since, by definition, $\tilde{\boldsymbol{a}}_n^{\circ}[t] = \arg\min_{\tilde{\boldsymbol{a}}_n} \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \ell_{\tau}^{(n)}(\tilde{\boldsymbol{a}}_n) + \lambda \sum_{n'=1, n'\neq n}^{N} \|\tilde{\boldsymbol{a}}_{n,n'}\|_2$, which therefore minimizes a version of the batch (B.6) or hindsight (B.30) objectives where the more recent residuals are weighted more heavily. Thus, $\tilde{\boldsymbol{a}}_n^{\circ}[t]$ constitutes a significant estimator in dynamic setups and therefore the dynamic regret also quantifies the ability of an estimator to track changes.

It can be easily shown that the static regret is upper-bounded by the dynamic regret. The dynamic regret in (E.12) would coincide with the static regret if $\tilde{\boldsymbol{a}}_n^{\circ}[t]$ were replaced with $\arg\min_{\tilde{\boldsymbol{a}}_n} \sum_{t=P}^{T} \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n)$. Attaining a low dynamic regret is therefore more challenging because the estimator under consideration is compared with a *time-varying* reference. This implies that a sublinear dynamic regret may not be attained if this time-varying reference changes too rapidly, which generally occurs when the tracked parameters vary too quickly. For this reason, the dynamic regret is commonly upper-bounded in terms of the cumulative distance between two consecutive instantaneous optimal solutions, known as *path length*:

$$W^{(n)}[T] \triangleq \sum_{t=P+1}^{T} \|\tilde{\boldsymbol{a}}_n^{\circ}[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t-1]\|_2 . \tag{B.38}$$

Next, we bound the dynamic regret of TIRSO.

Theorem 5. *Under assumptions A1, A2, and A3, let $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^{T}$ be generated by TIRSO (**Algorithm 10**) with a constant step size $\alpha \in (0, 1/L]$. If there exists σ such that*

$$\|\tilde{\boldsymbol{a}}_n^{\circ}[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t-1]\|_2 \leq \sigma, \ \forall\, t \geq P+1, \tag{B.39}$$

then the dynamic regret of TIRSO satisfies:

$$\tilde{R}_d^{(n)}[T] \leq \frac{1}{\alpha\beta_{\tilde{\ell}}} \Big((1+\kappa_{\boldsymbol{\Phi}}) \sqrt{PN} B_y + \lambda N \Big) \big(\|\tilde{\boldsymbol{a}}_n^{\circ}[P]\|_2 + W^{(n)}[T] \big),$$

where $\kappa_{\boldsymbol{\Phi}} \triangleq L/\beta_{\tilde{\ell}}$.

Proof. See Appendix B.13 in the supplementary material. □

Several remarks about Theorem 5 are in order. If the path length $W^{(n)}[T]$ is sublinear in T, then the dynamic regret is also sublinear in T. When the path length is not sublinear, the dynamic regret may not be sublinear, but we can still bound the *steady-state error* under certain conditions:

Theorem 6. *Under assumptions A1, A2, and A3, let* $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^{T}$ *be generated by TIRSO (**Algorithm 10**) with a constant step size* $\alpha \in (0, 1/L]$. *If there exists* σ *such that* (E.65) *holds, then*

$$\limsup_{t\to\infty} \|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]\|_2 \leq \frac{\sigma}{\alpha\beta_{\tilde{\ell}}}. \tag{B.40}$$

Proof. Following similar arguments as in the proof of Theorem 5, (B.40) follows by applying [70, Lemma 4]. □

This theorem establishes that the steady-state error incurred by TIRSO with $\alpha \in (0, 1/L]$ in dynamic scenarios eventually becomes bounded, which shows its tracking capability in time-varying environments. If $\alpha = 1/L$, then the upper bound on the steady-state error becomes $\sigma\kappa_{\boldsymbol{\Phi}}$, where $\kappa_{\boldsymbol{\Phi}} \triangleq L/\beta_{\tilde{\ell}}$ is an upper bound on the condition number of $\boldsymbol{\Phi}[t]$, $t \geq P$. This clearly agrees with intuition. In practice, one may not know the value of L and therefore selecting an α guaranteed to be in $(0, 1/L]$ would not be possible. In those cases, it makes sense to compute a running approximation of L given by $\hat{L}_t = \max_{P\leq\tau\leq t} \lambda_{\max}(\boldsymbol{\Phi}[\tau])$ and adopt the approximately constant step size $\alpha_t = c/\hat{L}_t$, where $c \in (0, 1]$. However, in setups where the true VAR parameters change over time, the max operation may lead the algorithm to use an overly pessimistic approximation of L. Thus, it may be preferable to directly adopt the *adaptive step size* $\alpha_t = c/\lambda_{\max}(\boldsymbol{\Phi}[t])$, as analyzed in Sec. B.5.

Remark. None of the algorithms and analytical results in this paper require any probabilistic assumption or mention to probability theory. This is because these results establish performance guarantees for the proposed online algorithms relative to the batch estimator or hindsight solutions. If one wished to obtain performance guarantees in terms of *probabilistic* metrics, such as consistency of the estimators, probabilistic assumptions would of course be required. For example, when $\lambda = 0$, the batch estimator in (B.3) boils down to the ordinary least squares estimator, which is consistent if the VAR process is stable and the noise is standard white [43, Lemma 3.1]. When $\lambda > 0$, consistency of (B.3) is discussed in [66]. Remarkably, consistency of the VAR coefficient estimates is not enough to ensure the correct identification of the true graph. Theorem 1 in [66] provides conditions that depend on the true VAR parameters that guarantee that the graph is successfully recovered.

B.5 Numerical Results and Analysis

Simulation tests for the proposed algorithms are performed on both synthetic and real data. All code will be made public at the authors' websites.

The proposed algorithms are evaluated based on the performance metrics described next,

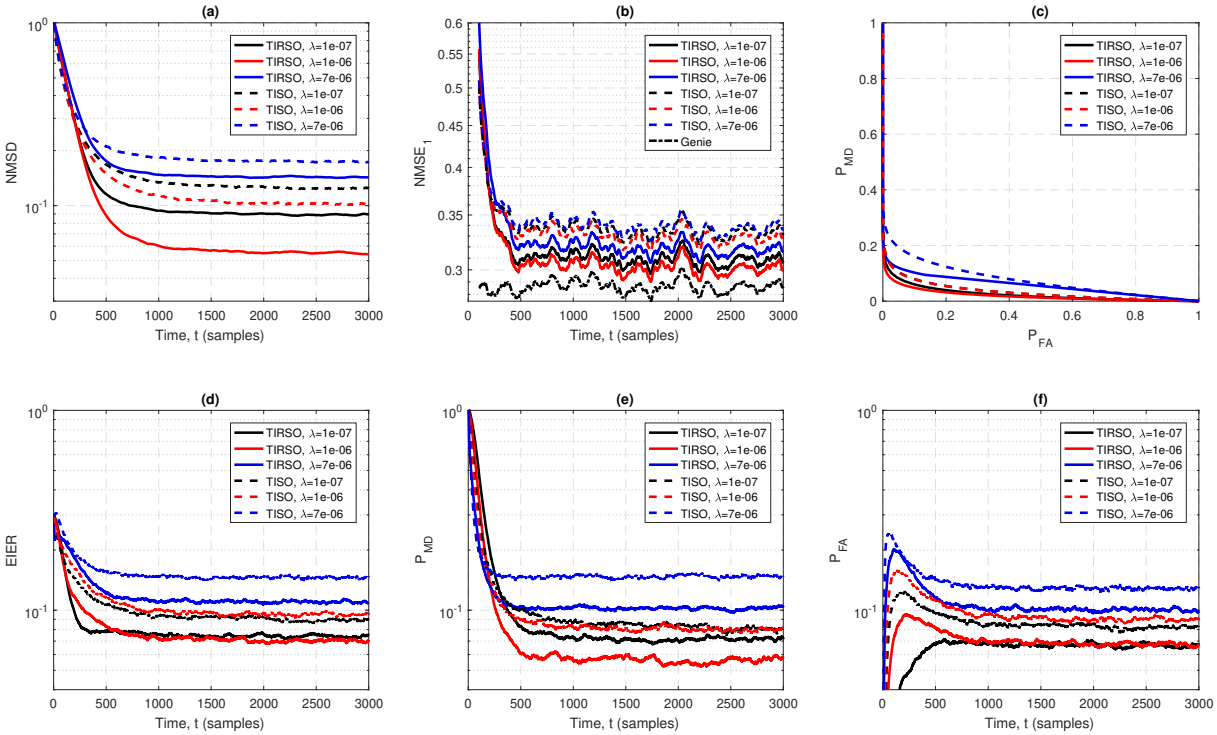


Figure B.2: Performance of TISO and TIRSO on stationary time series for different degrees of sparsity-promoting regularization ($N = 12$, $P = 2$, $p_e = 0.2$, $\sigma_u = 0.005$, $\gamma = 0.99$, $T = 3000$, $T_1 = 500$, $T_2 = 3000$, 300 Monte Carlo runs).

where expectations are approximated by the Monte Carlo method. For synthetic-data experiments, the normalized mean square deviation

$$\mathrm{NMSD}[t] \triangleq \frac{\mathbb{E}\big[\sum_{n=1}^{N}\|\hat{\boldsymbol{a}}_n[t]-\boldsymbol{a}_n^{\mathrm{true}}[t]\|_2^2\big]}{\mathbb{E}\big[\sum_{n=1}^{N}\|\boldsymbol{a}_n^{\mathrm{true}}[t]\|_2^2\big]} \tag{B.41}$$

measures the difference between the estimates $\{\hat{\boldsymbol{a}}_n[t]\}_t$ and the (possibly time-varying) true VAR coefficients $\{\boldsymbol{a}_n^{\mathrm{true}}[t]\}_t$. The ability to detect edges of the true VAR-causality graph is assessed using the probability of miss detection

$$P_{\mathrm{MD}}[t] \triangleq \frac{\sum_{n\neq n'}\mathbb{E}\left[\mathbb{1}\{\|\hat{\boldsymbol{a}}_{n,n'}[t]\|_2<\delta\}\mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2\neq 0\}\right]}{\sum_{n\neq n'}\mathbb{E}\left[\mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2\neq 0\}\right]}$$

for a given threshold δ, which is the probability of not identifying an edge that actually exists, and the probability of false alarm

$$P_{\mathrm{FA}}[t] \triangleq \frac{\sum_{n\neq n'}\mathbb{E}\left[\mathbb{1}\{\|\hat{\boldsymbol{a}}_{n,n'}[t]\|_2\geq\delta\}\mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2= 0\}\right]}{\sum_{n\neq n'}\mathbb{E}\left[\mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2= 0\}\right]},$$

which is the probability of detecting an edge that does not exist. Another relevant metric is the *edge identification error rate* (EIER), which measures how many edges are misidentified relative to the number of possible edges [113]:

$$\mathrm{EIER}[t] = \frac{1}{N(N-1)}\sum_{n'\neq n}\mathbb{E}\big[\big|\mathbb{1}\{\|\hat{\boldsymbol{a}}_{n,n'}[t]\|_2\geq\delta\} - \mathbb{1}\{\|\boldsymbol{a}_{n,n'}\|_2\neq 0\}\big|\big]. \tag{B.42}$$

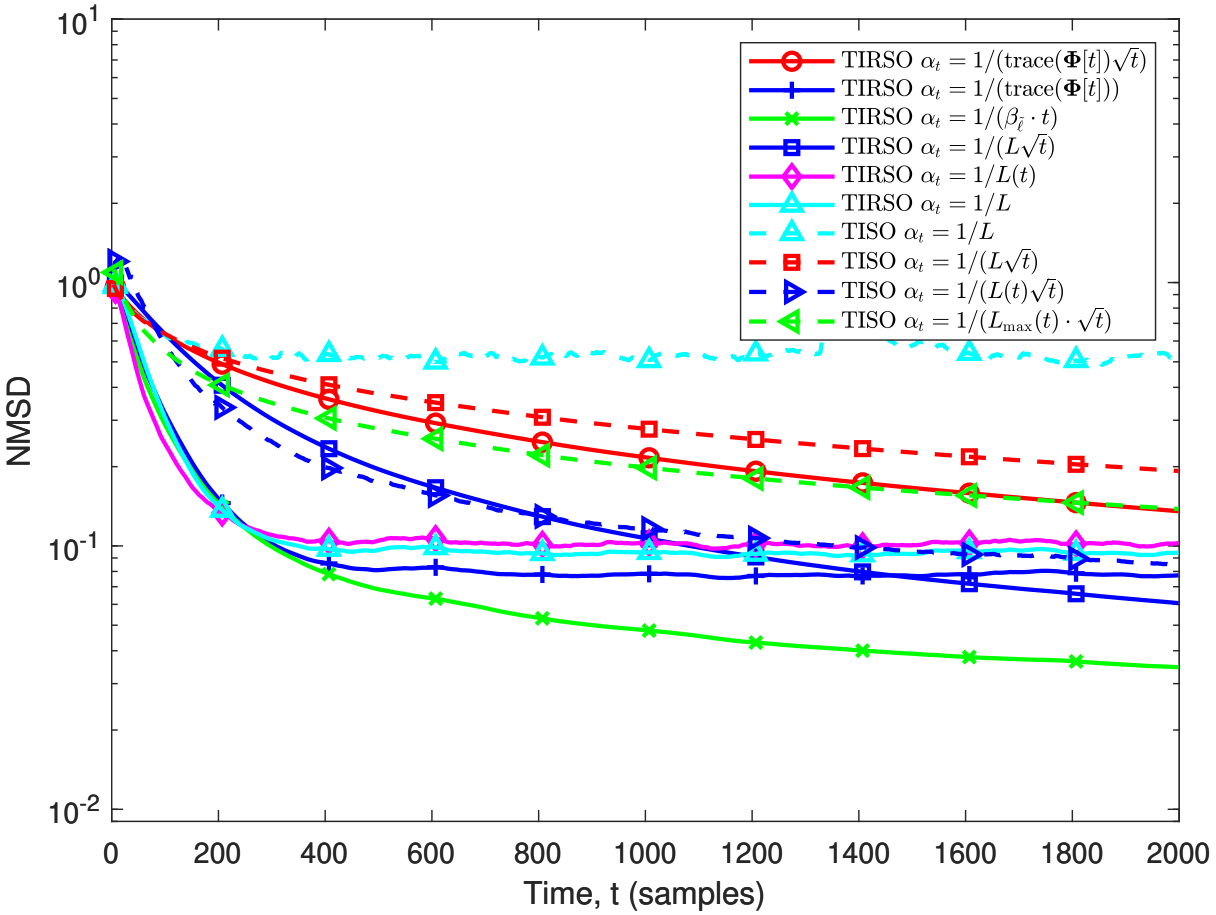


Figure B.3: NMSD vs. time: comparison of TISO and TIRSO for various options of step sizes ($N = 10$, $P = 3$, $p_e = 0.2$, $\sigma_u = 0.1$, $\gamma = 0.99$, $\lambda = 8 \times 10^{-4}$, $T = 2000$, 50 Monte Carlo runs). Moreover, $L_{\max}(t) := \max_{\tau=1}^{t} L(\tau)$.

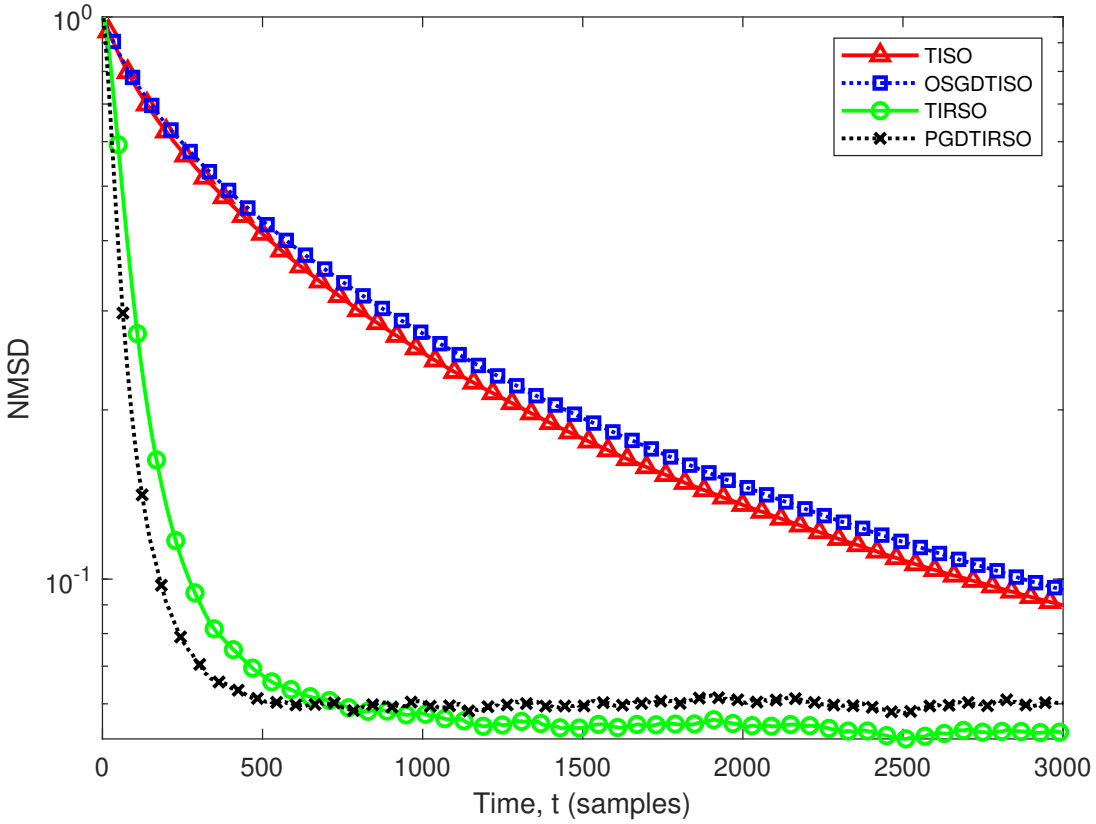


Figure B.4: NMSD vs. time: comparison of TISO and TIRSO with other algorithms. ($N = 10$, $P = 2$, $p_e = 0.2$, $\sigma_u = 0.01$, $\alpha_t = 0.1/L$, $\gamma = 0.99$, $T = 3000$, $K_{\text{PGD}} = 5$, 200 Monte Carlo runs). The parameter λ for each algorithm is selected based on minimum NMSD.

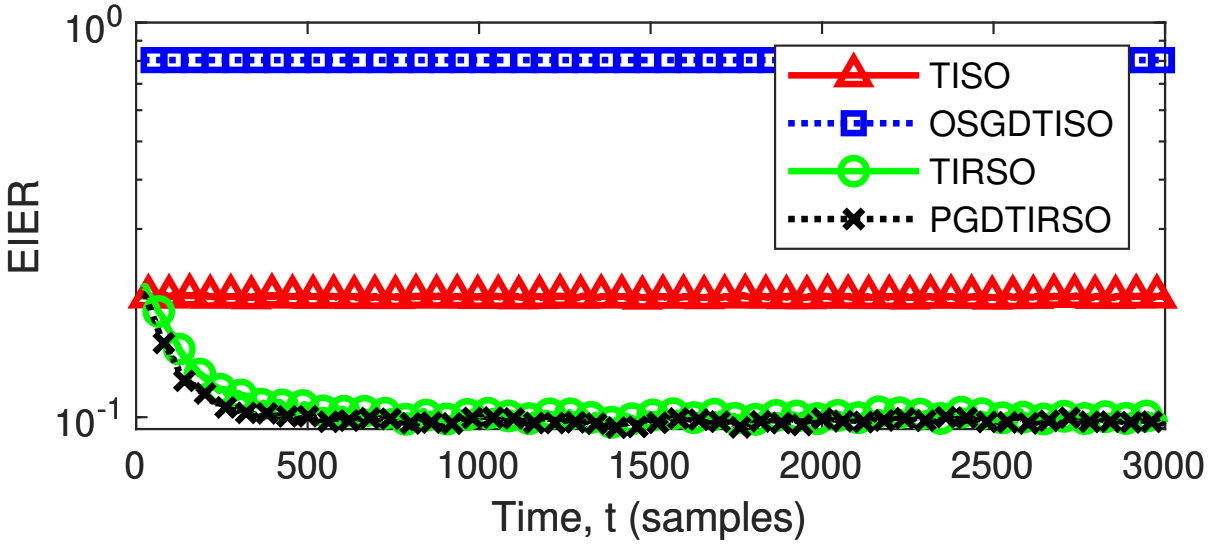


Figure B.5: EIER vs time for $\delta = 0$, same parameters as of Fig. B.4. The parameter λ for each algorithm is selected based on minimum EIER.

Note that self-loops are excluded in these metrics. To quantify the forecasting performance, define recursively the h-step ahead predictor given $\{\boldsymbol{y}[\tau]\}_{\tau\leq t}$ as:

$$\hat{\boldsymbol{y}}[t+h|t] \triangleq \sum_{p=1}^{P} \hat{\mathbf{A}}_p[t]\hat{\boldsymbol{y}}\left[t+h-p|t\right], \tag{B.43}$$

where $\{\hat{\mathbf{A}}_p[t]\}_{p=1}^{P}$ are the estimated VAR coefficients at time t and $\hat{\boldsymbol{y}}[t+j|t] = \boldsymbol{y}[t+j]$ for $j \leq 0$. The h-step normalized mean square error is given by

$$\mathrm{NMSE}_h[t] = \frac{\mathbb{E}\left[\, \|\boldsymbol{y}[t+h] - \hat{\boldsymbol{y}}[t+h|t]\|_2^2 \,\right]}{\mathbb{E}\left[\|\boldsymbol{y}[t+h]\|_2^2\right]}. \tag{B.44}$$

The values of all parameters involved in the experiments are listed in the captions and legends of the figures.

B.5.1 Synthetic Data Tests

Throughout this section, unless otherwise stated, the expectations in (E.80) to (B.44) are taken with respect to realizations of the graph, VAR parameters, and innovation process $\boldsymbol{u}[t]$. Similarly, the step size is set to $\alpha_t = 1/(4\lambda_{\max}(\boldsymbol{\Phi}[t]))$; see Sec. C.3. The regularization parameter is selected to approximately minimize NMSD.

B.5.1.1 Stationary VAR Processes

An Erdős-Rényi random graph is generated with edge probability p_e and self-loop probability 1. This graph determines which entries of the matrices $\{\boldsymbol{A}_p\}_{p=0}^{P}$ are zero. The rest of entries are drawn i.i.d. from a standard normal distribution. Matrices $\{\boldsymbol{A}_p\}_{p=0}^{P}$ are scaled down afterwards by a constant that ensures that the VAR process is stable [43]. The innovation process samples are drawn independently as $\boldsymbol{u}[t] \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 \boldsymbol{I}_N)$.

The first experiment analyzes TISO and TIRSO in a stationary setting. Figs. B.2(a) and B.2(b) depict the NMSD and NMSE_1 for three different values of λ. As a benchmark, Fig. B.2(b) includes the NMSE_1 of the *genie-aided predictor*, obtained from (B.43) after replacing $\hat{\mathbf{A}}_p$ with $\boldsymbol{A}_p$. It is observed that $\lambda = 10^{-6}$ yields a better NMSD and NMSE_1 than lower and higher values of λ. This corroborates the importance of promoting sparse solutions, as done in TISO and TIRSO. Furthermore, as expected, TIRSO generally converges faster than TISO. Fig. B.2(c) shows the receiver operating characteristic (ROC) curve, composed of pairs $(\mathrm{P_{FA}}, \mathrm{P_{MD}})$ for different values of the threshold δ. The values of these pairs are obtained by respectively averaging $\mathrm{P_{FA}}[t]$ and $\mathrm{P_{MD}}[t]$ over time in the interval $[T_1, T_2]$. Remarkably, both TISO and TIRSO can simultaneously attain $\mathrm{P_{FA}}$ and $\mathrm{P_{MD}}$ below 10%. This ability to satisfactorily detect edges is further investigated in Figs. B.2(d-f), where δ is set for each algorithm so that $\mathrm{P_{FA}}[t]$ and $\mathrm{P_{MD}}[t]$ have the same average over the time interval $[T_1, T_2]$.

Fig. B.3 analyzes different step size sequences. Because the true VAR parameters remain constant, the diminishing sequence yields the best performance; see Theorem 4. Besides, TISO and TIRSO are compared with benchmarks in Fig. B.4, namely online subgradient descent (OSGD) and proximal gradient descent (PGD). The former obtains a mini-

mizer for (B.6) in an online fashion (labeled as OSGDTISO since it uses the same information as TISO at each iteration). The latter approximates $\tilde{\boldsymbol{a}}_n^\circ[t] = \arg\min_{\tilde{\boldsymbol{a}}_n} \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n)$ by using the (batch) algorithm PGD for K_{PGD} iterations over $\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n)$ (labeled as PGDTIRSO since it uses the same information as TIRSO at each iteration). Fig. B.4 shows that TISO outperforms OSGDTISO in terms of NMSD, and TIRSO eventually attains better NMSD level than PGDTIRSO. Note that the computational complexity of PGDTIRSO is significantly larger than the complexity of TIRSO. Although the NMSD of TISO in Fig. B.4 is close to that of OSGD, a more in-depth study reveals that the former yields sparse iterates without any thresholding; moreover, TIRSO offers a significantly improved edge-detection performance (EIER), see Fig. B.5. Fig. B.6 compares the true (left) and recovered (right) graphs via TIRSO and TISO by thresholding the average of the estimated VAR coefficients across the intervals $[k/(3T), (k+1)/(3T)]$, $k = 0, 1, 2$. The threshold δ is selected to detect $p_e(N^2 - N)$ edges. Note that this is displayed for a single graph and realization of the VAR process; in other words, this is not a Monte Carlo experiment. It is observed that both TIRSO and TISO can identify the true graph quite accurately and approximate the true VAR coefficients soon afterwards.

B.5.1.2 Non-stationary VAR Processes

The next experiment analyzes TISO and TIRSO when $\boldsymbol{y}[t]$ is a (non-stationary) smooth-transition VAR process [114, Ch. 18] $\boldsymbol{y}[t] = \sum_{p=1}^{P} \big(\boldsymbol{A}_p + s_f[t](\boldsymbol{B}_p - \boldsymbol{A}_p)\big)\boldsymbol{y}[t-p] + \boldsymbol{u}[t]$. The signal $s_f[t]$ determines the transition profile from a VAR model with parameters $\{\boldsymbol{A}_p\}_p$ to a VAR model with parameters $\{\boldsymbol{B}_p\}_p$. In this experiment, $s_f[t] = 1 - \exp(-\kappa([t - T_B]_+)^2)$, where $\kappa > 0$ controls the transition speed and T_B denotes transition starting instant. Over an Erdős-Rényi random graph, $\{\boldsymbol{A}_p\}$ and $\{\boldsymbol{B}_p\}$ are generated independently as in Sec. B.5.1.1. It is easy to show that the coefficients $\boldsymbol{A}_p + s_f[t](\boldsymbol{B}_p - \boldsymbol{A}_p)$ yield a *stable* VAR process for all t.

Figs. B.7(a) and B.7(b) illustrate the influence of the forgetting factor γ, of critical importance in non-stationary setups. TISO and TIRSO are seen to satisfactorily estimate and track the model coefficients. As intuition predicts, the lower γ is, the more rapidly TIRSO can adapt to changes, but after a sufficiently long time after the transition, a higher γ is preferred.

Finally, to demonstrate that TISO and TIRSO successfully leverage sparsity to track *time-varying* topologies, Fig. B.8 illustrates an approximately optimal point in the trade-off of selecting λ.

B.5.2 Real-Data Tests

The real data is taken from Lundin's offshore oil and gas (O&G) platform Edvard-Grieg[7]. Each node corresponds to a temperature, pressure, or oil-level sensor placed in the decantation system that separates oil, gas, and water. The measured time series are physically coupled due to the pipelines connecting the system parts and due to the control systems. Hence, causal relations among time series are expected. Topology identification

[7] https://www.lundin-petroleum.com/operations/production/norway-edvard-grieg

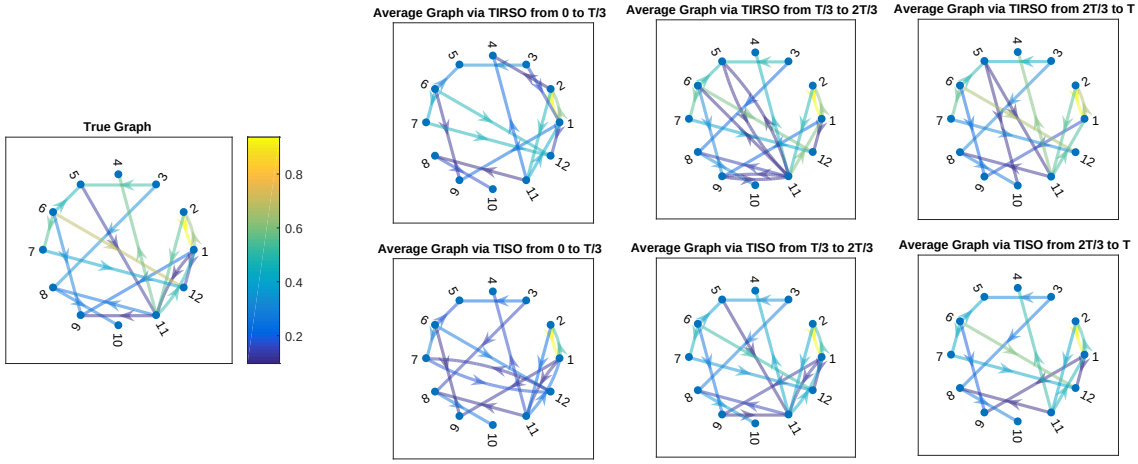


Figure B.6: True and recovered graphs ($N = 12$, $P = 2$, $p_e = 0.2$, $\sigma_u = 0.005$, $\gamma = 0.98$, $\lambda = 10^{-6}$, $T = 600$).

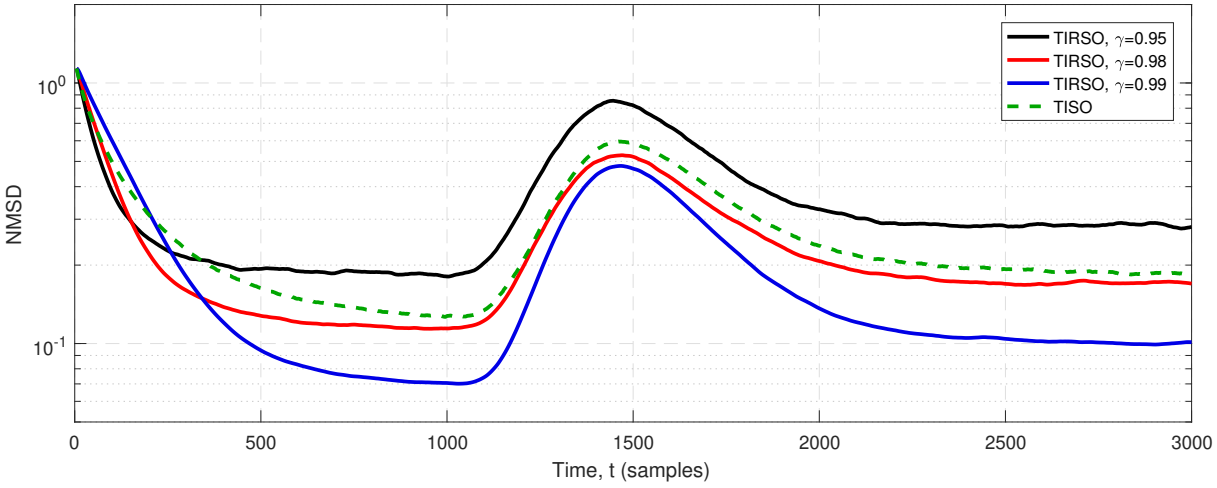
is motivated to forecast the short-term future state of the system and to unveil dependencies that cannot be detected by simple inspection. All time series are resampled to a common set of equally-spaced sampling instants using linear interpolation. Since the data was quantized and compressed using a lossy scheme, a significant amount of noise is expected. Each time series is normalized to have zero mean and unit sample standard deviation. Here, the step size is set to $\alpha_t = 1/(\lambda_{\max}(\boldsymbol{\Phi}[t]))$ and the NMSE is defined as $\text{NMSE}_h = 1/(\sum_t \|\boldsymbol{y}[t+h]\|_2^2) \sum_t \|\boldsymbol{y}[t+h] - \hat{\boldsymbol{y}}[t+h|t]\|_2^2$.

Fig. B.9 shows the NMSE_h vs. the *prediction horizon* h for the time series in the data set. The temperature, pressure, and oil level time series are respectively denoted by T, P, and L and an identifying index. As expected, the prediction error increases with h. The NMSE ranges from 10^{-4} to 1 due to the different predictability of each time series.

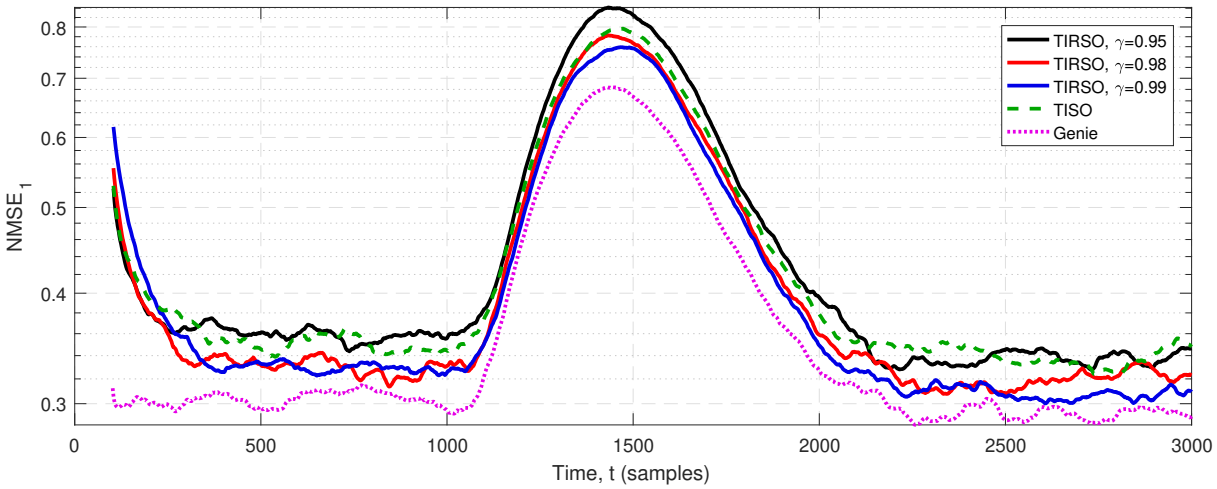
Fig. B.10 presents the graph obtained by thresholding the average coefficient estimates over a three-hour duration. The threshold is such that the number of reported edges is $4N$. Self-loops are omitted for clarity, and arrow colors encode edge weights. It is observed that most identified edges connect sensors within each subsystem.

B.6 Conclusions

Two online algorithms were proposed for identifying and tracking VAR-causality graphs from time series. These algorithms sequentially accommodate data and refine their sparse topology estimates accordingly. The proposed algorithms offer complementary benefits: whereas TISO is computationally simpler, TIRSO showcases improved tracking behavior. Performance is assessed theoretically and empirically. Asymptotic equivalence of the hindsight solutions of the proposed algorithms is established and sublinear regret bounds

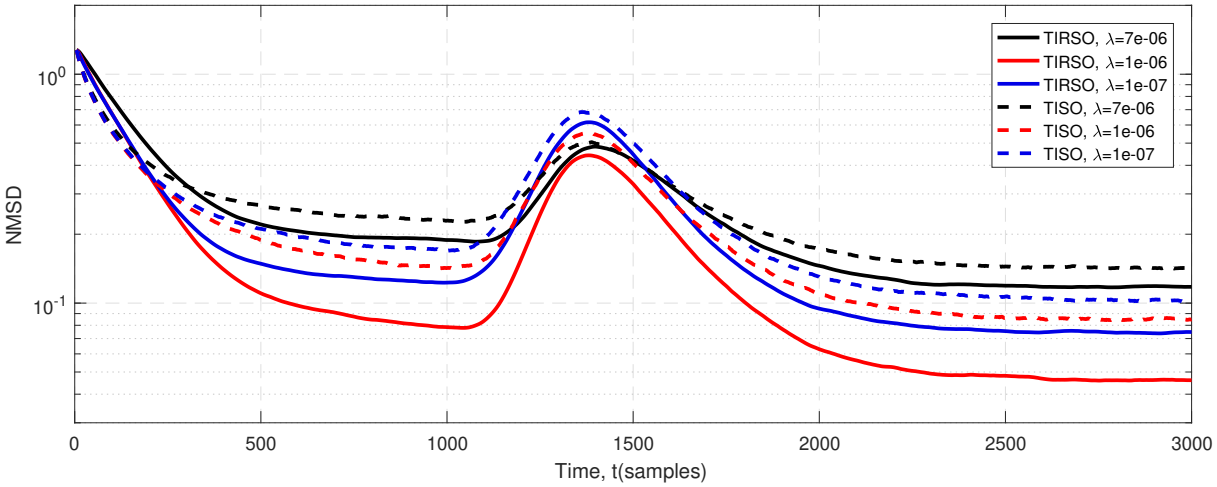


(a) NMSD vs.time

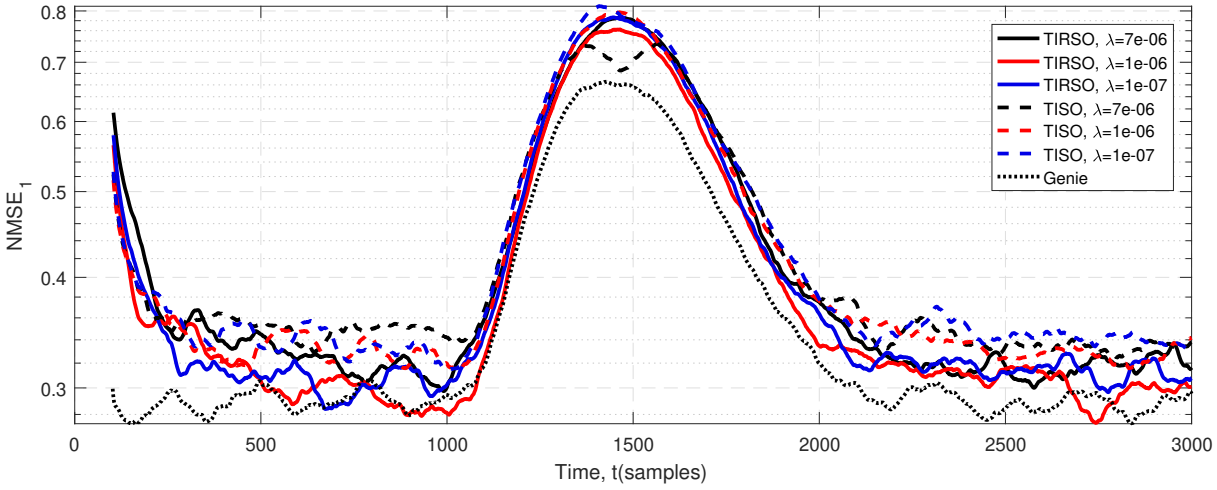


(b) NMSE_1 vs. time

Figure B.7: Effect of the forgetting factor on the performance in a smooth-transition VAR model ($\kappa = 0.99$, $T_B = 1000$, $N = 12$, $P = 2$, $p_e = 0.2$, 300 Monte Carlo runs).



(a) NMSD vs.time



(b) NMSE_1 vs. time

Figure B.8: Effect of the regularization parameter on the performance in a smooth-transition VAR model ($\kappa = 0.99$, $T_B = 1000$, $N = 12$, $T = 3000$ $P = 2$, $p_e = 0.2$, $\gamma = 0.98$, 200 Monte Carlo runs).

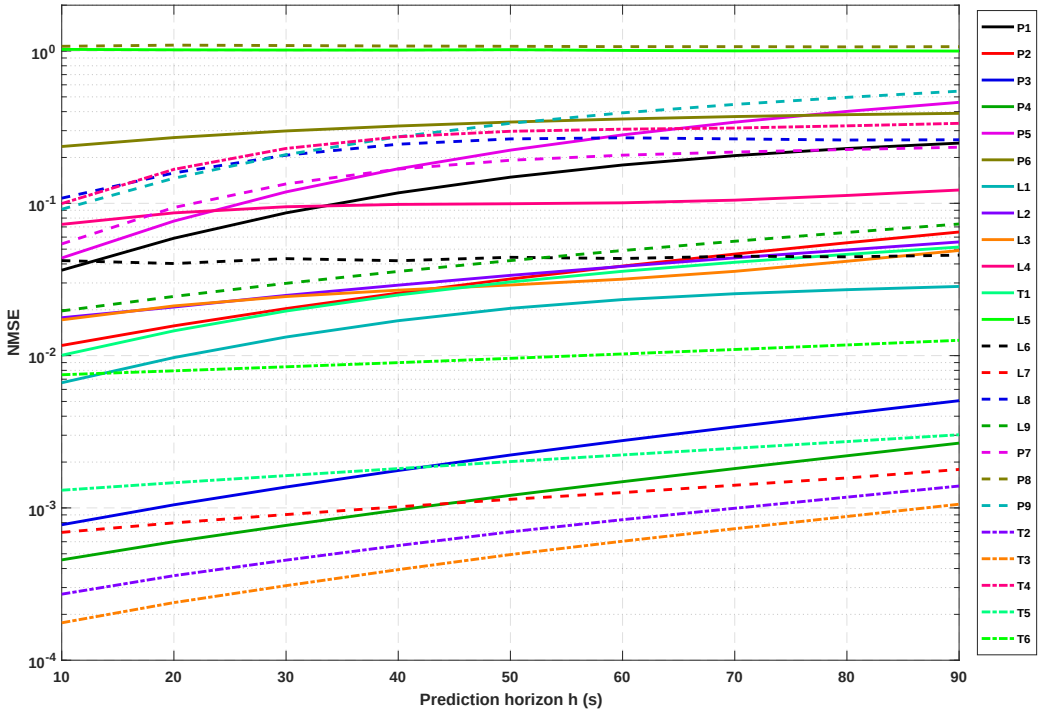


Figure B.9: Prediction NMSE vs. prediction horizon for individual variables of oil, gas, and water separation system. TIRSO is used with $P = 8$, $\gamma = 0.9$, $T = 4$ hours, sampling interval = 10 s. The parameter λ is selected based on minimum average NMSE.

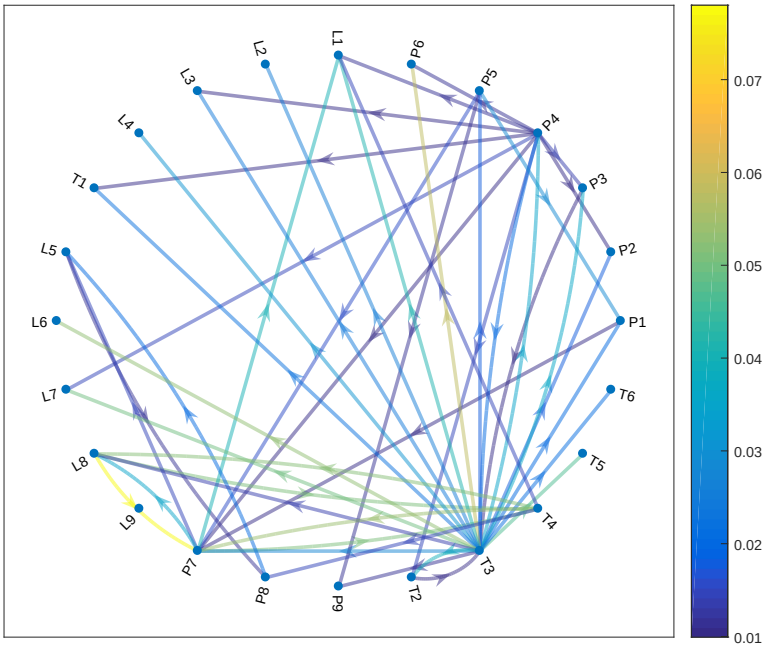


Figure B.10: The estimated topology of a subset of the variables. The sampling interval is set to 10 seconds. The topology is obtained via TIRSO with $\gamma = 0.9$, $T = 3$ hours, and $P = 8$. The parameter λ is selected based on minimum average NMSE.

are derived. Experiments with synthetic and real data validate the conclusions of the theoretical analysis. Future directions include explicitly modeling the variations in the VAR coefficients, possibly along the lines of [89, 90, 91], as well as identifying topologies whose adjacency matrix has a low-rank plus sparse structure along the lines of [92] to account for clusters.

Supplementary Material

B.7 Proof of Theorem 1

The first step is to rewrite (B.31) to be able to obtain a simple expression for $C_T(\boldsymbol{a}_n) - \tilde{C}_T(\boldsymbol{a}_n)$. To this end, substitute (B.20) into (B.31) and exchange the order of the summations to obtain

$$\begin{aligned}\tilde{C}_T(\boldsymbol{a}_n) &= \frac{1}{T-P}\sum_{\tau=P}^{T-1}\Big[\sum_{t=\tau}^{T-1}\gamma^{t-\tau}\mu\,\ell_\tau^{(n)}(\boldsymbol{a}_n) + \lambda\sum_{\substack{n'=1\\n'\neq n}}^{N}\|\boldsymbol{a}_{n,n'}\|_2\Big] \\ &= \frac{1}{T-P}\sum_{\tau=P}^{T-1}\theta_{\tau,T}\,\mu\,\ell_\tau^{(n)}(\boldsymbol{a}_n) + \lambda\sum_{\substack{n'=1\\n'\neq n}}^{N}\|\boldsymbol{a}_{n,n'}\|_2\,,\end{aligned}$$

where $\theta_{\tau,T} \triangleq \sum_{t=\tau}^{T-1}\gamma^{t-\tau}$. From the geometric series summation formula, which establishes that $\theta_{\tau,T} = (1-\gamma^{T-\tau})/(1-\gamma)$, and noting that $\mu = 1-\gamma$, the above equation becomes

$$\tilde{C}_T(\boldsymbol{a}_n) = \frac{1}{T-P}\sum_{\tau=P}^{T-1}(1-\gamma^{T-\tau})\,\ell_\tau^{(n)}(\boldsymbol{a}_n) + \lambda\sum_{\substack{n'=1\\n'\neq n}}^{N}\|\boldsymbol{a}_{n,n'}\|_2\,.$$

From (B.29) and the equation above, the difference $d_T(\boldsymbol{a}_n) \triangleq C_T(\boldsymbol{a}_n) - \tilde{C}_T(\boldsymbol{a}_n)$ between the TISO and TIRSO hindsight objectives is given by:

$$d_T(\boldsymbol{a}_n) = \frac{1}{T-P}\sum_{\tau=P}^{T-1}\gamma^{T-\tau}\,\ell_\tau^{(n)}(\boldsymbol{a}_n). \tag{B.45}$$

To prove part 1, it suffices to show that $d_T(\boldsymbol{a}_n) \to 0$ as $T \to \infty$ for all $\boldsymbol{a}_n$. To this end, expand $\ell_t^{(n)}(\boldsymbol{a}_n)$

$$\ell_t^{(n)}(\boldsymbol{a}_n) = \frac{1}{2}\left(y_n^2[t] + \boldsymbol{a}_n^\top\,\boldsymbol{g}[t]\,\boldsymbol{g}^\top[t]\,\boldsymbol{a}_n - 2\,y_n[t]\,\boldsymbol{g}^\top[t]\,\boldsymbol{a}_n\right), \tag{B.46}$$

and apply Cauchy-Schwarz inequality to obtain

$$\ell_t^{(n)}(\boldsymbol{a}_n) \leq \frac{1}{2}\left[\|\boldsymbol{a}_n\|_2\cdot\|\boldsymbol{g}[t]\|_2\right]^2 + \frac{1}{2}B_y + \sqrt{B_y}\,\|\boldsymbol{g}[t]\|_2\cdot\|\boldsymbol{a}_n\|_2\,. \tag{B.47}$$

On the other hand, the hypothesis $|y_n[t]|^2 \leq B_y \forall n,t$ implies that $\|\boldsymbol{y}[t]\|_2^2 \leq NB_y$, and hence

$$\|\boldsymbol{g}[t]\|_2^2 = \sum_{\tau=t-P}^{t-1}\|\boldsymbol{y}[\tau]\|_2^2 \leq P\max_{t-P\leq\tau\leq t-1}\|\boldsymbol{y}[\tau]\|_2^2 \leq PNB_y.$$

Substituting the upper bound of $\|\boldsymbol{g}[t]\|_2^2$ into (B.47) yields

$$\ell_t^{(n)}(\boldsymbol{a}_n) \leq \frac{1}{2}NPB_y\,\|\boldsymbol{a}_n\|_2^2 + \frac{1}{2}B_y + \sqrt{NP}B_y\,\|\boldsymbol{a}_n\|_2 \triangleq G(\boldsymbol{a}_n) \tag{B.48}$$

Applying the latter bound to (B.45) results in

$$d_T(\boldsymbol{a}_n) \leq \frac{1}{T-P}\sum_{\tau=P}^{T-1}\gamma^{T-\tau}\,G(\boldsymbol{a}_n)$$
$$= \frac{G(\boldsymbol{a}_n)\gamma^T}{T-P}\sum_{\tau=P}^{T-1}\gamma^{-\tau} = \frac{G(\boldsymbol{a}_n)\left(1-\gamma^{T-P}\right)}{(T-P)(\gamma^{-1}-1)}. \tag{B.49}$$

Taking the limit of the right-hand side clearly yields

$$\lim_{T\to\infty}\frac{G(\boldsymbol{a}_n)\left(1-\gamma^{T-P}\right)}{(T-P)(\gamma^{-1}-1)} = 0. \tag{B.50}$$

Noting from (B.45) that $d_T(\boldsymbol{a}_n) \geq 0$, it follows that $\lim_{T\to\infty} d_T(\boldsymbol{a}_n) = 0$, which concludes the proof of part 1.

To prove part 2, note from (B.45) that $d_T(\boldsymbol{a}_n) \geq 0$, which in turn implies that

$$\tilde{C}_T(\boldsymbol{a}_n) \leq C_T(\boldsymbol{a}_n), \tag{B.51}$$

for all $\boldsymbol{a}_n$ and $T > P$. On the other hand, it follows from (B.30) that

$$\tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq \tilde{C}_T(\boldsymbol{a}_n^*[T]). \tag{B.52}$$

Thus, by combining (B.51) and (B.52),

$$\tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq C_T(\boldsymbol{a}_n^*[T]). \tag{B.53}$$

Similarly, from (B.28), it holds that $C_T(\boldsymbol{a}_n^*[T]) \leq C_T(\tilde{\boldsymbol{a}}_n^*[T])$. Subtracting $\tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T])$ from both sides of the latter inequality yields

$$C_T(\boldsymbol{a}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq C_T(\tilde{\boldsymbol{a}}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) = d_T(\tilde{\boldsymbol{a}}_n^*[T]).$$

By combining (B.53) and (B.54), it holds that

$$0 \leq C_T(\boldsymbol{a}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq d_T(\tilde{\boldsymbol{a}}_n^*[T]). \tag{B.54}$$

Since $\lim_{T\to\infty} d_T(\tilde{\boldsymbol{a}}_n^*[T]) = 0$, (B.54) implies that

$$\lim_{T\to\infty} C_T(\boldsymbol{a}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) = 0. \tag{B.55}$$

Finally, to establish part 3, note that it follows from assumption A2, (B.21) and (B.31) that $\tilde{C}_T$ is $\tilde{\beta}$-strongly convex for some $\tilde{\beta} > 0, \forall\, T$. Thus, from (B.30), one finds that

$$\tilde{C}_T(\boldsymbol{a}_n^*[T]) \geq \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) + \frac{\tilde{\beta}}{2}\left\|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2. \tag{B.56}$$

By combining (B.51) and (B.56), it follows that

$$C_T(\boldsymbol{a}_n^*[T]) \geq \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) + \frac{\tilde{\beta}}{2}\left\|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2, \tag{B.57}$$

or, equivalently,

$$C_T(\boldsymbol{a}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \geq \frac{\tilde{\beta}}{2} \|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\|_2^2 \geq 0. \tag{B.58}$$

Taking limits gives rise to

$$\lim_{T\to\infty} \left[C_T(\boldsymbol{a}_n^*[T]) - \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T])\right] \geq \lim_{T\to\infty} \left[\frac{\tilde{\beta}}{2}\|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\|_2^2\right] \geq 0. \tag{B.59}$$

From (B.55) and the sandwich theorem applied to (B.59), we have

$$\lim_{T\to\infty} \left[\frac{\tilde{\beta}}{2}\|\boldsymbol{a}_n^*[T] - \tilde{\boldsymbol{a}}_n^*[T]\|_2^2\right] = 0, \tag{B.60}$$

which concludes the proof.

B.8 Proof of Theorem 2

Consider first the regret of TISO with constant step size.

Lemma 1. *Let $\{\boldsymbol{a}_n[t]\}_{t=P}^T$ be generated by TISO (**Algorithm 9**) with constant step size $\alpha_t = \alpha = \mathcal{O}\big(1/\sqrt{T}\big)$. Under assumptions A1 and A4, we have*

$$R_s^{(n)}[T] = \mathcal{O}\left(PNB_y\, B_{\boldsymbol{a}}^2\, \sqrt{T}\right). \tag{B.61}$$

Proof. See Appendix B.9. □

Observe that the step size in Theorem 1 depends on T and therefore (B.61) cannot be interpreted as directly establishing sublinear regret for TISO. To understand this result, consider a sequence of copies of TISO, each one for a value of T. Each copy has a (potentially) different step size, but uses the same step size for all t. Expression (B.61) bounds the regret of the T-th copy at time T. However, Theorem 1 can be used next to establish sublinear regret for step size sequences that remain constant over windows of exponentially increasing length; see the *doubling trick* [46].

To this end, let the regret in the window $[t_1, t_2]$ be

$$R_s^{(n)}[t_1, t_2] \triangleq \sum_{t=t_1}^{t_2} h_t^{(n)}(\boldsymbol{a}_n[t]) - h_t^{(n)}(\boldsymbol{a}_n^*[t_1, t_2]), \tag{B.62}$$

where $\{\boldsymbol{a}_n[t]\}_t \subset \mathbb{R}^{NP}$ is an arbitrary sequence and

$$\boldsymbol{a}_n^*[t_1, t_2] \triangleq \arg\min_{\boldsymbol{a}_n} \sum_{t=t_1}^{t_2} h_t^{(n)}(\boldsymbol{a}_n). \tag{B.63}$$

The next result establishes a bound on the static regret given the regret at each window.

Lemma 2. *For $T = t_0 2^M$ and for an arbitrary sequence $\{\boldsymbol{a}_n[t]\}_t \subset \mathbb{R}^{NP}$, the regret in* (B.32) *is bounded as:*

$$R_s^{(n)}[T] \leq R_s^{(n)}[P, t_0] + \sum_{m=1}^{M} R_s^{(n)}[t_0 2^{m-1} + 1, t_0 2^m]. \tag{B.64}$$

Proof. For $T = t_0 2^M$, expression (B.32) can be written as:

$$R_s^{(n)}[T] = \sum_{t=P}^{t_0 2^M} h_t^{(n)}(\boldsymbol{a}_n[t]) - \sum_{t=P}^{t_0 2^M} h_t^{(n)}\left(\boldsymbol{a}_n^*[T]\right). \tag{B.65}$$

On the other hand, it follows from (B.62) that (B.64) is equivalent to

$$\begin{aligned} R_s^{(n)}[T] \leq & \sum_{t=P}^{t_0} \left[h_t^{(n)}(\boldsymbol{a}_n[t]) - h_t^{(n)}\left(\boldsymbol{a}_n^*[P, t_0]\right) \right] \\ & + \sum_{m=1}^{M} \sum_{t=t_0 2^{m-1}+1}^{t_0 2^m} \left[h_t^{(n)}(\boldsymbol{a}_n[t]) - h_t^{(n)}\left(\boldsymbol{a}_n^*\left[t_0 2^{m-1}+1, t_0 2^m\right]\right) \right], \end{aligned} \tag{B.66}$$

The inequality in (B.66) can also be rewritten as

$$R_s^{(n)}[T] \leq \sum_{t=P}^{t_0 2^M} h_t^{(n)}(\boldsymbol{a}_n[t]) - \left[\sum_{t=P}^{t_0} h_t^{(n)}\left(\boldsymbol{a}_n^*[P, t_0]\right) + \sum_{m=1}^{M} \sum_{t=t_0 2^{m-1}+1}^{t_0 2^m} h_t^{(n)}\left(\boldsymbol{a}_n^*\left[t_0 2^{m-1}+1, t_0 2^m\right]\right) \right]. \tag{B.67}$$

By comparing (B.65) and (B.67), proving (B.64) is equivalent to showing that

$$\sum_{t=P}^{t_0 2^M} h_t^{(n)}\left(\boldsymbol{a}_n^*[T]\right) \geq \left[\sum_{t=P}^{t_0} h_t^{(n)}\left(\boldsymbol{a}_n^*[P, t_0]\right) + \sum_{m=1}^{M} \sum_{t=t_0 2^{m-1}+1}^{t_0 2^m} h_t^{(n)}\left(\boldsymbol{a}_n^*\left[t_0 2^{m-1}+1, t_0 2^m\right]\right) \right]. \tag{B.68}$$

From the definitions of $\boldsymbol{a}_n^*[T]$ in (B.28) and $\boldsymbol{a}_n^*[t_1, t_2]$ in (B.63), the above inequality holds since $\inf_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) \leq \inf_{\boldsymbol{x}=\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$. □

The next step is to bound the regret at each window using Lemma 1. To this end, one must set $\alpha_{[m]}$ as a function $\mathcal{O}(1/\sqrt{T_m})$, where $T_m \triangleq t_0 2^m - t_0 2^{m-1} = t_0 2^{m-1}$ is the length of the $(m+1)$-th window, $m = 1, \ldots, M$. Invoking Lemma 1, the regret for the $(m+1)$-th window is given by $R_s^{(n)}[t_0 2^{m-1}+1, t_0 2^m] = \mathcal{O}(PNB_y B_{\boldsymbol{a}}^2 \sqrt{2^{m-1}})$. By Lemma 2, the regret of TISO becomes

$$\begin{aligned} R_s^{(n)}[T] &= \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \sqrt{t_0 - P + 1}\right) \\ &\quad + \sum_{m=1}^{M} \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \sqrt{t_0 2^{m-1}}\right) \\ &= \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \sum_{m=1}^{M} \sqrt{t_0 2^{m-1}}\right) \\ &= \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 (\sqrt{2})^M\right) \\ &= \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \left(2^{\log_2 \frac{T}{t_0}}\right)^{\frac{1}{2}}\right) \\ &= \mathcal{O}\left(PNB_y B_{\boldsymbol{a}}^2 \sqrt{T}\right), \end{aligned}$$

which concludes the proof.

B.9 Proof of Lemma 1

First we present a lemma that establishes that the hindsight solution of TISO is bounded and then we will present the proof of Lemma 1.

Lemma 3. *Under assumptions A1, A2, and A4, the hindsight solution of TISO* $\boldsymbol{a}_n^*[T]$ *given in* (B.28) *is bounded as*

$$\|\boldsymbol{a}_n^*[T]\|_2 \leq B_{\boldsymbol{a}} \triangleq \frac{1}{\beta}\left(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta B_y}\right). \tag{B.69}$$

Proof. Note that $a_n^*[T]$ belongs to the sublevel set of TISO hindsight objective for $\boldsymbol{a}_n = \mathbf{0}_{NP}$, given by

$$\mathcal{S}_T \triangleq \{\boldsymbol{a}_n : C_T(\boldsymbol{a}_n) \leq C_T(\mathbf{0}_{NP})\}, \tag{B.70}$$

where $C_T(\mathbf{0}_{NP})$ is upper bounded by

$$\begin{aligned} C_T(\mathbf{0}_{NP}) &= \frac{1}{T-P}\sum_{t=P}^{T-1}\frac{1}{2}y_n^2[t] \\ &\leq \frac{1}{2(T-P)}\sum_{t=P}^{T-1} B_y = \frac{B_y}{2}. \end{aligned}$$

This means that we can write:

$$\mathcal{S}_T \subset \left\{\boldsymbol{a}_n^*[T] : C_T(\boldsymbol{a}_n^*[T]) \leq \frac{B_y}{2}\right\}. \tag{B.71}$$

Next, we find a lower bound to $C_T(\boldsymbol{a}_n^*[T])$ that is an increasing function of $\|\boldsymbol{a}_n^*[T]\|_2$ as follows

$$\begin{aligned} C_T(\boldsymbol{a}_n^*[T]) &= \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}(\boldsymbol{a}_n^*[T])^\top \boldsymbol{g}[t]\boldsymbol{g}^\top[t]\boldsymbol{a}_n^*[T] - y_n[t]\boldsymbol{g}^\top[t]\boldsymbol{a}_n^*[T] + \frac{1}{2}y_n^2[t] + \Omega^{(n)}(\boldsymbol{a}_n^*[T])\right] \\ &\geq \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}(\boldsymbol{a}_n^*[T])^\top \boldsymbol{g}[t]\boldsymbol{g}^\top[t]\boldsymbol{a}_n^*[T] - y_n[t]\boldsymbol{g}^\top[t]\boldsymbol{a}_n^*[T]\right] \\ &\geq \frac{1}{2}\lambda_{\min}\left(\frac{1}{T-P}\sum_{t=P}^{T-1}\boldsymbol{g}[t]\boldsymbol{g}^\top[t]\right)\|\boldsymbol{a}_n^*[T]\|_2^2 - \frac{1}{T-P}\sum_{t=P}^{T-1} y_n[t]\,\|\boldsymbol{g}[t]\|_2 \cdot \|\boldsymbol{a}_n^*[T]\|_2 \\ &\geq \frac{1}{2}\beta\,\|\boldsymbol{a}_n^*[T]\|_2^2 - B_y\sqrt{PN}\,\|\boldsymbol{a}_n^*[T]\|_2. \end{aligned}$$

Therefore,

$$\mathcal{S}_T \subset \left\{\boldsymbol{a}_n^*[T] : \frac{1}{2}\beta\|\boldsymbol{a}_n^*[T]\|_2^2 - B_y\sqrt{PN}\|\boldsymbol{a}_n^*[T]\|_2 \leq \frac{B_y}{2}\right\}. \tag{B.73}$$

Further, we can write

$$\mathcal{S}_T \subset \{\boldsymbol{a}_n^*[T] : \|\boldsymbol{a}_n^*[T]\|_2 \leq B_{\boldsymbol{a}}\}, \tag{B.74}$$

with $B_{\boldsymbol{a}} \triangleq 1/\beta(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta B_y})$. Expression (B.74) implies that the TISO hindsight solution is bounded. □

Now, we present the proof of Lemma 1. This proof is based on the idea that if the inequality $\|\nabla \ell_t^{(n)}(\boldsymbol{a}_n)\|_2^2 \leq 2PNB_y\, \ell_t^{(n)}(\boldsymbol{a}_n), \forall\, t, n$ holds and the strong convexity parameter of ψ is 1, then it follows from [67, Corollary 5] that:

$$\begin{aligned} R_s^{(n)}[T] &= \mathcal{O}\Big(\tfrac{1}{2}\,\rho\sqrt{T-P}\;\|\boldsymbol{a}_n^*[T] - \boldsymbol{a}_n[P]\|_2^2\Big) \\ &= \mathcal{O}\Big(\tfrac{1}{2}\,\rho\sqrt{T}\;\|\boldsymbol{a}_n^*[T]\|_2^2\Big) \\ &= \mathcal{O}\Big(PNB_y\sqrt{T}\,\|\boldsymbol{a}_n^*[T]\|_2^2\Big) \\ &= \mathcal{O}\Big(PNB_y\sqrt{T}B_{\boldsymbol{a}}^2\Big), \end{aligned}$$

where $B_{\boldsymbol{a}}$ is defined in (B.69). We still need to show that the inequality $\|\nabla \ell_t^{(n)}(\boldsymbol{a}_n)\|_2^2 \leq 2PNB_y \ell_t^{(n)}(\boldsymbol{a}_n)$, $\forall\, t, n$, holds. To this end, note from (B.15) that:

$$\begin{aligned} \big\|\nabla \ell_t^{(n)}(\boldsymbol{a}_n)\big\|_2^2 &= \big\|\boldsymbol{g}[t]\,\big(\boldsymbol{g}^\top[t]\boldsymbol{a}_n - y_n[t]\big)\big\|_2^2 \\ &= \|\boldsymbol{g}[t]\|_2^2 \cdot \big|y_n[t] - \boldsymbol{g}^\top[t]\,\boldsymbol{a}_n\big|^2. \end{aligned} \tag{B.75}$$

On the other hand, the hypothesis $|y_n[t]|^2 \leq B_y\ \forall\ n, t$ implies that $\|\boldsymbol{y}[t]\|_2^2 \leq NB_y$ and, therefore:

$$\|\boldsymbol{g}[t]\|_2^2 = \sum_{\tau=t-P}^{t-1} \|\boldsymbol{y}[\tau]\|_2^2 \leq P \max_{t-P\leq\tau\leq t-1} \|\boldsymbol{y}[\tau]\|_2^2 \leq PNB_y. \tag{B.76}$$

Combining (B.75) and (B.76) yields

$$\big\|\nabla \ell_t^{(n)}(\boldsymbol{a}_n)\big\|_2^2 \leq PNB_y\,\big|y_n[t] - \boldsymbol{g}^\top[t]\,\boldsymbol{a}_n\big|^2. \tag{B.77}$$

Thus, to satisfy

$$\big\|\nabla \ell_t^{(n)}(\boldsymbol{a}_n)\big\|_2^2 \leq \rho\,\ell_t^{(n)}(\boldsymbol{a}_n) = \rho\,\frac{1}{2}\,\big(y_n[t] - \boldsymbol{g}^\top[t]\,\boldsymbol{a}_n\big)^2,$$

it suffices to set $\rho = 2PNB_y$.

B.10 Proof of Theorem 3

The first step is to obtain a bound for constant step size.

Lemma 4. *Let $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^T$ be generated by TIRSO (**Algorithm 10**) with constant step size $\alpha_t = \alpha = \mathcal{O}\big(1/\sqrt{T}\big)$. Under assumptions A1, A2, and A3, we have*

$$\tilde{R}_s^{(n)}[T] = \mathcal{O}\Big(L\,B_{\tilde{\boldsymbol{a}}}^2\,\sqrt{T}\Big). \tag{B.78}$$

Proof. See Appendix B.11. □

The rest of the proof proceeds along the lines of the proof of Theorem 2.

B.11 Proof of Lemma 4

First, we present a lemma that establishes that the hindsight solution of TIRSO is bounded. Then, we will present the proof of Lemma 4.

Lemma 5. *Under the assumptions A1 and A2, the hindsight solution of TIRSO* $\tilde{\boldsymbol{a}}_n^*[T]$ *given in* (B.30) *is bounded as*

$$\|\tilde{\boldsymbol{a}}_n^*[T]\|_2 \leq B_{\tilde{\boldsymbol{a}}} \triangleq \frac{1}{\beta_{\tilde{\ell}}}\left(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta_{\tilde{\ell}} B_y}\right). \tag{B.79}$$

Proof. The proof follows similar steps to those of Lemma 3. Consider the sublevel set of TIRSO hindsight objective for $\tilde{\boldsymbol{a}}_n^*[T] = \mathbf{0}_{NP}$,

$$\tilde{\mathcal{S}}_T \triangleq \left\{\tilde{\boldsymbol{a}}_n^*[T] : \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq \tilde{C}_T(\mathbf{0}_{NP})\right\}, \tag{B.80}$$

where $\tilde{C}_T(\mathbf{0}_{NP})$ is upper bounded as follows:

$$\begin{aligned}
\tilde{C}_T(\mathbf{0}_{NP}) &= \frac{1}{T-P}\sum_{t=P}^{T-1}\frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[t] \\
&\leq \frac{B_y\mu}{2(T-P)}\sum_{t=P}^{T-1}\sum_{\tau=P}^{t}\gamma^{t-\tau} \\
&= \frac{B_y\mu}{2(T-P)}\sum_{t=P}^{T-1}\frac{1-\gamma^{t-P+1}}{1-\gamma} \\
&\leq \frac{B_y}{2(T-P)}\sum_{t=P}^{T-1}1 = \frac{B_y}{2}.
\end{aligned}$$

This implies that

$$\tilde{\mathcal{S}}_T \subset \left\{\tilde{\boldsymbol{a}}_n^*[T] : \tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) \leq \frac{B_y}{2}\right\}. \tag{B.82}$$

Next, we find a lower bound to $\tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T])$ that is an increasing function of $\|\tilde{\boldsymbol{a}}_n^*[T]\|_2$ as follows

$$\begin{aligned}
\tilde{C}_T(\tilde{\boldsymbol{a}}_n^*[T]) &= \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}(\tilde{\boldsymbol{a}}_n^*[T])^\top\boldsymbol{\Phi}[t]\tilde{\boldsymbol{a}}_n^*[T] - \boldsymbol{r}_n^\top[t]\tilde{\boldsymbol{a}}_n^*[T] + \frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[t] + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\right] \\
&\geq \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}(\tilde{\boldsymbol{a}}_n^*[T])^\top\boldsymbol{\Phi}[t]\tilde{\boldsymbol{a}}_n^*[T] - \boldsymbol{r}_n^\top[t]\tilde{\boldsymbol{a}}_n^*[T] + \frac{\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[t]\right] \\
&\geq \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}\lambda_{\min}(\boldsymbol{\Phi}[t])\|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 - \|\boldsymbol{r}_n[t]\|_2\,\|\tilde{\boldsymbol{a}}_n^*[T]\|_2\right] \\
&\geq \frac{1}{T-P}\sum_{t=P}^{T-1}\left[\frac{1}{2}\beta_{\tilde{\ell}}\|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 - B_y\sqrt{PN}\|\tilde{\boldsymbol{a}}_n^*[T]\|_2\right] \\
&= \frac{1}{2}\beta_{\tilde{\ell}}\|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 - B_y\sqrt{PN}\|\tilde{\boldsymbol{a}}_n^*[T]\|_2.
\end{aligned}$$

Therefore,

$$\tilde{\mathcal{S}}_T \subset \left\{ \tilde{\boldsymbol{a}}_n^*[T] : \frac{1}{2}\beta_{\tilde{\ell}} \|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 - B_y\sqrt{PN}\|\tilde{\boldsymbol{a}}_n^*[T]\|_2 \leq \frac{B_y}{2} \right\}. \tag{B.84}$$

Further, we can write

$$\tilde{\mathcal{S}}_T \subset \left\{ \tilde{\boldsymbol{a}}_n^*[T] : \|\tilde{\boldsymbol{a}}_n^*[T]\|_2 \leq \frac{1}{\beta_{\tilde{\ell}}} \left(B_y\sqrt{PN} + \sqrt{B_y^2 PN + \beta_{\tilde{\ell}} B_y} \right) \right\}. \tag{B.85}$$

Expression (B.85) implies that the TIRSO hindsight solution is bounded. □

Now, we present the proof of Lemma 4. The proof has two parts. The first step is to prove that there exists $\tilde{\rho} > 0$ such that

$$\left\| \nabla \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) \right\|_2^2 \leq \tilde{\rho}\, \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n), \quad \forall\, t,\, n, \tag{B.86}$$

holds for all $\boldsymbol{a}_n$. The second step is to apply the result of [67, Corollary 5] in the present case. To prove the first part, from (B.21) and $\nabla \tilde{\ell}_t^{(n)}(\boldsymbol{a}_n) = \boldsymbol{\Phi}[t]\boldsymbol{a}_n - \boldsymbol{r}_n[t]$, it follows that (B.86) is equivalent to

$$\|\boldsymbol{\Phi}[t]\boldsymbol{a}_n - \boldsymbol{r}_n[t]\|_2^2 \leq \tilde{\rho}\Big(\frac{1}{2}\boldsymbol{a}_n^\top \boldsymbol{\Phi}[t]\boldsymbol{a}_n - \boldsymbol{r}_n^\top[t]\boldsymbol{a}_n + \frac{1}{2}\sum_{\tau=P}^{t} \mu\gamma^{t-\tau} y_n^2[t] \Big), \ \forall\, t,\, n. \tag{B.87}$$

By expanding the left-hand side of (B.87), rearranging terms, and introducing $Z_t(\boldsymbol{a}_n)$ as

$$\begin{aligned} Z_t(\boldsymbol{a}_n) \triangleq\ & \boldsymbol{a}_n^\top \Big(\frac{\tilde{\rho}}{2}\boldsymbol{\Phi}[t] - \boldsymbol{\Phi}^\top[t]\boldsymbol{\Phi}[t] \Big)\boldsymbol{a}_n + (2\boldsymbol{r}_n^\top[t]\boldsymbol{\Phi}[t] - \tilde{\rho}\, \boldsymbol{r}_n^\top[t])\boldsymbol{a}_n \\ & + \frac{\tilde{\rho}\mu}{2}\sum_{\tau=P}^{t} \gamma^{t-\tau} y_n^2[t] - \boldsymbol{r}_n^\top[t]\boldsymbol{r}_n[t], \end{aligned} \tag{B.88}$$

the condition in (B.86) is equivalent to $Z_t(\boldsymbol{a}_n) \geq 0$. So the goal becomes finding $\tilde{\rho}$ such that $Z_t(\boldsymbol{a}_n) \geq 0$ for all $\boldsymbol{a}_n$ and t. For this condition to hold, it is necessary that (a) $\inf_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n)$ is finite for all t, and (b) $\inf_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n) \geq 0$ for all t. It can be seen [115, Appendix A.5] that condition (a) holds iff (a1) the Hessian matrix $\boldsymbol{H}Z_t(\boldsymbol{a}_n) = \tilde{\rho}\boldsymbol{\Phi}[t] - 2\boldsymbol{\Phi}^\top[t]\boldsymbol{\Phi}[t]$ is positive semidefinite, and (a2) $2\boldsymbol{\Phi}[t]\boldsymbol{r}_n[t] - \tilde{\rho}\boldsymbol{r}_n[t] \in \mathcal{R}(\boldsymbol{H}Z_t(\boldsymbol{a}_n))$, where $\mathcal{R}(\boldsymbol{A})$ denotes the span of the columns of a matrix $\boldsymbol{A}$. The first step is to find $\tilde{\rho}$ such that (a1) holds. To this end, consider the eigenvalue decomposition of $\boldsymbol{\Phi}[t] = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where the index t is omitted to simplify notation. Therefore,

$$\boldsymbol{H}Z_t(\boldsymbol{a}_n) = \boldsymbol{U}\left(\tilde{\rho}\boldsymbol{\Lambda} - 2\boldsymbol{\Lambda}^2 \right)\boldsymbol{U}^\top. \tag{B.89}$$

Let $\lambda_{\max}(\boldsymbol{\Phi}[t])$ denote the maximum eigenvalue of $\boldsymbol{\Phi}[t]$. It follows from (B.89) that $\boldsymbol{H}Z_t(\boldsymbol{a}_n)$ is positive semidefinite if

$$\tilde{\rho} \geq 2\lambda_{\max}(\boldsymbol{\Phi}[t]). \tag{B.90}$$

It remains to be shown that there exists $\tilde{\rho} > 0$ such that (B.90), (a2), and (b) simultaneously hold. To this end, focus first on (a2), which can be rewritten as

$$\begin{aligned} 2\boldsymbol{\Phi}[t]\boldsymbol{r}_n[t] - \tilde{\rho}\boldsymbol{r}_n[t] &\in \mathcal{R}\left(\tilde{\rho}\boldsymbol{\Phi}[t] - 2\boldsymbol{\Phi}^\top[t]\boldsymbol{\Phi}[t] \right) \\ &= \mathcal{R}\left(\boldsymbol{\Phi}[t]\left(\tilde{\rho}\boldsymbol{I} - 2\boldsymbol{\Phi}[t] \right) \right). \end{aligned} \tag{B.91}$$

Clearly, if $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$, then $\tilde{\rho}\boldsymbol{I} - 2\boldsymbol{\Phi}[t]$ is invertible and, hence, $\mathcal{R}(\boldsymbol{\Phi}[t](\tilde{\rho}\boldsymbol{I} - 2\boldsymbol{\Phi}[t])) = \mathcal{R}(\boldsymbol{\Phi}[t])$ [116, Ch. 4]. Thus, (B.91) holds if $2\boldsymbol{\Phi}[t]\boldsymbol{r}_n[t] \in \mathcal{R}(\boldsymbol{\Phi}[t])$ and $\tilde{\rho}\boldsymbol{r}_n[t] \in \mathcal{R}(\boldsymbol{\Phi}[t])$. The former condition is trivial. To verify the latter, define

$$\boldsymbol{y}_n \triangleq [y_n[P], \ldots, y_n[t]]^\top \in \mathbb{R}^{t-P+1\times 1}, \tag{B.92a}$$

$$\boldsymbol{G} \triangleq [\boldsymbol{g}[P], \ldots, \boldsymbol{g}[t]] \in \mathbb{R}^{NP\times t-P+1}, \tag{B.92b}$$

$$\boldsymbol{\Gamma} \triangleq \operatorname{diag}\left(\mu[\gamma^{t-P}, \ldots, \gamma^0]\right) \in \mathbb{R}^{t-P+1\times t-P+1}, \tag{B.92c}$$

and $\boldsymbol{B} \triangleq \boldsymbol{G}\boldsymbol{\Gamma}^{1/2}$; note that $\boldsymbol{\Phi}[t] = \boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top = \boldsymbol{B}\boldsymbol{B}^\top$. It follows that $\boldsymbol{r}_n[t] = \boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{y}_n = \boldsymbol{B}\boldsymbol{\Gamma}^{1/2}\boldsymbol{y}_n \in \mathcal{R}(\boldsymbol{B}) = \mathcal{R}(\boldsymbol{B}\boldsymbol{B}^\top) = \mathcal{R}(\boldsymbol{\Phi}[t])$. Therefore, $\tilde{\rho}\boldsymbol{r}_n[t] \in \mathcal{R}(\boldsymbol{\Phi}[t])$ holds and, consequently, (a2) holds whenever $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$.

So far, this proof has established that, if $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$, then both (a1) and (a2) hold. The next step is to show that (b) also holds when $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$. To this end, set the gradient of $Z_t(\boldsymbol{a}_n)$ equal to zero and use $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$ to obtain $\boldsymbol{\Phi}^\dagger[t]\boldsymbol{r}_n[t] \in \arg\min_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n)$, where the symbol $\dagger$ denotes pseudo-inverse. From this expression and (B.88), it follows that

$$\begin{aligned}\inf_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n) &= Z_t(\boldsymbol{\Phi}^\dagger[t]\boldsymbol{r}_n[t])\\ &= \boldsymbol{r}_n^\top[t]\boldsymbol{\Phi}^\dagger[t]\Big(\frac{\tilde{\rho}}{2}\boldsymbol{\Phi}[t] - \boldsymbol{\Phi}^\top[t]\boldsymbol{\Phi}[t]\Big)\boldsymbol{\Phi}^\dagger[t]\boldsymbol{r}_n[t] + \big(2\boldsymbol{r}_n^\top[t]\boldsymbol{\Phi}[t]\\ &\quad - \tilde{\rho}\boldsymbol{r}_n^\top[t]\big)\boldsymbol{\Phi}^\dagger[t]\boldsymbol{r}_n[t] + \frac{\tilde{\rho}\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[t] - \boldsymbol{r}_n^\top[t]\boldsymbol{r}_n[t].\end{aligned}$$

Applying the properties of the pseudoinverse and simplifying results in

$$\inf_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n) = \frac{\tilde{\rho}\mu}{2}\sum_{\tau=P}^{t}\gamma^{t-\tau}y_n^2[t] - \frac{\tilde{\rho}}{2}\boldsymbol{r}_n^\top[t]\boldsymbol{\Phi}^\dagger[t]\boldsymbol{r}_n[t]. \tag{B.93}$$

From this expression, note that the condition $\inf_{\boldsymbol{a}_n} Z_t(\boldsymbol{a}_n) \geq 0$ is equivalent to

$$\boldsymbol{y}_n^\top\boldsymbol{\Gamma}\boldsymbol{y}_n \geq \boldsymbol{y}_n^\top\boldsymbol{\Gamma}\boldsymbol{G}^\top\left(\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top\right)^\dagger\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{y}_n, \tag{B.94}$$

and, upon defining $\tilde{\boldsymbol{y}}_n \triangleq \boldsymbol{\Gamma}^{1/2}\boldsymbol{y}_n$,

$$\tilde{\boldsymbol{y}}_n^\top\tilde{\boldsymbol{y}}_n \geq \tilde{\boldsymbol{y}}_n^\top\boldsymbol{\Gamma}^{1/2}\boldsymbol{G}^\top\left(\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top\right)^\dagger\boldsymbol{G}\boldsymbol{\Gamma}^{1/2}\tilde{\boldsymbol{y}}_n^\top. \tag{B.95}$$

This inequality trivially holds when $\tilde{\boldsymbol{y}}_n = \boldsymbol{0}_{t-P+1}$. Thus, assume without loss of generality that $\tilde{\boldsymbol{y}}_n \neq \boldsymbol{0}_{t-P+1}$. By setting $\boldsymbol{A} \triangleq \boldsymbol{\Gamma}^{1/2}\boldsymbol{G}^\top(\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top)^\dagger$, one obtains $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{\Gamma}^{1/2}\boldsymbol{G}^\top(\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top)^\dagger\boldsymbol{G}\boldsymbol{\Gamma}^{1/2}$ and $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{\Phi}[t]\boldsymbol{\Phi}^\dagger[t]$.

Since the nonzero eigenvalues of $\boldsymbol{A}\boldsymbol{B}$ and $\boldsymbol{B}\boldsymbol{A}$ are the same [117, Sec. 3.2.11] and the maximum eigenvalue of $\boldsymbol{B}\boldsymbol{A}$ is 1, then the maximum eigenvalue of $\boldsymbol{A}\boldsymbol{B}$ is also 1. Therefore

$$\frac{\tilde{\boldsymbol{y}}_n^\top\boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}_n}{\tilde{\boldsymbol{y}}_n^\top\tilde{\boldsymbol{y}}_n} = \frac{\tilde{\boldsymbol{y}}_n^\top\boldsymbol{\Gamma}^{1/2}\boldsymbol{G}^\top\left(\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{G}^\top\right)^\dagger\boldsymbol{G}\boldsymbol{\Gamma}^{1/2}\tilde{\boldsymbol{y}}_n}{\tilde{\boldsymbol{y}}_n^\top\tilde{\boldsymbol{y}}_n} \leq 1, \tag{B.96}$$

and, hence, (B.95) holds. To sum up, conditions (a) and (b) hold if $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$. In other words, (B.86) holds for any choice of $\tilde{\rho}$ such that $\tilde{\rho} > 2\lambda_{\max}(\boldsymbol{\Phi}[t])$ for all t. This

completes the first part of the proof. The second part of the proof consists of setting $\tilde{\rho} = \sup_t \lambda_{\max}(\mathbf{\Phi}[t]) + \epsilon$ with $\epsilon > 0$ an arbitrary constant, and invoking [67, Corollary 5] to conclude that

$$\tilde{R}_s^{(n)}[T] = \mathcal{O}\left(\tilde{\rho} \, \|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 \, \sqrt{T}\right).$$

Using assumption A3 and substituting the upper bound on $\|\tilde{\boldsymbol{a}}_n^*[T]\|_2$ from (B.79) into the above expression completes the proof.

B.12 Proof of Theorem 4

To prove Theorem 4, first we present two lemmas. Before presenting the result related to logarithmic regret of TIRSO, it is worth mentioning that a related result is presented in [67, Th. 7], which is applicable to strongly convex regularization functions. Note that in TIRSO, the data-fitting function is strongly convex. It can be easily shown that COMID applied to a problem with strongly convex regularizer produces different iterates than COMID applied to a strongly convex data-fitting function.

Lemma 6. *Under assumption A2, let the sequence* $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^T$ *be generated by TIRSO (**Algorithm 10**) with a step size* α_t*, and let* $\tilde{\boldsymbol{a}}_n^*[T]$ *be the hindsight solution for TIRSO at time* T *defined in* (B.30)*. Then*

$$\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) \leq \frac{1}{2\alpha_t}(1 - \alpha_t\beta_{\tilde{\ell}}) \, \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t]\|_2^2 - \frac{1}{2\alpha_t} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2 + \frac{\alpha_t}{2} \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2, \quad \text{(B.97)}$$

for $P \leq t \leq T$, $\forall \, \boldsymbol{g}_t^{\tilde{\ell}} \in \partial(\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]))$.

Proof. For a strongly convex $\tilde{\ell}_t^{(n)}$, by the subgradient inequality, we have

$$\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) \geq \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + (\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t])^\top \boldsymbol{g}_t^{\tilde{\ell}} + \frac{\beta_{\tilde{\ell}}}{2} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t]\|_2^2, \quad \text{(B.98)}$$

$\forall \boldsymbol{g}_t^{\tilde{\ell}} \in \partial(\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]))$. On the other hand, since $\Omega^{(n)}$ is convex,

$$\Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) \geq \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) + (\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1])^\top \boldsymbol{g}_{t+1}^{\Omega}, \quad \text{(B.99)}$$

$\forall \, \boldsymbol{g}_{t+1}^{\Omega} \in \partial(\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]))$. Adding (B.98) and (B.99), scaling by α_t, and rearranging

terms,

$$
\begin{aligned}
&\alpha_t \left(\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\right) \\
&\leq \alpha_t \Big((\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T])^\top \boldsymbol{g}_t^{\tilde{\ell}} + (\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n^*[T])^\top \boldsymbol{g}_{t+1}^{\Omega} - \frac{\beta_{\tilde{\ell}}}{2} \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 \Big) \\
&\overset{(a)}{=} \left(\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right)^\top \left(\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1] - \alpha_t \boldsymbol{g}_t^{\tilde{\ell}} - \alpha_t \boldsymbol{g}_{t+1}^{\Omega}\right) \\
&\quad + \alpha_t \left(\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1]\right)^\top \boldsymbol{g}_t^{\tilde{\ell}} - \frac{\alpha_t \beta_{\tilde{\ell}}}{2} \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 + \left(\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right)^\top \left(\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]\right) \\
&\overset{(b)}{\leq} \alpha_t \left(\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1]\right)^\top \boldsymbol{g}_t^{\tilde{\ell}} - \frac{\alpha_t \beta_{\tilde{\ell}}}{2} \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 + \left(\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right)^\top \left(\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]\right) \\
&\overset{(c)}{=} \alpha_t \left\langle \frac{1}{\sqrt{\alpha_t}} (\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1]), \sqrt{\alpha_t} \boldsymbol{g}_t^{\tilde{\ell}} \right\rangle - \frac{\alpha_t \beta_{\tilde{\ell}}}{2} \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 + \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t]\right\|_2^2 \\
&\quad - \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2 - \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]\right\|_2^2 \\
&\overset{(d)}{\leq} \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2 + \frac{\alpha_t^2}{2} \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 - \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]\right\|_2^2 - \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2 \\
&\quad + \left(\frac{1}{2} - \frac{\alpha_t \beta_{\tilde{\ell}}}{2}\right) \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 \\
&= \frac{\alpha_t^2}{2} \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 + \left(\frac{1}{2} - \frac{\alpha_t \beta_{\tilde{\ell}}}{2}\right) \left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 - \frac{1}{2} \left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2, \qquad \text{(B.100)}
\end{aligned}
$$

where (a) results from adding and subtracting the term $\tilde{\boldsymbol{a}}_n^\top[t+1]\boldsymbol{g}_t^{\tilde{\ell}} + (\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1])^\top(\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1])$ followed by rearranging terms; in (b) the inequality $(\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1])^\top(\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n[t+1] - \alpha_t \boldsymbol{g}_t^{\tilde{\ell}} - \alpha_t \boldsymbol{g}_{t+1}^{\Omega}) \leq 0$ is used, which is implied by the optimality of $\tilde{\boldsymbol{a}}_n[t+1]$ in (B.24), i.e., $(\boldsymbol{a}_n - \tilde{\boldsymbol{a}}_n[t+1])^\top(\tilde{\nabla} \tilde{J}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t+1])) \geq 0, \forall\, \boldsymbol{a}_n$; in (c) the Pythagorean theorem for Euclidean distance (i.e. $(\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1])^\top(\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]) = 1/2\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t]\|_2^2 - 1/2\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2 - 1/2\|\tilde{\boldsymbol{a}}_n[t+1] - \tilde{\boldsymbol{a}}_n[t]\|_2^2$) is used; in (d) the inequality $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 1/2(\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{y}\|_2^2)$ is used. Dividing both sides of (B.100) by α_t completes the proof. □

Next, we establish that TIRSO estimates $\tilde{\boldsymbol{a}}_n[t]$ are bounded and a bound on $\|\nabla \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2$ that depends on parameters of the algorithm, is derived.

Lemma 7. *Under assumptions A1 and A2, and let the sequence of iterates $\{\tilde{\boldsymbol{a}}_n[t]\}$ be generated by TIRSO (**Algorithm 10**). Then*

$$
\left\|\tilde{\boldsymbol{a}}_n[t+1]\right\|_2 \leq (1 - \alpha_t \beta_{\tilde{\ell}}) \left\|\tilde{\boldsymbol{a}}_n[t]\right\|_2 + \alpha_t \sqrt{PN} B_y. \qquad \text{(B.101)}
$$

Proof. From the update expression of TIRSO, we have

$$
\begin{aligned}
\left\|\tilde{\boldsymbol{a}}_n[t+1]\right\|_2 &\leq \left\|\tilde{\boldsymbol{a}}_n^{\mathrm{f}}[t+1]\right\|_2 \\
&= \left\|\tilde{\boldsymbol{a}}_n[t] - \alpha_t \boldsymbol{v}_n[t]\right\|_2 \\
&= \left\|\tilde{\boldsymbol{a}}_n[t] - \alpha_t \left(\boldsymbol{\Phi}[t] \tilde{\boldsymbol{a}}_n[t] - \boldsymbol{r}_n[t]\right)\right\|_2 \\
&= \left\|\left(\boldsymbol{I} - \alpha_t \boldsymbol{\Phi}[t]\right) \tilde{\boldsymbol{a}}_n[t] + \alpha_t \boldsymbol{r}_n[t]\right\|_2 \\
&\leq \lambda_{\max}\left(\boldsymbol{I} - \alpha_t \boldsymbol{\Phi}[t]\right) \left\|\tilde{\boldsymbol{a}}_n[t]\right\|_2 + \alpha_t \left\|\boldsymbol{r}_n[t]\right\|_2 \\
&= \left(1 - \alpha_t \lambda_{\min}(\boldsymbol{\Phi}[t]\right) \left\|\tilde{\boldsymbol{a}}_n[t]\right\|_2 + \alpha_t \left\|\boldsymbol{r}_n[t]\right\|_2 \\
&\leq (1 - \alpha_t \beta_{\tilde{\ell}}) \left\|\tilde{\boldsymbol{a}}_n[t]\right\|_2 + \alpha_t \left\|\boldsymbol{r}_n[t]\right\|_2. \qquad \text{(B.102)}
\end{aligned}
$$

Now, we derive an upper bound on $\|\boldsymbol{r}_n[t]\|_2$. By the definition of $\boldsymbol{r}_n[t]$ in (B.22b) and assumption A1, we have

$$\begin{aligned}\|\boldsymbol{r}_n[t]\|_2 &= \left\|\mu \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau]\, \boldsymbol{g}[\tau]\right\|_2 \\ &\leq \mu \left\|\sum_{\tau=P}^{t} \gamma^{t-\tau} \sqrt{B_y}\sqrt{B_y}\mathbf{1}_{NP}\right\|_2 \qquad \text{(B.103a)} \\ &= \mu B_y \sqrt{PN}\gamma^t \sum_{\tau=P}^{t} \left(\frac{1}{\gamma}\right)^{\tau} \\ &= B_y\sqrt{PN}(1-\gamma^{t-P+1}) \\ &\leq \sqrt{PN}B_y. \qquad \text{(B.103b)}\end{aligned}$$

Substituting the upper bound of $\boldsymbol{r}_n[t]$ from (E.49b) into (C.24) completes the proof. □

Lemma 8. *Under assumptions A1, A2, and A3, and let the sequence of iterates* $\{\tilde{\boldsymbol{a}}_n[t]\}$ *be generated by TIRSO (**Algorithm 10**) with* $\alpha_t = 1/(\beta_{\tilde{\ell}} t)$. *Then*

$$\|\tilde{\boldsymbol{a}}_n[t]\|_2 \leq 1/\beta_{\tilde{\ell}}\sqrt{PN}B_y, \forall\, t \geq P, \qquad \text{(B.104)}$$

$$\left\|\nabla \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \leq G_{\tilde{\ell}} \triangleq \left(1+\frac{L}{\beta_{\tilde{\ell}}}\right)\sqrt{PN}B_y, \forall\, t \geq P. \qquad \text{(B.105)}$$

Proof. Invoking Lemma 7 and setting $\alpha_t = 1/(\beta_{\tilde{\ell}} t)$ in (B.101),

$$\begin{aligned}\|\tilde{\boldsymbol{a}}_n[t+1]\|_2 &= \left(1-\frac{1}{\beta_{\tilde{\ell}} t}\beta_{\tilde{\ell}}\right)\|\tilde{\boldsymbol{a}}_n[t]\|_2 + \frac{1}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y \\ &\leq \left(1-\frac{1}{t}\right)\|\tilde{\boldsymbol{a}}_n[t]\|_2 + \frac{1}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y \qquad \text{(B.106a)} \\ &\leq \left(1-\frac{1}{t}\right)\left[\left(1-\frac{1}{t-1}\right)\|\tilde{\boldsymbol{a}}_n[t-1]\|_2 + \frac{1}{\beta_{\tilde{\ell}}(t-1)}\sqrt{PN}B_y\right] + \frac{1}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y \\ &\leq \left(\frac{t-2}{t}\right)\|\tilde{\boldsymbol{a}}_n[t-1]\|_2 + \frac{2}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y. \qquad \text{(B.106b)}\end{aligned}$$

Substituting the upper bound of $\|\tilde{\boldsymbol{a}}_n[t-1]\|_2$ using (B.106a), we have

$$\|\tilde{\boldsymbol{a}}_n[t+1]\|_2 \leq \left(\frac{t-3}{t}\right)\|\tilde{\boldsymbol{a}}_n[t-2]\|_2 + \frac{3}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y.$$

After k substitutions, the above bound can be written in terms of k as follows

$$\|\tilde{\boldsymbol{a}}_n[t+1]\|_2 \leq \left(\frac{t-k}{t}\right)\|\tilde{\boldsymbol{a}}_n[t-k+1]\|_2 + \frac{k}{\beta_{\tilde{\ell}} t}\sqrt{PN}B_y,$$

$1 \leq k \leq t - P + 1$. The bound on $\|\tilde{\boldsymbol{a}}_n[t+1]\|_2$ in terms of the initial estimate $\|\tilde{\boldsymbol{a}}_n[P]\|_2$ is obtained for $k = t - P + 1$ in the above inequality, given by

$$\begin{aligned}\|\tilde{\boldsymbol{a}}_n[t+1]\|_2 &\leq \left(\frac{P-1}{t}\right)\|\tilde{\boldsymbol{a}}_n[P]\|_2 + \frac{t-P+1}{\beta_{\tilde{\ell}}t}\sqrt{PN}B_y \\ &= \frac{\sqrt{PN}B_y}{\beta_{\tilde{\ell}}} - \frac{P-1}{\beta_{\tilde{\ell}}t}\sqrt{PN}B_y \\ &\leq \frac{\sqrt{PN}B_y}{\beta_{\tilde{\ell}}},\ t \geq P.\end{aligned}$$

This completes the proof of (B.104), the first part of the theorem. To prove the second part of the theorem, by taking the value of the gradient in (B.23), and by the triangular inequality,

$$\begin{aligned}\left\|\nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 &= \|\boldsymbol{\Phi}[t]\tilde{\boldsymbol{a}}_n[t] - \boldsymbol{r}_n[t]\|_2 \\ &\leq \|\boldsymbol{\Phi}[t]\tilde{\boldsymbol{a}}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2 \\ &\leq \lambda_{\max}(\boldsymbol{\Phi}[t])\,\|\tilde{\boldsymbol{a}}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2 \qquad \text{(B.108a)} \\ &\leq L\frac{\sqrt{PN}B_y}{\beta_{\tilde{\ell}}} + \sqrt{PN}B_y \\ &\leq \left(1 + \frac{L}{\beta_{\tilde{\ell}}}\right)\sqrt{PN}B_y.\end{aligned}$$

□

Now, we are ready to prove Theorem 4. We start from the result presented in Lemma

6. Summing both sides of (B.97) from $t = P$ to T results in

$$\begin{aligned}
&\sum_{t=P}^{T}\Big(\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\Big)\\
\leq&\sum_{t=P}^{T}\Big(\frac{\alpha_t}{2}\left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 + \Big(\frac{1}{2\alpha_t} - \frac{\beta_{\tilde{\ell}}}{2}\Big)\left\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^*[T]\right\|_2^2 - \frac{1}{2\alpha_t}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2\Big) \qquad \text{(B.109)}\\
=&\frac{1}{2}\sum_{t=P}^{T}\left(\frac{1}{\alpha_t} - \beta_{\tilde{\ell}}\right)\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t]\right\|_2^2 - \frac{1}{2}\sum_{t=P}^{T}\frac{1}{\alpha_t}\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2 + \frac{1}{2}\sum_{t=P}^{T}\alpha_t\left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2\\
=&\frac{1}{2}\sum_{k=P-1}^{T-1}\left(\frac{1}{\alpha_{k+1}} - \beta_{\tilde{\ell}}\right)\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[k+1]\right\|_2^2 - \frac{1}{2}\sum_{t=P}^{T}\frac{1}{\alpha_t}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2\\
&+\frac{1}{2}\sum_{t=P}^{T}\alpha_t\left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2\\
=&\frac{1}{2}\sum_{k=P-1}^{T-1}\frac{1}{\alpha_{k+1}}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[k+1]\right\|_2^2 + \frac{1}{2}\sum_{t=P}^{T}\alpha_t\left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 - \frac{1}{2}\sum_{t=P-1}^{T-1}\frac{1}{\alpha_t}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\right\|_2^2\\
&+\frac{1}{2\alpha_{P-1}}\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\|_2^2 - \frac{1}{2\alpha_T}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[T+1]\right\|_2^2\\
\overset{(a)}{\leq}&\frac{1}{2}\sum_{t=P-1}^{T-1}\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2\left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \beta_{\tilde{\ell}}\right) + \frac{1}{2}\sum_{t=P}^{T}\alpha_t\left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2\\
&+\frac{1}{2\alpha_{P-1}}\left\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\right\|_2^2, \qquad \text{(B.110)}
\end{aligned}$$

where the inequality in (a) results from ignoring the term $1/(2\alpha_T)\|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[T+1]\|_2^2$ and combining similar terms. To relate the l.h.s. of (B.109) and the static regret in this case, consider the definition of the static regret for TIRSO in (B.33)

$$\begin{aligned}
\tilde{R}_s^{(n)}[T] &= \sum_{t=P}^{T}\Big[\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\Big] + \sum_{t=P}^{T}\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t])\\
&= \sum_{t=P}^{T}\Big[\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\Big] + \sum_{t=P-1}^{T-1}\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]). \qquad \text{(B.111)}
\end{aligned}$$

Adding and subtracting the term $\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[T+1])$ to the r.h.s. of (B.111) and rearranging of terms results in

$$\begin{aligned}
\tilde{R}_s^{(n)}[T] &= \sum_{t=P}^{T}\Big[\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\Big]\\
&\quad+\sum_{t=P}^{T}\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[P]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[T+1])\\
&\leq \sum_{t=P}^{T}\big[\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t+1]) - \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n^*[T]) - \Omega^{(n)}(\tilde{\boldsymbol{a}}_n^*[T])\big], \qquad \text{(B.112)}
\end{aligned}$$

where $\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[P]) = 0$ and $\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[T+1]) \geq 0$ are used in the above inequality. Observe that the r.h.s. of the above inequality coincides with the l.h.s. of (B.109). Therefore, from (B.110) and (B.112), we have

$$\tilde{R}_s^{(n)}[T] \leq \frac{1}{2}\sum_{t=P-1}^{T-1} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2 \left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \beta_{\tilde{\ell}}\right) + \frac{1}{2}\sum_{t=P}^{T} \alpha_t \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 + \frac{1}{2\alpha_{P-1}} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\|_2^2 .$$

Setting $\alpha_t = 1/(\beta_{\tilde{\ell}} t)$ in the above inequality yields

$$\begin{aligned}
\tilde{R}_s^{(n)}[T] &\leq \frac{1}{2}\sum_{t=P}^{T-1} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[t+1]\|_2^2 \left(\beta_{\tilde{\ell}}(t+1) - \beta_{\tilde{\ell}} t - \beta_{\tilde{\ell}}\right) \\
&\quad + \frac{1}{2}\sum_{t=P}^{T} \frac{1}{\beta_{\tilde{\ell}} t} \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 + \frac{1}{2\alpha_{P-1}} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\|_2^2 \\
&= \frac{1}{2\beta_{\tilde{\ell}}}\sum_{t=P}^{T} \frac{1}{t} \left\|\boldsymbol{g}_t^{\tilde{\ell}}\right\|_2^2 + \frac{1}{2\alpha_{P-1}} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\|_2^2 \\
&\overset{(a)}{\leq} \frac{G_{\tilde{\ell}}^2}{2\beta_{\tilde{\ell}}}\sum_{t=P}^{T} \frac{1}{t} + \frac{1}{2\alpha_{P-1}} \|\tilde{\boldsymbol{a}}_n^*[T] - \tilde{\boldsymbol{a}}_n[P]\|_2^2 \\
&\overset{(b)}{\leq} \frac{G_{\tilde{\ell}}^2}{2\beta_{\tilde{\ell}}} (\log(T-P+1)+1) + \frac{1}{2\alpha_{P-1}} \|\tilde{\boldsymbol{a}}_n^*[T]\|_2^2 \\
&\overset{(c)}{\leq} \frac{G_{\tilde{\ell}}^2}{2\beta_{\tilde{\ell}}} (\log(T-P+1)+1) + \frac{1}{2\alpha_{P-1}} B_{\tilde{\boldsymbol{a}}}^2,
\end{aligned}$$

where in (a) the bound on the gradient given in (B.105) is used; in (b) the inequality $\sum_{t=1}^{T} 1/t \leq \log(T)+1$ and the fact $\tilde{\boldsymbol{a}}_n[P] = \boldsymbol{0}_{NP}$ is used, and (c) is obtained by using the bound from (B.79).

B.13 Proof of Theorem 5

We derive the dynamic regret of TIRSO. To this end, since $\tilde{h}_t$ is convex, we have by definition

$$\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^\circ[t]) \geq \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \left(\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right)^\top (\tilde{\boldsymbol{a}}_n^\circ[t] - \tilde{\boldsymbol{a}}_n[t]) \tag{B.113}$$

$\forall\, \tilde{\boldsymbol{a}}_n^\circ[t], \tilde{\boldsymbol{a}}_n[t]$, where $\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) = \nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \boldsymbol{u}_t$ with $\boldsymbol{u}_t \in \partial\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t])$. Rearranging (E.67) and summing both sides of the inequality from $t = P$ to T results in:

$$\sum_{t=P}^{T} \left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^\circ[t])\right] \leq \sum_{t=P}^{T} \left(\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right)^\top \cdot (\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^\circ[t]) .$$

By applying the Cauchy–Schwarz inequality on each term of the summation in the r.h.s. of the above inequality, we obtain

$$\sum_{t=P}^{T} \left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^\circ[t])\right] \leq \sum_{t=P}^{T} \left\|\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \cdot \|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^\circ[t]\|_2 . \tag{B.114}$$

The next step is to derive an upper bound on $\|\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2$. From the definition of $\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])$ and by the triangular inequality, we have

$$\|\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 \leq \|\nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 + \|\boldsymbol{u}_t\|_2 . \tag{B.115}$$

To bound $\|\nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2$, we invoke Lemma 7 and set $\alpha_t = \alpha$ to obtain

$$\begin{aligned} \|\tilde{\boldsymbol{a}}_n[t+1]\|_2 &\leq (1-\alpha\beta_{\tilde{\ell}})\,\|\tilde{\boldsymbol{a}}_n[t]\|_2 + \alpha\sqrt{PN}B_y && \text{(B.116a)}\\ &= \delta\,\|\tilde{\boldsymbol{a}}_n[t]\|_2 + \alpha\sqrt{PN}B_y, && \text{(B.116b)} \end{aligned}$$

where $\delta \triangleq 1-\alpha\beta_{\tilde{\ell}}$. Observe that for $0 < \alpha \leq 1/L$, we have $0 < \delta < 1$. Substituting (B.116b) recursively, we obtain

$$\begin{aligned} \|\tilde{\boldsymbol{a}}_n[t+1]\|_2 &\leq \delta\left(\delta\,\|\tilde{\boldsymbol{a}}_n[t-1]\|_2 + \alpha\sqrt{PN}B_y\right) + \alpha\sqrt{PN}B_y\\ &= \delta^2\,\|\tilde{\boldsymbol{a}}_n[t-1]\|_2 + \delta\alpha\sqrt{PN}B_y + \alpha\sqrt{PN}B_y\\ &\leq \delta^3\,\|\tilde{\boldsymbol{a}}_n[t-2]\|_2 + \delta^2\alpha\sqrt{PN}B_y + \delta\alpha\sqrt{PN}B_y \quad + \alpha\sqrt{PN}B_y \leq \ldots\\ &\leq \delta^k\,\|\tilde{\boldsymbol{a}}_n[t-k+1]\|_2 + \alpha\sqrt{PN}B_y\sum_{i=0}^{k-1}\delta^i, \end{aligned}$$

where $1 \leq k \leq t-P+1$. For $k = t-P+1$, the above inequality becomes

$$\begin{aligned} \|\tilde{\boldsymbol{a}}_n[t+1]\|_2 &\leq \delta^{t-P+1}\,\|\tilde{\boldsymbol{a}}_n[P]\|_2 + \alpha\sqrt{PN}B_y\sum_{i=0}^{t-P}\delta^i\\ &= \frac{\alpha\sqrt{PN}B_y\left(1-\delta^{t-P+1}\right)}{1-\delta}\\ &\leq \frac{\alpha\sqrt{PN}B_y}{1-(1-\alpha\beta_{\tilde{\ell}})} = \frac{1}{\beta_{\tilde{\ell}}}\sqrt{PN}B_y, \end{aligned}$$

which implies that $\|\nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 \leq (1+L/\beta_{\tilde{\ell}})\sqrt{PN}B_y$, as in the proof of Lemma 8 by following the same arguments as in (E.47). Next, we need to find an upper bound on $\|\boldsymbol{u}_t\|_2$ in (E.70). To this end, we apply the result in [46, Lemma 2.6] to $\Omega^{(n)}$, which establishes that all the subgradients of $\Omega^{(n)}$ are bounded by its Lipschitz continuity parameter $L_{\Omega^{(n)}}$. In the following, we show that $L_{\Omega^{(n)}} = \lambda\sqrt{N}$. Lipschitz smoothness of $\Omega^{(n)}$ means that there exists $L_{\Omega^{(n)}}$ such that

$$\left|\Omega^{(n)}(\boldsymbol{a}) - \Omega^{(n)}(\boldsymbol{b})\right| \leq L_{\Omega^{(n)}}\,\|\boldsymbol{a}-\boldsymbol{b}\|_2 , \tag{B.119}$$

for all $\boldsymbol{a}, \boldsymbol{b}$. By definition, we have $\Omega^{(n)}(\boldsymbol{x}_n) = \lambda\sum_{n'=1,n'\neq n}^{N}\|\boldsymbol{x}_{n,n'}\|_2$ with $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^\top, ..., \boldsymbol{x}_{n,N}^\top]^\top$, $\boldsymbol{x}_{n,n'} \in \mathbb{R}^P, n' = 1, ..., N$. Let $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^\top, ..., \boldsymbol{z}_{n,N}^\top]^\top, \boldsymbol{z}_{n,n'} \in \mathbb{R}^P, n' = 1, ..., N$ and by taking

the l.h.s. of (E.63), we have

$$\begin{aligned}
\left|\Omega^{(n)}(\boldsymbol{x}_n) - \Omega^{(n)}(\boldsymbol{z}_n)\right| &= \lambda \left| \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{x}_{n,n'}\|_2 - \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{z}_{n,n'}\|_2 \right| \\
&= \lambda \left| \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \left[\|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{z}_{n,n'}\|_2 \right] \right| \\
&\leq \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \left| \|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{z}_{n,n'}\|_2 \right| \qquad \text{(B.120a)} \\
&\leq \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{x}_{n,n'} - \boldsymbol{z}_{n,n'}\|_2 \qquad \text{(B.120b)} \\
&\leq \lambda \sum_{n'=1}^{N} \|\boldsymbol{x}_{n,n'} - \boldsymbol{z}_{n,n'}\|_2 \\
&\leq \lambda \sqrt{N} \, \|\boldsymbol{x}_n - \boldsymbol{z}_n\|_2 \,, \qquad \text{(B.120c)}
\end{aligned}$$

where the inequality in (E.64a) holds due to the triangle inequality for scalars ($\|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{y}_{n,n'}\|_2$ as scalars); (E.64b) holds due to the reverse triangle inequality (given by $|\|\boldsymbol{x}_1\|_2 - \|\boldsymbol{x}_2\|_2| \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$); and (E.64c) follows from the inequality $\|\boldsymbol{b}\|_1 \leq \sqrt{N}\|\boldsymbol{b}\|_2$ with $\boldsymbol{b} \in \mathbb{R}^N$ [118, Sec. 2.2.2]. The inequality in (E.64c) implies that (E.63) is satisfied with $L_{\Omega^{(n)}} = \lambda\sqrt{N}$, i.e., $\Omega^{(n)}$ is $\lambda\sqrt{N}$-Lipschitz continuous. Thus, we have $\|\tilde{\nabla}\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 \leq (1 + L/\beta_{\tilde{\ell}})\sqrt{PN}B_y + \lambda\sqrt{N}$. Substituting this bound in (E.69) leads to:

$$\sum_{t=P}^{T} \left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t]) \right] \leq \sum_{t=P}^{T} \left[\left(1 + \frac{L}{\beta_{\tilde{\ell}}}\right) \sqrt{PN} B_y + \lambda\sqrt{N} \right] \|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]\|_2 \,. \tag{B.121}$$

Next, we show that TIRSO for a constant step size can alternatively be derived by applying online proximal gradient descent to minimize $\tilde{\ell}_t^{(n)} + \Omega^{(n)}$. With $\tilde{\ell}_t^{(n)}$ given by (B.20) and $\Omega^{(n)}$ is given by (B.12b), applying the online proximal gradient algorithm with a constant step size α yields:

$$\tilde{\boldsymbol{a}}_n[t+1] = \mathbf{prox}_{\Omega^{(n)}}^{\alpha} \left(\tilde{\boldsymbol{a}}_n[t] - \alpha \nabla \tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) \right), \tag{B.122}$$

where the proximal operator of a function Ψ at point $\boldsymbol{v}$ is defined by [119]:

$$\mathbf{prox}_{\Psi}^{\eta}(\boldsymbol{v}) \triangleq \underset{\boldsymbol{x} \in \text{dom } \Psi}{\arg\min} \left[\Psi(\boldsymbol{x}) + \frac{1}{2\eta} \|\boldsymbol{x} - \boldsymbol{v}\|_2^2 \right]. \tag{B.123}$$

The parameter η controls the trade-off between minimizing $\Psi(\cdot)$ and being close to $\boldsymbol{v}$. According to the definition in Sec. B.3.2, $\tilde{\boldsymbol{a}}_n^{\text{f}}[t] \triangleq \tilde{\boldsymbol{a}}_n[t] - \alpha\nabla\tilde{\ell}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])$, and $\tilde{\boldsymbol{a}}_n^{\text{f}}[t] =$

$[(\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,1}[t])^\top, \ldots, (\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,N}[t])^\top]^\top$, which enables us to write the above update expression as

$$\begin{aligned}\tilde{\boldsymbol{a}}_n[t+1] &= \mathbf{prox}^{\alpha}_{\Omega^{(n)}}\left(\tilde{\boldsymbol{a}}^{\mathrm{f}}_n[t]\right)\\ &= \arg\min_{\boldsymbol{z}_n}\left(\Omega^{(n)}(\boldsymbol{z}_n) + \frac{1}{2\alpha}\left\|\boldsymbol{z}_n - \tilde{\boldsymbol{a}}^{\mathrm{f}}_n[t]\right\|_2^2\right)\\ &= \arg\min_{\{\boldsymbol{z}_{n,n'}\}_{n'=1}^N}\Bigg(\lambda\sum_{n'=1}^{N}\mathbb{1}\{n\neq n'\}\left\|\boldsymbol{z}_{n,n'}\right\|_2\\ &\quad + \frac{1}{2\alpha}\sum_{n'=1}^{N}\left\|\boldsymbol{z}_{n,n'} - \tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\right\|_2^2\Bigg).\end{aligned}$$

Observe that the above problem is separable and the solution to the n'-th problem is given by:

$$\begin{aligned}\tilde{\boldsymbol{a}}_{n,n'}[t+1] &= \arg\min_{\boldsymbol{z}_{n,n'}}\left[\mathbb{1}\{n\neq n'\}\left\|\boldsymbol{z}_{n,n'}\right\|_2 + \frac{1}{2\alpha\lambda}\left\|\boldsymbol{z}_{n,n'} - \tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\right\|_2^2\right]\\ &= \tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\left[1 - \frac{\alpha\lambda\mathbb{1}\{n\neq n'\}}{\left\|\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\right\|_2}\right]_+,\end{aligned} \tag{B.124}$$

which is the same as (B.26) with a constant step size α. Therefore, TIRSO can be equivalently derived by applying online proximal gradient descent method. Next, we apply Lemma 2 in [70] in order to bound $\sum_{t=P}^{T}\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}^{\circ}_n[t]\|_2$ in (E.71). The hypotheses of Lemma 2 are Lipschitz smoothness of $\tilde{\ell}^{(n)}_t$, Lipschitz continuity of $\Omega^{(n)}$, and strong convexity of $\tilde{\ell}^{(n)}_t$. Lipschitz continuity of $\Omega^{(n)}$ is proved in (E.64c) whereas strong convexity of $\tilde{\ell}^{(n)}_t$ is implied by the assumption A2. So we need to verify that $\tilde{\ell}^{(n)}_t$ is Lipschitz-smooth, which means that there is L' such that

$$\left\|\nabla\tilde{\ell}^{(n)}_t(\boldsymbol{a}) - \nabla\tilde{\ell}^{(n)}_t(\boldsymbol{b})\right\|_2 \leq L'\left\|\boldsymbol{a}-\boldsymbol{b}\right\|_2, \tag{B.125}$$

for all $\boldsymbol{a}, \boldsymbol{b}$. To this end, taking the l.h.s. of (B.125) and substituting the value of the gradient of $\tilde{\ell}^{(n)}_t$ from (B.23) results in:

$$\begin{aligned}\left\|\boldsymbol{\Phi}[t]\boldsymbol{a} - \boldsymbol{r}_n[t] - \boldsymbol{\Phi}[t]\boldsymbol{b} + \boldsymbol{r}_n[t]\right\|_2 &= \left\|\boldsymbol{\Phi}[t](\boldsymbol{a}-\boldsymbol{b})\right\|_2\\ &\leq \lambda_{\max}(\boldsymbol{\Phi}[t])\left\|\boldsymbol{a}-\boldsymbol{b}\right\|_2,\end{aligned}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of the input matrix. Due to assumption A3, the inequality in (B.125) holds with $L' = L$. To apply Lemma 2 in [70], one can set K in [70] as $T-P+1$, g_k as $\Omega^{(n)}$, and f_k as $\tilde{\ell}^{(n)}_{P+k-1}$, it follows that $\boldsymbol{x}_k$ in [70] equals $\tilde{\boldsymbol{a}}_n[P+k-1]$ and $\boldsymbol{x}^{\circ}_k$ equals $\tilde{\boldsymbol{a}}^{\circ}_n[P+k-1]$. Then, since we have already shown above that the hypotheses of Lemma 2 in [70] hold in our case, applying it to bound $\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}^{\circ}_n[t]\|_2$ in (E.71) yields:

$$\begin{aligned}\sum_{t=P}^{T}\left[\tilde{h}^{(n)}_t(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}^{(n)}_t(\tilde{\boldsymbol{a}}^{\circ}_n[t])\right] &\leq \frac{1}{\alpha\beta_{\tilde{\ell}}}\Bigg[\left(1+\frac{L}{\beta_{\tilde{\ell}}}\right)\sqrt{PN}B_y\\ &\quad + \lambda\sqrt{N}\Bigg]\left(\left\|\tilde{\boldsymbol{a}}_n[P] - \tilde{\boldsymbol{a}}^{\circ}_n[P]\right\|_2 + W^{(n)}[T]\right).\end{aligned} \tag{B.126}$$

Noting that $\tilde{\boldsymbol{a}}_n[P] = \mathbf{0}_{NP}$ concludes the proof.

Appendix C

Paper C

Title:	Dynamic regret analysis for online tracking of time-varying structural equation model topologies
Authors:	**Bakht Zaman**, Luis M. Lopez-Ramos, and Baltasar Beferull-Lozano
Affiliation:	Center Intelligent Signal Processing and Wireless Networks (WISENET) Department of ICT, University of Agder, Grimstad, Norway
Conference:	Accepted in IEEE Conf. Ind. Electron. Applicat.

Dynamic Regret Analysis for Online Tracking of Time-varying Structural Equation Model Topologies

Bakht Zaman, Luis M. Lopez-Ramos, and Baltasar Beferull-Lozano

***Abstract*— Identifying dependencies among variables in a complex system is an important problem in network science. Structural equation models (SEM) have been used widely in many fields for topology inference, because they are tractable and incorporate exogenous influences in the model. Topology identification based on static SEM is useful in stationary environments; however, in many applications a time-varying underlying topology is sought. This paper presents an online algorithm to track sparse time-varying topologies in dynamic environments and most importantly, performs a detailed analysis on the performance guarantees. The tracking capability is characterized in terms of a bound on the dynamic regret of the proposed algorithm. Numerical tests show that the proposed algorithm can track changes under different models of time-varying topologies.**

C.1 Introduction

Time series are generated and observed in many applications. Using multiple time series data from a complex system, identifying a structure explaining dependencies (connections) among variables is a well-motivated problem in many fields [13]. Such a networked structure may offer insights about the system dynamics and can assist in inference tasks such as prediction, event detection, and signal reconstruction [2],[120],[45].

There are different models and approaches that are extensively used in topology identification in certain applications: see, e.g., [22], [44], [120], and references therein. Among these models, structural equation model (SEM) is a popular model [24]: this is mainly due to its tractability and the ability to identify directed relations by means of the inclusion of exogenous variables, which are naturally available in many applications. These exogenous variables represent influences that do not depend on the (endogenous) variables in the model, and their inclusion contributes to the model identifiability [68]. Static SEMs have been applied to topology identification problems in various fields, e.g., gene regulatory network discovery from gene expression data [52]. However, static SEM cannot capture topology changes if the underlying dynamics are nonstationary and each observation is obtained at instants relatively spaced in time, which occurs in various applications.

In time-varying environments, a dynamic SEM can be applied [69]. A dynamic SEM is considered in [35] to track information cascades of popular news topics over social networks, which are assumed to have sparse dynamic topologies. In the same work, several online algorithms are presented, but not supported by any performance guarantees, so that their tracking capabilities are not theoretically characterized. In [121], an online algorithm for tracking dynamic topologies is proposed where the exogenous input is not fully known, also without convergence guarantees.

In this paper, an online algorithm to track the changes in dynamic SEM topologies in the lines of [35] is described and its dynamic regret is analyzed, to theoretically characterize its tracking capabilities. The dynamic regret measures the cumulative difference between the cost function evaluated at the estimates and the cost function evaluated at a sequence of time-varying optimal solutions. Specifically, we provide a bound on the dynamic regret that depends on easily measurable properties of the data, the algorithm hyper-parameters, and a metric of how much the model varies along time.

The rest of the paper is organized as follows: Sec. E.2 contains the model, problem formulation, and the derivation of the algorithm. Sec. C.3 establishes the dynamic regret bound, including its formal proofs. Numerical results are presented in Sec. C.4 and Sec. D.4 concludes the paper.

C.2 Model and Problem Formulation

Consider a networked system with N nodes, indexed by i. At each time frame indexed by t, a number C of interactions (frequently denoted as *contagions*) indexed by c are observed in the system, with y_{ic}^t denoting the intensity of the c-th contagion in node i at time t. Also, let x_{ic} denote the susceptibility of node i to external influence (infection) by contagion c. The dynamic linear structural equation model (SEM) is given by [35]:

$$y_{ic}^t = \sum_{j=1, j\neq i}^{N} a_{ij}^t y_{jc}^t + b_{ii}^t x_{ic} + e_{ic}^t, \tag{C.1}$$

for $i = 1, \ldots, N,\ c = 1, \ldots, C,\ t = 1, \ldots, T$, where the coefficients a_{ij}^t are the time-varying SEM parameters that encode the topology of the network, b_{ii}^t quantifies the level of influence of external sources on node i, and e_{ic} denotes the measurement errors and un-modeled dynamics. A pictorial representation of the SEM is presented i Fig. By defining $\boldsymbol{y}_c^t = [y_{1c}^t, \ldots, y_{Nc}^t]^\top \in \mathbb{R}^N$, $\boldsymbol{x}_c = [x_{1c}, \ldots, x_{Nc}]^\top \in \mathbb{R}^N$, $\boldsymbol{B}^t = \mathrm{diag}(\boldsymbol{b}^t) \in \mathbb{R}^{N\times N}$ with $\boldsymbol{b}^t = [b_{11}^t, \ldots, b_{NN}^t]^\top$, and $\boldsymbol{e}_c^t = [e_{1c}^t, \ldots, e_{Nc}^t]^\top \in \mathbb{R}^N$, the model in (C.1) can also be written in a compact form as:

$$\boldsymbol{y}_c^t = \boldsymbol{A}^t \boldsymbol{y}_c^t + \boldsymbol{B}^t \boldsymbol{x}_c + \boldsymbol{e}_c^t, \quad c = 1, \ldots, C. \tag{C.2}$$

The matrix $\boldsymbol{A}^t \in \mathbb{R}^{N\times N}$ can be seen as a time-varying adjacency matrix for an SEM-based network. The observations for all contagions can be collected in a matrix by defining $\boldsymbol{Y}^t = [\boldsymbol{y}_1^t, \ldots, \boldsymbol{y}_C^t] \in \mathbb{R}^{N\times C}$, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C] \in \mathbb{R}^{N\times C}$, and $\boldsymbol{E}^t = [\boldsymbol{e}_1^t, \ldots, \boldsymbol{e}_C^t] \in \mathbb{R}^{N\times C}$. The dynamic SEM takes the following form:

$$\boldsymbol{Y}^t = \boldsymbol{A}^t \boldsymbol{Y}^t + \boldsymbol{B}^t \boldsymbol{X} + \boldsymbol{E}^t. \tag{C.3}$$

The problem statement becomes: Given the observations $\{\boldsymbol{Y}^t\}_{t=1}^T$ and $\boldsymbol{X}$, find $\{\boldsymbol{A}^t\}_{t=1}^T$ and $\{\boldsymbol{B}^t\}_{t=1}^T$. Along the lines of [35], we consider the exponentially-weighted least-squares criterion:

$$f_t(\boldsymbol{A}, \boldsymbol{B}) \triangleq \frac{1}{2} \sum_{\tau=1}^{t} \gamma^{t-\tau} \left\| \boldsymbol{Y}^\tau - \boldsymbol{A}\boldsymbol{Y}^\tau - \boldsymbol{B}\boldsymbol{X} \right\|_F^2 \tag{C.4}$$

and the regularizer $\Omega(\boldsymbol{A}) \triangleq \lambda \left\|\text{vec}(\boldsymbol{A})\right\|_1$, and formulate the estimation problem as

$$\{\hat{\boldsymbol{A}}^t, \hat{\boldsymbol{B}}^t\} = \underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min}\ f_t(\boldsymbol{A},\boldsymbol{B}) + \Omega(\boldsymbol{A}) \tag{C.5a}$$

$$\text{s.t.}: a_{ii} = 0, \forall i \tag{C.5b}$$

$$b_{ij} = 0, \forall i \neq j. \tag{C.5c}$$

The parameter $\gamma \in (0,1]$ is a forgetting factor that regulates how much past information influences the solution at time t, and λ is the sparsity-promoting regularization parameter. The constraint $a_{ii} = 0$ eliminates any component of the trivial solution $\boldsymbol{A} = \boldsymbol{I}$. The constraint $b_{ij} = 0$ guarantees a diagonal $\boldsymbol{B}$, meaning that external sources for a certain node i do not affect any other node $j \neq i$. Dealing with constraints can be easily avoided if we rewrite the objective including only the nonzero elements of the matrices. We can rewrite $f_t(\boldsymbol{A},\boldsymbol{B})$ as:

$$f_t(\boldsymbol{A},\boldsymbol{B}) = \frac{1}{2}\sum_{\tau=1}^{t}\sum_{i=1}^{N}\gamma^{t-\tau}\left\|\boldsymbol{y}_i^{\tau\top} - \boldsymbol{a}_{-i}^\top \boldsymbol{Y}_{-i}^\tau - b_{ii}\boldsymbol{x}_i^\top\right\|_F^2 \tag{C.6a}$$

$$= \frac{1}{2}\sum_{\tau=1}^{t}\sum_{i=1}^{N}\gamma^{t-\tau}\left\|\boldsymbol{y}_i^{\tau\top} - [\boldsymbol{a}_{-i}^\top\ b_{ii}]\begin{bmatrix}\boldsymbol{Y}_{-i}^\tau \\ \boldsymbol{x}_i^\top\end{bmatrix}\right\|_F^2, \tag{C.6b}$$

where $\boldsymbol{y}_i^{\tau\top}$ is the i-th row of $\boldsymbol{Y}^\tau$, $\boldsymbol{x}_i^\top$ is the i-th row of $\boldsymbol{X}$, $\boldsymbol{a}_{-i}^\top$ is the i-th row of $\boldsymbol{A}$ without i-th entry, and $\boldsymbol{Y}_{-i}^\tau$ is obtained by removing the i-th row from $\boldsymbol{Y}^\tau$.

Further, we can define $\boldsymbol{v}_i \triangleq [\boldsymbol{a}_{-i}^\top\ b_{ii}]^\top$ and $\boldsymbol{Z}_i^\tau \triangleq [(\boldsymbol{Y}_{-i}^\tau)^\top\ \boldsymbol{x}_i]^\top$ to rewrite (C.6a):

$$f_t(\boldsymbol{A},\boldsymbol{B}) = \frac{1}{2}\sum_{\tau=1}^{t}\sum_{i=1}^{N}\gamma^{t-\tau}\left\|\boldsymbol{y}_i^{\tau\top} - \boldsymbol{v}_i^\top \boldsymbol{Z}_i^\tau\right\|_F^2$$

$$= \frac{1}{2}\sum_{\tau=1}^{t}\sum_{i=1}^{N}\gamma^{t-\tau}\left\|\boldsymbol{y}_i^{\tau} - (\boldsymbol{Z}_i^\tau)^\top \boldsymbol{v}_i\right\|_2^2 \tag{C.7}$$

Note that f_t in (C.6a) is separable across i (nodes), so that

$$f_t(\boldsymbol{A},\boldsymbol{B}) = \sum_{i=1}^{N} f_t^i(\boldsymbol{v}_i), \tag{C.8a}$$

$$\text{where} \qquad f_t^i(\boldsymbol{v}_i) \triangleq \frac{1}{2}\sum_{\tau=1}^{t}\gamma^{t-\tau}\left\|\boldsymbol{y}_i^{\tau} - (\boldsymbol{Z}_i^\tau)^\top \boldsymbol{v}_i\right\|_2^2. \tag{C.8b}$$

Similarly, upon defining

$$\Omega^i(\boldsymbol{v}_i) \triangleq \lambda \left\|\boldsymbol{a}_{-i}\right\|_1, \tag{C.9}$$

the regularization function is also separable across the rows of $\boldsymbol{A}$, as $\left\|\text{vec}(\boldsymbol{A})\right\|_1 = \sum_{i=1}^{N}\Omega^i(\boldsymbol{v}_i)$.

In the next subsection, the online proximal gradient algorithm in [70] will be applied to solve (C.5) leveraging the separability we just presented. Before presenting the algorithm, we re-write $f_t^i(\boldsymbol{v}_i)$ in a form that will simplify the computation of its gradient. By expanding (C.8b) and ignoring terms not dependent on $\boldsymbol{v}_i$:

$$f_t^i(\boldsymbol{v}_i) \propto \frac{1}{2}\sum_{\tau=1}^{t}\gamma^{t-\tau}\Big[\boldsymbol{v}_i^\top \boldsymbol{Z}_i^\tau (\boldsymbol{Z}_i^\tau)^\top \boldsymbol{v}_i - 2\boldsymbol{y}_i^{\tau\top}(\boldsymbol{Z}_i^\tau)^\top \boldsymbol{v}_i\Big],$$

the gradient of $f_t^i(\boldsymbol{v}_i)$ is given by

$$\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i) = \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t \boldsymbol{v}_i - \boldsymbol{r}_i^t, \tag{C.10a}$$

$$\text{where } \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t \triangleq \sum_{\tau=1}^{t} \gamma^{t-\tau} \boldsymbol{Z}_i^\tau (\boldsymbol{Z}_i^\tau)^\top \tag{C.10b}$$

$$\text{and } \boldsymbol{r}_i^t \triangleq \sum_{\tau=1}^{t} \gamma^{t-\tau} \boldsymbol{Z}_i^\tau (\boldsymbol{y}_i^{\tau\top})^\top. \tag{C.10c}$$

Note that the variables defined in the latter two expressions can be computed recursively, as will be expressed in the tabulated algorithm (lines 5 and 6).

C.2.1 Proximal online gradient algorithm

The update of the proximal online gradient descent algorithm [70], applied to the i-th portion of the separable problem presented in the previous section, yields

$$\boldsymbol{v}_i[t+1] = \mathbf{prox}_{\Omega_i}^{\alpha} \left(\boldsymbol{g}_i^{\alpha}[t](\boldsymbol{v}_i[t]) \right), \tag{C.11}$$

where $\alpha > 0$, $\boldsymbol{g}_i^{\alpha}[t](\boldsymbol{u}) \triangleq \boldsymbol{u} - \alpha \nabla_{\boldsymbol{u}} f_i^t(\boldsymbol{u})$, and

$$\mathbf{prox}_{\Psi}^{\alpha}(\boldsymbol{w}) \triangleq \underset{\boldsymbol{s} \in \mathrm{dom}\Psi}{\arg\min} \left[\Psi(\boldsymbol{s}) + \frac{1}{2\alpha} \|\boldsymbol{s} - \boldsymbol{w}\|_2^2 \right]. \tag{C.12}$$

From the definition of $\boldsymbol{v}_i$ and (C.9), it becomes clear that

$$\mathbf{prox}_{\Omega_i}^{\alpha}(\boldsymbol{s}) = \left[S_{\alpha\lambda}([\boldsymbol{s}]_{1:N-1})^\top [\boldsymbol{s}]_N \right]^\top \tag{C.13}$$

with $S_{\alpha\lambda}(\boldsymbol{w})$ denoting the standard soft-thresholding operator. The complete procedure is presented in **Algorithm 11**. Observe that the step size α is required to be small enough, specifically $\alpha < 1/L_f$ where $\lambda_{\max}(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t) \leq L_f, \ \forall \ i, t$.

C.3 Dynamic Regret Analysis

The performance of online algorithms is evaluated by means of the regret, which is the difference in performance between the online algorithm and a solution which can be computed based on the data in hindsight. The regret measure can be static or dynamic. In the case of the static regret, the best comparator minimizes the objective averaged over all past instants, which implicitly assumes a stationary model. Therefore, the static regret cannot express the tracking performance of an online algorithm in dynamic environments, where the generating parameters are time-varying. To characterize the tracking performance of online algorithms, the dynamic regret [56] is used, which results from comparing the online algorithm against an optimal sequence of time-varying hindsight solutions. Specifically, upon defining $h_t(\boldsymbol{A}[t], \boldsymbol{B}[t]) \triangleq f_t(\boldsymbol{A}[t], \boldsymbol{B}[t]) + \Omega(\boldsymbol{A}[t])$, the dynamic regret is given by:

$$R_d[T] = \sum_{t=1}^{T} \left[h_t(\boldsymbol{A}[t], \boldsymbol{B}[t]) - h_t(\boldsymbol{A}^\star[t], \boldsymbol{B}^\star[t]) \right]. \tag{C.14}$$

Algorithm 11 Online algorithm for tracking dynamic SEM-based Topologies

Input: $\gamma, \lambda, \alpha \in (0, 1/L_f], \{\boldsymbol{Y}^t\}_{t=1}^T, \boldsymbol{X}$
Output: $\{\boldsymbol{A}[t]\}_{t=1}^T$, $\{\boldsymbol{B}[t]\}_{t=1}^T$
Initialization:
$\boldsymbol{v}_i[1] = 0_{N\times 1}, \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^0 = \boldsymbol{0}_{N\times N},\ \boldsymbol{r}_i^0 = \boldsymbol{0}_{N\times 1}, i = 1, \ldots, N$

for $t = 1, 2, \ldots, T$ **do**
  Receive data $\boldsymbol{Y}^t$
  for $i = 1, 2, \ldots, N$ **do**
    $\boldsymbol{Z}_i^t = [(\boldsymbol{Y}_{-i}^t)^\top\ (\boldsymbol{x}_i^\top)^\top]^\top$
    $\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t = \gamma\, \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^{t-1} + \boldsymbol{Z}_i(\boldsymbol{Z}_i)^\top$
    $\boldsymbol{r}_i^t = \gamma\, \boldsymbol{r}_i^{t-1} + \boldsymbol{Z}_i^t(\boldsymbol{y}_i^{t\top})^\top$
    $\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) = \boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t \boldsymbol{v}_i[t] - \boldsymbol{r}_i^t$
    $\boldsymbol{v}_i^{\mathrm{f}}[t] = \boldsymbol{v}_i[t] - \alpha \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t])$
    $\boldsymbol{v}_i[t+1] = \mathbf{prox}_{\Omega^i}^{\alpha}\left(\boldsymbol{v}_i^{\mathrm{f}}[t]\right)$
  end for
  end for
  Form $\boldsymbol{A}[t]$ and $\boldsymbol{B}[t]$ from $\boldsymbol{v}_i[t], i = 1, ..., N$
end for

with $(\boldsymbol{A}^\star[t], \boldsymbol{B}^\star[t])$ representing the estimate produced by a clairvoyant that knows $h_t(\cdot)$ in advance (in contrast, the online algorithm does not have access to $h_t(\cdot)$ while producing $(\boldsymbol{A}[t], \boldsymbol{B}[t])$). Using (C.6a), the above expression can be written as:

$$\begin{aligned} R_d[T] &= \sum_{t=1}^{T}\sum_{i=1}^{N} \left[f_t^i(\boldsymbol{v}_i[t]) + \Omega^i(\boldsymbol{v}_i[t]) - f_t^i(\boldsymbol{v}_i^\star[t]) - \Omega^i(\boldsymbol{v}_i^\star[t])\right] \\ &= \sum_{t=1}^{T}\sum_{i=1}^{N} \left[h_t^i(\boldsymbol{v}_i[t]) - h_t^i(\boldsymbol{v}_i^\star[t])\right] = \sum_{i=1}^{N} R_d^i[T], \end{aligned}$$

where $h_t^i(\boldsymbol{v}_i[t]) \triangleq f_t^i(\boldsymbol{v}_i[t]) + \Omega^i(\boldsymbol{v}_i[t])$, $\boldsymbol{v}_i^\star[t] \triangleq \arg\min_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i) + \Omega^i(\boldsymbol{v}_i)$, and $R_d^i[T] \triangleq \sum_{t=1}^T [h_t^i(\boldsymbol{v}_i[t]) - h_t^i(\boldsymbol{v}_i^\star[t])]$. Observe that the regret expression is separable across index i (nodes). Thus, for the sake of simplicity, we derive the regret for the i-th node, i.e., $R_d^i[T]$. The total regret will be obtained by adding the individual regret expressions. We define the path length for each subproblem (corresponding to each node i) as:

$$W_i[T] \triangleq \sum_{t=2}^{T} \left\|\boldsymbol{v}_i^\star[t] - \boldsymbol{v}_i^\star[t-1]\right\|_2, \tag{C.15}$$

which represents the aggregated variations in the consecutive optimal solutions. In this work, the following assumptions are considered:

A1. *Bounded process:* There exists B_{xy} such that $|y_{ic}^t|^2 \le B_{xy}$ and $|x_{ic}|^2 \le B_{xy}$, $\quad \forall\, i, c, t.$

A2. *Strong convexity:* Each function f_t^i is β-strongly convex, i.e., $\lambda_{\min}(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t) \ge \beta > 0, \forall\, i, t.$

A3. *Lipschitz smoothness:* Each function f_t^i is L_f-Lipschitz smooth, i.e., $\lambda_{\max}(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t) \leq L_f,\ \forall\ i,t.$

A4. *Bounded variations of the optimal solution:* The distance between two consecutive optimal solution is bounded, i.e.,

$$\|\boldsymbol{v}_i^\star[t] - \boldsymbol{v}_i^\star[t+1]\|_2 \leq d, d \geq 0, \forall\, t, i. \tag{C.16}$$

These above assumptions are standard in the literature. Assumption A1 does not entail any loss of generality and is satisfied in most real-world applications. Next, we present an upper bound on the dynamic regret.

Theorem 1. *The individual dynamic regret of* ***Algorithm 11*** *for a node i is given by:*

$$R_d^i[T] = D_h\left(\|\boldsymbol{v}_i^\star[1]\|_2 + W_i[T]\right), \tag{C.17}$$

where

$$D_h \triangleq \frac{1}{\alpha\beta}\left(\frac{B_{xy}C\sqrt{N}}{1-\gamma}\left(1+\frac{L_f}{\beta}\right) + \lambda\sqrt{N-1}\right), \tag{C.18}$$

under assumptions A1, A2, A3, and A4.

Proof. Since h_t^i is convex, we have by definition that:

$$h_t^i(\boldsymbol{v}_i^\star[t]) \geq h_t^i(\boldsymbol{v}_i[t]) + (\tilde{\nabla} h_t^i(\boldsymbol{v}_i[t]))^\top(\boldsymbol{v}_i^\star[t] - \boldsymbol{v}_i[t]), \tag{C.19}$$

$\forall\, \boldsymbol{v}_i[t], \boldsymbol{v}_i^\star[t]$, $\tilde{\nabla} h_t^i(\boldsymbol{v}_i[t])$ denotes a subgradient of $h_t^i(\boldsymbol{v}_i[t])$ given by $\tilde{\nabla} h_t^i(\boldsymbol{u}) = \nabla f_t^i(\boldsymbol{u}) + \tilde{\nabla}\Omega^i(\boldsymbol{u})$ with $\tilde{\nabla}\Omega^i(\boldsymbol{u}) \in \partial\Omega^i(\boldsymbol{u})$. Rearranging and summing the above inequality from $t = 1$ to T, we have

$$\begin{aligned}\sum_{t=1}^{T}\left[h_t^i(\boldsymbol{v}_i[t]) - h_t^i(\boldsymbol{v}_i^\star)\right] &\leq \sum_{t=1}^{T}\left(\tilde{\nabla} h_t^i(\boldsymbol{v}_i[t])\right)^\top(\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]) \\ &\leq \sum_{t=1}^{T}\left\|\tilde{\nabla} h_t^i(\boldsymbol{v}_i[t])\right\|_2 \cdot \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2\,,\end{aligned} \tag{C.20}$$

where the second inequality follows from the Cauchy-Schwarz inequality. Next, we derive a bound on $\|\tilde{\nabla} h_t^i(\boldsymbol{v}_i[t])\|_2$. Note first that it holds that

$$\left\|\tilde{\nabla}(h_t^i(\boldsymbol{v}_i[t])\right\|_2 \leq \left\|\nabla f_t^i(\boldsymbol{v}_i[t])\right\|_2 + \left\|\tilde{\nabla}\Omega^i(\boldsymbol{v}_i[t])\right\|_2. \tag{C.21}$$

Thus, we have to prove that $\|\nabla f_t^i(\boldsymbol{v}_i[t])\|_2$ and $\|\tilde{\nabla}\Omega^i(\boldsymbol{v}_i[t])\|_2$ are bounded $\forall\, \boldsymbol{v}_i[t]$. First, we prove that $\|\nabla f_t^i(\boldsymbol{v}_i[t])\|_2$ is bounded. To this end, from (C.10a), using the triangular inequality, the spectral radius of $\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t$, and assumption A3, we obtain the following:

$$\begin{aligned}\left\|\nabla f_t^i(\boldsymbol{v}_i[t+1])\right\|_2 &= \left\|\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\boldsymbol{v}_i[t+1] - \boldsymbol{r}_i^t\right\|_2 \\ &\leq \left\|\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\boldsymbol{v}_i[t+1]\right\|_2 + \left\|\boldsymbol{r}_i^t\right\|_2 \\ &\leq \lambda_{\max}(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t)\,\|\boldsymbol{v}_i[t+1]\|_2 + \left\|\boldsymbol{r}_i^t\right\|_2 \\ &\leq L_f\,\|\boldsymbol{v}_i[t+1]\|_2 + \left\|\boldsymbol{r}_i^t\right\|_2.\end{aligned} \tag{C.22}$$

We need to derive a bound on $\|\boldsymbol{v}_i[t+1]\|_2$ and $\|\boldsymbol{r}_i^t\|_2$. First, we derive a bound on $\|\boldsymbol{r}_i^t\|_2$. From the definition of $\boldsymbol{r}_i^t$ in (C.10c), and using assumption A1, we obtain the bound as follows:

$$\begin{aligned}\left\|\boldsymbol{r}_i^t\right\|_2 &= \left\|\sum_{\tau=1}^{t}\gamma^{t-\tau}\boldsymbol{Z}_i^\tau(\boldsymbol{y}_i^{\tau\top})^\top\right\|_2 \leq \left\|\sum_{\tau=1}^{t}\gamma^{t-\tau}B_{xy}\mathbf{1}_{N\times C}\mathbf{1}_C\right\|_2 \\ &= B_{xy}\left\|\sum_{\tau=1}^{t}\gamma^{t-\tau}C\mathbf{1}_N\right\|_2 = B_{xy}C\sum_{\tau=1}^{t}\gamma^{t-\tau}\left\|\mathbf{1}_N\right\|_2 \\ &\leq \frac{B_{xy}C\sqrt{N}}{1-\gamma} = \frac{B_{xy}C\sqrt{N}}{\mu},\end{aligned} \tag{C.23}$$

where $\mu \triangleq 1-\gamma$. Thus, we have derived a bound on $\|\boldsymbol{r}_i^t\|_2$. To derive an upper bound on $\|\boldsymbol{v}_i[t+1]\|_2$, from the update expression of the algorithm, and using assumption A2, we have that:

$$\begin{aligned}\|\boldsymbol{v}_i[t+1]\|_2 &\leq \left\|\boldsymbol{v}_i[t]-\alpha\nabla f_i^t(\boldsymbol{v}_i[t])\right\|_2 \\ &= \left\|\boldsymbol{v}_i[t]-\alpha\left(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\boldsymbol{v}_i[t]-\boldsymbol{r}_i^t\right)\right\|_2 \\ &= \left\|\left(\boldsymbol{I}-\alpha\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\right)\boldsymbol{v}_i[t]+\alpha\boldsymbol{r}_i^t\right\|_2 \\ &\leq \lambda_{\max}\left(\boldsymbol{I}-\alpha\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\right)\|\boldsymbol{v}_i[t]\|_2+\alpha\left\|\boldsymbol{r}_i^t\right\|_2 \\ &= \left(1-\alpha\lambda_{\min}(\boldsymbol{\Phi}_{\boldsymbol{Z}_i}^t\right)\|\boldsymbol{v}_i[t]\|_2+\alpha\left\|\boldsymbol{r}_i^t\right\|_2 \\ &\leq (1-\alpha\beta)\|\boldsymbol{v}_i[t]\|_2+\alpha\left\|\boldsymbol{r}_i^t\right\|_2.\end{aligned} \tag{C.24}$$

Substituting the bound on $\boldsymbol{r}_i^t$ from (E.49b) in the above inequality, we obtain:

$$\begin{aligned}\|\boldsymbol{v}_i[t+1]\|_2 &\leq (1-\alpha\beta)\|\boldsymbol{v}_i[t]\|_2+\alpha\frac{B_{xy}C\sqrt{N}}{\mu} \\ &= \delta\|\boldsymbol{v}_i[t]\|_2+\frac{\alpha B_{xy}C\sqrt{N}}{\mu}.\end{aligned}$$

By recursive substitution in the above inequality:

$$\begin{aligned}\|\boldsymbol{v}_i[t+1]\|_2 &\leq \delta\left(\delta\|\boldsymbol{v}_i[t-1]\|_2+\frac{\alpha B_{xy}C\sqrt{N}}{\mu}\right)+\frac{\alpha B_{xy}C\sqrt{N}}{\mu} \\ &= \delta^2\|\boldsymbol{v}_i[t-1]\|_2+\delta\frac{\alpha B_{xy}C\sqrt{N}}{\mu}+\frac{\alpha B_{xy}C\sqrt{N}}{\mu} \\ &\leq \delta^3\|\boldsymbol{v}_i[t-2]\|_2+\frac{\alpha B_{xy}C\sqrt{N}}{\mu}(\delta^2+\delta+1)\leq\ldots \\ &\leq \delta^k\|\boldsymbol{v}_i[t-k+1]\|_2+\frac{\alpha B_{xy}C\sqrt{N}}{\mu}\sum_{i=0}^{k-1}\delta^i,\end{aligned}$$

where $1 \leq k \leq t$. For $k = t$, the above inequality becomes

$$\begin{aligned}\|\boldsymbol{v}_i[t+1]\|_2 &\leq \delta^t \|\boldsymbol{v}_i[+1]\|_2 + \frac{\alpha B_{xy} C\sqrt{N}}{\mu} \sum_{i=0}^{t-1} \delta^i \\ &= \frac{\alpha B_{xy} C\sqrt{N}}{\mu} \frac{1-\delta^t}{1-\delta} \leq \frac{\alpha B_{xy} C\sqrt{N}}{\mu} \frac{1}{\alpha\beta} \\ &= \frac{B_{xy} C\sqrt{N}}{\mu\beta}. \end{aligned} \tag{C.26}$$

By substituting the bounds from (C.26) and (E.49b) into (C.22), we obtain the bound on $\|\nabla f_t^i(\boldsymbol{v}_i[t+1])\|_2$ as follows:

$$\left\|\nabla f_t^i(\boldsymbol{v}_i[t+1])\right\|_2 \leq \frac{L_f B_{xy} C\sqrt{N}}{\mu\beta} + \frac{B_{xy} C\sqrt{N}}{\mu} \tag{C.27a}$$

$$= \frac{B_{xy} C\sqrt{N}}{\mu} \left(1 + \frac{L_f}{\beta}\right). \tag{C.27b}$$

To prove that $\|\tilde{\nabla}\Omega^i(\boldsymbol{v}_i[t])\|_2$ is bounded $\forall\, \boldsymbol{v}_i[t]$, first we compute the Lipschitz continuity parameter of Ω^i, i.e., L_Ω and then apply the result in [46, Lemma 2.6], which establishes that all the subgradients of a function are bounded by its Lipschitz continuity parameter. To find L_Ω, let $\boldsymbol{a}' \triangleq [\boldsymbol{a}^\top m]^\top, \boldsymbol{b}' \triangleq [\boldsymbol{b}^\top n]^\top, \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{N-1},\ m, n \in \mathbb{R}$. By the triangular inequality and the reverse triangular inequality, we have that:

$$\begin{aligned}\left|\Omega^i(\boldsymbol{a}') - \Omega^i(\boldsymbol{b}')\right| &= \left|\lambda \|\boldsymbol{a}\|_1 - \lambda \|\boldsymbol{b}\|_1\right| \\ &= \left|\lambda \sum_{i=1}^{N-1} \left[|a_i| - |b_i|\right]\right| \\ &\leq \lambda \sum_{i=1}^{N-1} \left||a_i| - |b_i|\right| \leq \lambda \sum_{i=1}^{N-1} |a_i - b_i| \\ &= \lambda \|\boldsymbol{a} - \boldsymbol{b}\|_1 \leq \lambda\sqrt{N-1}\, \|\boldsymbol{a} - \boldsymbol{b}\|_2 \\ &\leq \lambda\sqrt{N-1}\, \|\boldsymbol{a}' - \boldsymbol{b}'\|_2 . \end{aligned}$$

Thus, we have that $L_\Omega = \lambda\sqrt{N-1}$. Substituting these bounds in (E.70), we have

$$\left\|\tilde{\nabla}(f_t^i(\boldsymbol{v}_i)\right\|_2 + \left\|\tilde{\nabla}\Omega^i(\boldsymbol{v}_i))\right\|_2 \leq \frac{B_{xy} C\sqrt{N}}{\mu} \left(1 + \frac{L_f}{\beta}\right) + \lambda\sqrt{N-1}. \tag{C.28}$$

Substituting the above bound in (C.20), we obtain

$$\begin{aligned}&\sum_{t=1}^{T} \left[h_t^i(\boldsymbol{v}_i[t]) - h_t^i(\boldsymbol{v}_i^\star)\right] \\ &\leq \sum_{t=1}^{T} \left(\frac{B_{xy} C\sqrt{N}}{\mu} \left(1 + \frac{L_f}{\beta}\right) + \lambda\sqrt{N-1}\right) \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 \\ &= \left(\frac{B_{xy} C\sqrt{N}}{\mu} \left(1 + \frac{L_f}{\beta}\right) + \lambda\sqrt{N-1}\right) \sum_{t=1}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 . \end{aligned} \tag{C.29}$$

Next, we derive a bound on $\sum_{t=1}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2$. To this end, the first step is to prove the following result:

$$\|\boldsymbol{v}_i[t+1] - \boldsymbol{v}_i^\star[t]\|_2 \leq \rho \, \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 \,, \tag{C.30}$$

where $\rho = 1 - \alpha\beta$. To this end, squaring the l.h.s. of (C.30) and by definition of $\boldsymbol{v}_i[t+1]$, we have

$$\begin{aligned}
\|\boldsymbol{v}_i[t+1] - \boldsymbol{v}_i^\star[t]\|_2^2 &= \left\| \begin{bmatrix} \boldsymbol{a}_{-i}[t+1] - \boldsymbol{a}_{-i}^\star[t] \\ b_{ii}[t+1] - b_{ii}^\star[t] \end{bmatrix} \right\|_2^2 \\
&= \left\|\boldsymbol{a}_{-i}[t+1] - \boldsymbol{a}_{-i}^\star[t]\right\|_2^2 + \|b_{ii}[t+1] - b_{ii}^\star[t]\|_2^2 \\
&= \left\|\mathbf{prox}^{\alpha}_{\lambda\|\cdot\|_1}(\boldsymbol{a}_{-i}[t] - \alpha\nabla_{\boldsymbol{a}_{-i}} f_t^i(\boldsymbol{a}_{-i}[t])) \right. \\
&\quad \left. - \mathbf{prox}^{\alpha}_{\lambda\|\cdot\|_1}(\boldsymbol{a}_{-i}^\star[t] - \alpha\nabla_{\boldsymbol{a}_{-i}} f_t^i(\boldsymbol{a}_{-i}^\star[t]))\right\|_2^2 \\
&\quad + \left(b_{ii}[t] - \alpha\nabla_{b_{ii}} f_t^i(b_{ii}[t]) - (b_{ii}^\star[t] - \alpha\nabla_{b_{ii}} f_t^i(b_{ii}^\star[t]))\right)^2 \\
&\leq \left\|(\boldsymbol{a}_{-i}[t] - \alpha\nabla_{\boldsymbol{a}_{-i}} f_t^i(\boldsymbol{a}_{-i}[t])) - (\boldsymbol{a}_{-i}^\star[t] - \alpha\nabla_{\boldsymbol{a}_{-i}} f_t^i(\boldsymbol{a}_{-i}^\star[t]))\right\|_2^2 \\
&\quad + \left(b_{ii}[t] - \alpha\nabla_{b_{ii}} f_t^i(b_{ii}[t]) - (b_{ii}^\star[t] - \alpha\nabla_{b_{ii}} f_t^i(b_{ii}^\star[t]))\right)^2 \\
&= \left\|(\boldsymbol{v}_i[t] - \alpha\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t])) - (\boldsymbol{v}_i^\star[t] - \alpha\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t]))\right\|_2^2 \\
&= \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 + \alpha^2 \left\|\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])\right\|_2^2 \\
&\quad - 2\alpha(\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t])^\top(\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])) \\
&\leq \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 + \alpha^2 \left\|\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])\right\|_2^2 \\
&\quad - 2\alpha\Big(\frac{\beta L_f}{L_f + \beta}\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 \\
&\quad + \frac{1}{L_f + \beta}\left\|\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])\right\|_2^2\Big),
\end{aligned}$$

where the above inequality is implied by assumptions A2 and A3. Given that $\alpha \in (0, 1/L_f]$, using assumption A2, and by further simplifications, we have:

$$\begin{aligned}
\|\boldsymbol{v}_i[t+1] - \boldsymbol{v}_i^\star[t]\|_2^2 &\leq \left(1 - \frac{2\alpha\beta L_f}{L_f + \beta}\right)\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 + \left(\alpha^2 - \frac{2\alpha}{L_f + \beta}\right) \cdot \\
&\quad \left\|\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])\right\|_2^2 \\
&= \left(1 - \frac{2\alpha\beta L_f}{L_f + \beta}\right)\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 - \left(\frac{2\alpha}{L_f + \beta} - \alpha^2\right) \cdot \\
&\quad \left\|\nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i[t]) - \nabla_{\boldsymbol{v}_i} f_t^i(\boldsymbol{v}_i^\star[t])\right\|_2^2 \\
&\leq \left(1 - \frac{2\alpha\beta L_f}{L_f + \beta}\right)\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 - \beta^2\left(\frac{2\alpha}{L_f + \beta} - \alpha^2\right) \cdot \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 \\
&= \left(1 - \frac{2\alpha\beta L_f}{L_f + \beta} - \frac{2\alpha\beta^2}{L_f + \beta} + \alpha^2\beta^2\right)\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 \\
&= \left(1 - 2\alpha\beta + \alpha^2\beta^2\right)\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 \\
&= \rho^2 \, \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2^2 \,,
\end{aligned}$$

where $\rho \triangleq 1 - \alpha\beta$. Taking square root on both sides of the above inequality yields (C.30).

Next, we show that

$$\sum_{t=1}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 \leq \frac{1}{1-\rho} \left[\|\boldsymbol{v}_i[1] - \boldsymbol{v}_i^\star[1]\|_2 + W_i[T]\right]. \tag{C.33}$$

To prove the above expression, consider the cumulative gap

$$\begin{aligned}
\sum_{t=2}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 &= \sum_{t=2}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t-1] + \boldsymbol{v}_i^\star[t-1] - \boldsymbol{v}_i^\star[t]\|_2 \\
&\leq \sum_{t=2}^{T} \left[\|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t-1]\|_2 + \|\boldsymbol{v}_i^\star[t] - \boldsymbol{v}_i^\star[t-1]\|_2\right] \\
&= \sum_{t=1}^{T-1} \|\boldsymbol{v}_i[t+1] - \boldsymbol{v}_i^\star[t]\|_2 + W_i[T] \\
&\leq \sum_{t=1}^{T-1} \rho \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 + W_i[T] \\
&\leq \sum_{t=1}^{T} \rho \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 + W_i[T].
\end{aligned}$$

Adding $\|\boldsymbol{v}_i[1] - \boldsymbol{v}_i^\star[1]\|_2$ on both sides of the above inequality results in:

$$\sum_{t=1}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 \leq \sum_{t=1}^{T} \rho \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2 + \|\boldsymbol{v}_i[1] - \boldsymbol{v}_i^\star[1]\|_2 + W_i[T]. \tag{C.35}$$

By rearranging terms in the above inequality, we obtain the result in (C.33). Thus, we can substitute $\sum_{t=1}^{T} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^\star[t]\|_2$ with its bound from (C.33) into (C.29) and note that $v_i[1] = 0_{N\times 1}$ in **Algorithm 11**. This completes the proof. □

Remarks. The bound on the total dynamic regret is given by:

$$R_d[T] = D_h \sum_{i=1}^{N} \left(\|\boldsymbol{v}_i^\star[1]\|_2 + W_i[T]\right), \tag{C.36}$$

where D_h is defined in (C.18). Notice that this means that the bound on the dynamic regret is a function of the parameters of the data and the parameters of the algorithm. Moreover, for a sublinear path length, the dynamic regret of the proposed algorithm is sublinear.

C.4 Numerical Results

In this section, the performance of the algorithm is analyzed by presenting numerical tests. The experimental results are based on synthetic data.

To generate the matrices $\boldsymbol{A}^t$, a binary adjacency matrix $\boldsymbol{A}_{\text{binary}}$ is generated according to an Erdős-Rényi model with edge probability p_e. No self-loops are considered, i.e., the diagonal entries of $\boldsymbol{A}_{\text{binary}}$ are zero. Two models are considered in the simulations: a)

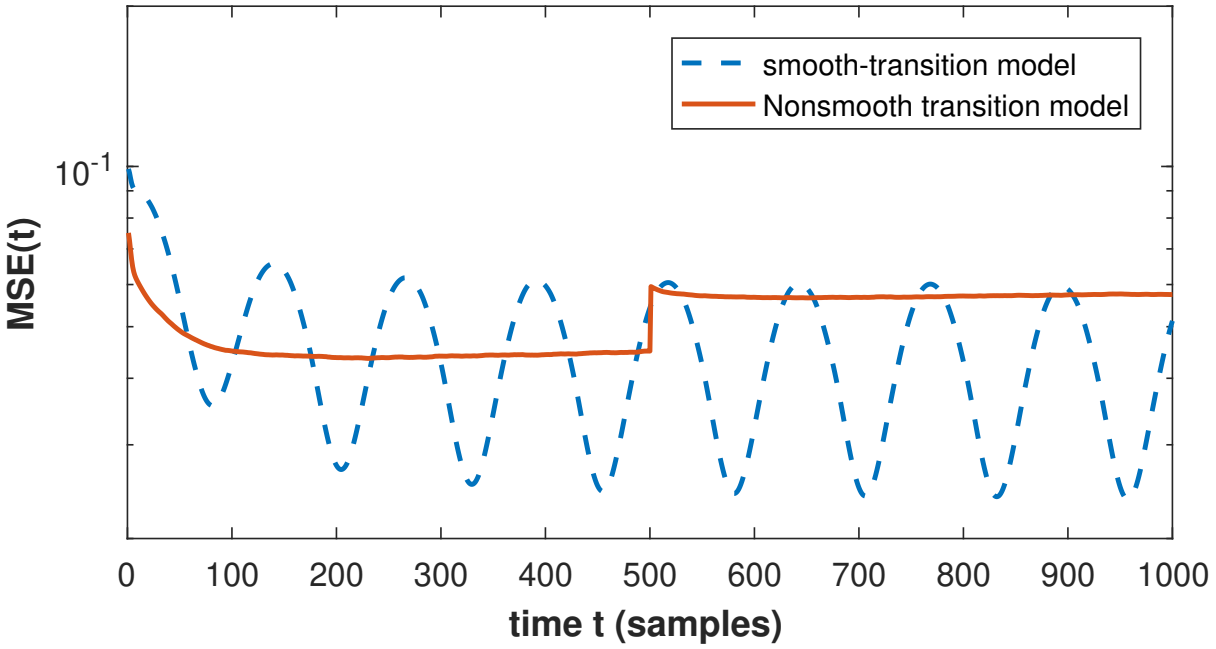


Figure C.1: MSE vs. time t. Parameters: $N = 10, p_e = 0.15, C = 5, \sigma = 0.1, \lambda = 15, \gamma = 0.9, \alpha = 1/L_f$.

smooth-transition model and b) non-smooth transition model. In the smooth-transition model, the nonzero elements of $\boldsymbol{A}^t$ follow the pattern of the 1's in $\boldsymbol{A}_{\text{binary}}$. For $t = 1$, the nonzero elements of $\boldsymbol{A}^t$ take one of the following four functions via random selection: i) $a1(t) = 0.5 + 0.5\sin(0.1t)$, ii) $a2(t) = 0.5 + 0.5\cos(0.1t)$, iii) $a3(t) = \exp(-0.01t)$, and iv) $a4(t) = 0$. For $t > 1$, the elements of $\boldsymbol{A}^t$ evolve according to the function selected initially by evaluating the functions for t. In the non-smooth transition model, a static model for $\boldsymbol{A}^t$ is considered for $t < T/2$. The nonzero elements of A^t are drawn one time from a standard Gaussian distribution. At $t = T/2$, the model changes from one to another. In both models, the matrices $\boldsymbol{B}^t$ are assumed to be constant, i.e., $\boldsymbol{B}^t = \text{diag}(\boldsymbol{b})$, where $\boldsymbol{b}$ is fixed and chosen one time randomly from a standard Gaussian distribution. This assumption means that the coefficients of external influences are constant over time, which is natural since $\boldsymbol{X}$ is constant in the model (cf. (C.3)). At each time t, for each contagion c, $\boldsymbol{e}_c^t$ is drawn from $\mathcal{N}(\boldsymbol{0}_{N\times 1}, \sigma \boldsymbol{I}_{N\times N})$. At time t, $\boldsymbol{Y}^t$ is generated using (C.3).

Fig. C.1 presents the mean-square error (MSE) given by $1/N^2 \sum_{i=1}^{N} \|\boldsymbol{v}_i[t] - \boldsymbol{v}_i^{\text{true}}[t]\|_2^2$ versus time, and Fig. C.2 shows the dynamic regret $R_d[T]$ for both models. Since the optimal solution is time-varying, the algorithm is required to track the changes in the optimal solutions. Observe from Fig. C.1 that the MSE has a decreasing trend, meaning that the proposed algorithm is able to track the changes in the time-varying topologies. Fig. C.2 shows that the dynamic regret of the non-smooth (single breaking point) transition model is lower than that of the smooth-transition model, since the model is always changing in the smooth-transition model.

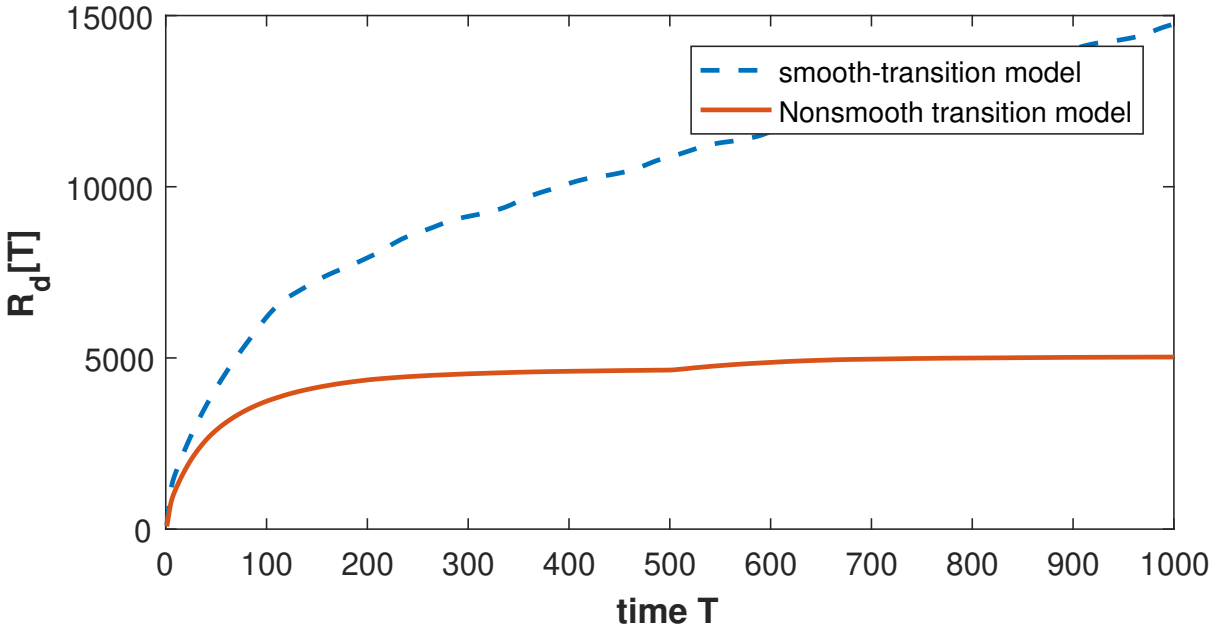


Figure C.2: Dynamic regret vs time (T). Parameters: $N = 10, p_e = 0.15, C = 5, \sigma = 0.1, \lambda = 15, \gamma = 0.9, \alpha = 1/L_f$.

C.5 Conclusion

An online algorithm for tracking dynamic SEM-based topologies is presented in this paper. A bound was derived on the dynamic regret (a much better metric than static regret for time-varying scenarios) of the proposed algorithm. This bound is a function of the numeric properties of the data that are easy to obtain, parameters of the algorithm, and the path length, which is a metric of how much the model parameters vary in a time interval. When the path length is sublinear in time, the dynamic regret of the algorithm becomes sublinear, meaning that the online algorithm enjoys a performance comparable to the optimal offline estimator. The tracking capabilities of the algorithm have been numerically validated for a time-varying scenario under two different assumptions on the model variations, namely a smooth-transition and an abrupt-transition model.

Appendix D

Paper D

Title:	Dynamic network identification from non-stationary vector autoregressive time series
Authors:	L. M. Lopez-Ramos, D. Romero, **B. Zaman**, and B. Beferull-Lozano
Affiliation:	Center Intelligent Signal Processing and Wireless Networks (WISENET) Department of ICT, University of Agder, Grimstad, Norway
Conference:	in Proc. IEEE Global Conf. Signal Inf. Process., 2018, pp. 773–777.

Dynamic Network Identification From Non-stationary Vector Autoregressive Time Series

Luis M. Lopez-Ramos, Daniel Romero, Bakht Zaman, and Baltasar Beferull-Lozano

***Abstract*— Learning the dynamics of complex systems features a large number of applications in data science. Graph-based modeling and inference underpins the most prominent family of approaches to learn complex dynamics due to their ability to capture the intrinsic sparsity of direct interactions in such systems. They also provide the user with interpretable graphs that unveil behavioral patterns and changes. To cope with the time-varying nature of interactions, this paper develops an estimation criterion and a solver to learn the parameters of a time-varying vector autoregressive model supported on a network of time series. The notion of local breakpoint is proposed to accommodate changes at individual edges. It contrasts with existing works, which assume that changes at all nodes are aligned in time. Numerical experiments validate the proposed schemes.**

D.1 Introduction

Understanding the interactions among the parts of a complex dynamic system lies at the core of data science itself and countless applications in biology, sociology, neuroscience, finance, as well as engineering realms such as cybernetics, mechatronics, and control of industrial processes. Successfully learning the presence or evolution of these interactions allows forecasting and unveils complex behaviors typically by spotting causality relations [28]. To cope with the ever increasing complexity of the analyzed systems, traditional model-based paradigms are giving way to the more contemporary data-driven perspectives, where network-based approaches enjoy great popularity due to their ability to both discern between direct and indirect causality relations as well as to provide interaction graphs amenable to intuitive human interpretation. In this context, the time-varying nature of these interactions motivates inference schemes capable of handling non-stationarity multivariate data.

Inference from multiple time series has been traditionally addressed through vector autoregressive (VAR) models [43]. To cope with non-stationarity, VAR coefficients are assumed to evolve smoothly over time [71, 72, 73], to vary according to a hidden Markov model [74], or to remain constant over time intervals separated by *structural breakpoints* [75, 76, 77, 78, 79, 42, 80]. Due to the high number of effective degrees of freedom of their models, these schemes can only satisfactorily estimate VAR coefficients if the data generating system experiences slow changes over time. To alleviate this difficulty, a natural approach is to exploit the fact that interactions among different parts of a complex system are generally *mediated*. For example, in an industrial plant where tank A is connected to B, B is connected to C, and C is not connected to A, the pressure of a fluid in a tank A affects *directly* the pressure of tank B and *indirectly* (through B) the pressure at tank C. Thus, a number of works focused on non-stationary data introduce graphs to

capture this notion of *direct* interactions, either relying on graphical models [38, 39, 14, 34] or structural equation models [26, 40]. Unfortunately, these approaches can only deal with memoryless interactions, which limits their applicability to many real-world scenarios. Schemes that do account for memory and graph structure include models based on VAR [66, 20] and structural VAR models; see [122] and references therein. However, these methods can not handle non-stationarities. To sum up, none of the aforementioned schemes identifies interaction graphs in time-varying systems with memory. To the best of our knowledge, the only exception is [44], but it can only cope with slowly changing VAR coefficients.

To alleviate these limitations, the present paper relies on a time-varying VAR (TVAR) model to propose a novel estimation criterion for non-stationary data that accounts for memory and a network structure in the interactions. The resulting estimates provide allow forcasting and *impulse response causality* analysis [43, Ch. 2]. A major novelty is the notion of *local structural breakpoint*, which captures the intuitive fact that changes in the interactions need not be synchronized across the system; in contrast to most existing works. Furthermore, a low-complexity solver is proposed to minimize the aforementioned criterion and a windowing technique is proposed to accommodate prior information on the system dynamics and reduce computational complexity.

The rest of the paper is structured as follows. Sec. D.2 introduces the model and the proposed criterion, with some practical considerations in Secs. D.2.3 and D.2.4; and Sec. D.2.5 presents an iterative solver. Numerical experiments are described in Sec. D.3 and conclusions in Sec. D.4.

D.2 Dynamic network identification

After reviewing TVAR models and introducing the notion of time-varying causality graphs, this section proposes an estimation criterion and an iterative solver. Extensions and general considerations are provided subsequently.

D.2.1 Time-varying interaction graphs

A multivariate time series is a collection $\{\mathbf{y}_t\}_{t=1}^T$ of vectors $\mathbf{y}_t := [y_{1,t}, y_{2,t}, \ldots, y_{P,t}]^\top$. The i-th (scalar) time series comprises the samples $\{y_{i,t}\}_{t=1}^T$ and can correspond e.g. with the activity over time of the i-th region of interest in a brain network, or with the measurements of the i-th sensor in a sensor network. A customary model for multivariate time series generated by non-stationary dynamic systems is the so-called L-th order TVAR model [43, Ch. 1]:

$$\mathbf{y}_t = \sum_{\ell=1}^{L} \mathbf{A}_t^{(\ell)} \mathbf{y}_{t-\ell} + \boldsymbol{\varepsilon}_t \tag{D.1}$$

where the matrix entries $\{a_{ij,t}^{(\ell)}\}_{i,j\in[1,P],t\in[1,T]}$ are the model coefficients and $\boldsymbol{\varepsilon}_{i,t}$ form the innovation process. Throughout, the notation $[m, n]$ with m and n integers satisfying $m \leq n$ will stand for $\{m, m+1, \ldots, n\}$. A time-invariant VAR model is a special case of (D.1) where $a_{ij,t}^{(\ell)} = a_{ij,t'}^{(\ell)}$ $\forall(t, t')$.

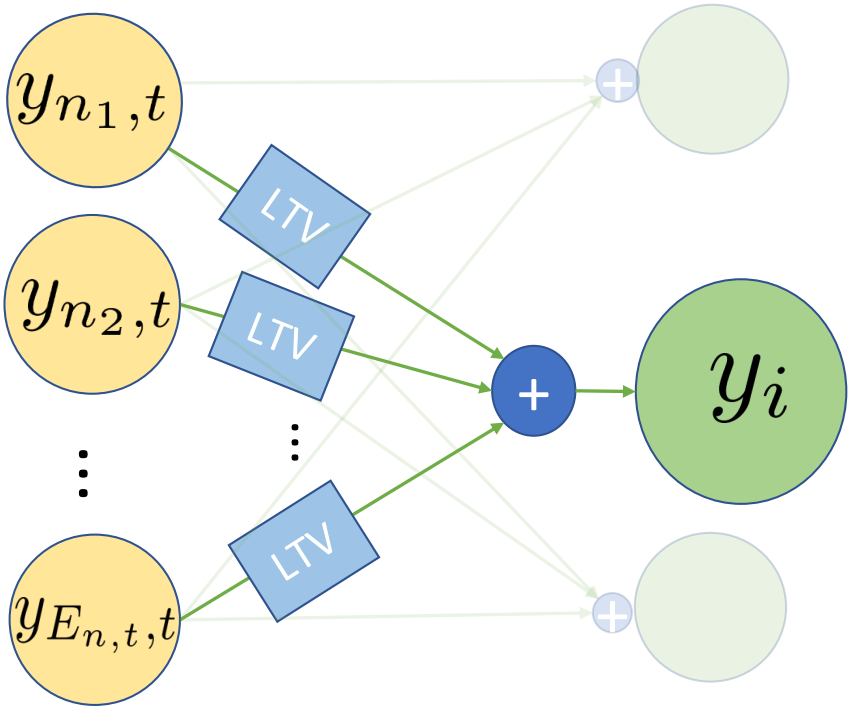


Figure D.1: Graph associated with a TVAR model.

An insightful interpretation of time-varying VAR models stems from expressing (D.1) as

$$y_{i,t} = \sum_{\ell=1}^{L} \sum_{j=1}^{P} a_{ij,t}^{(\ell)} y_{j,t-\ell} + \varepsilon_{i,t} \tag{D.2a}$$

$$= \sum_{j=1}^{P} [y_{j,t-1}, y_{j,t-2}, \ldots, y_{j,t-L}]\ \mathbf{a}_{ij,t} + \varepsilon_{i,t} \tag{D.2b}$$

where $\mathbf{a}_{ij,t} := [a_{ij,t}^{(1)}, a_{ij,t}^{(2)}, \ldots, a_{ij,t}^{(L)}]^\top$. From (D.2a), the i-th sequence $\{y_{i,t}\}_{t=1}^{T}$ equals the innovation plus the sum of all sequences $\{\{y_{p,t}\}_{t=1}^{T}\}_{p=1}^{P}$ after being filtered with a *linear time-varying* (LTV) filter with coefficients $\{a_{ij,t}^{(l)}\}_{l=1}^{L}$.

As described in Sec. D.1, interactions between time series are generally indirect (unmediated), which translates into many of these LTV filters being identically zero. To mathematically capture this interaction pattern, previous works consider the notion of graph associated with a time-invariant VAR process (see e.g. [66]), which is generalized next to *time-varying* VAR models (D.1). To this end, identify the i-th time series with the i-th vertex (or node) in the vertex set $\mathcal{V} := [1, P]$ and define the time-varying edge set as $\mathcal{E}_t := \{(i,j) \in \mathcal{V} \times \mathcal{V} : \mathbf{a}_{ij,t} \neq \mathbf{0}\}$. Thus, each edge of this time-varying graph can be thought of as an LTV filter, as depicted in Fig. D.1.

D.2.2 Proposed estimation criterion

The main goal of this paper is to estimate $\{\{\mathbf{A}_t^{(\ell)}\}_{\ell=1}^{L}\}_{t=L+1}^{T}$ given $\{\mathbf{y}_t\}_{t=1}^{T}$. Without additional assumptions, reasonable estimates cannot be found because the number of unknowns is $(T-L)P^2L$ whereas the number of samples is just PL. This difficulty is typically alleviated by assuming certain structure usually found in real-world dynamic systems. As detailed next, the structure adopted here embodies both the sparsity of causal interactions and the spatial locality of changes in those interactions.

The proposed estimation criterion is given by

$$\min_{\{\mathbf{A}_t^{(\ell)}\}} \sum_{t=L+1}^{T} \left\| \mathbf{y}_t - \sum_{\ell=1}^{L} \mathbf{A}_t^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 + \sum_{(i,j)} \left(\lambda \sum_{t=L+1}^{T} \|\mathbf{a}_{ij,t}\|_2 + \gamma \sum_{t=L+2}^{T} \|\mathbf{a}_{ij,t} - \mathbf{a}_{ij,t-1}\|_2 \right) \tag{D.3}$$

where the first term promotes estimates that fit the data and the two regularizers in parentheses are explained next. The regularization parameters $\lambda > 0$ and $\gamma > 0$ can be selected through cross-validation to balance the relative weight of data and prior information (addressed in Sec. D.2.4).

The first regularizer is a group-lasso penalty that promotes edge sparsity or, equivalently, that a large number of LTV filters $\mathbf{a}_{ij,t}$ are $\mathbf{0}$. As delineated in Secs. D.1 and D.2.1, this corresponds to the intuitive notion that most interactions in a complex network are indirect and therefore nodes are connected only with a small fraction of other nodes. This regularizer generalizes the one in [66] to time-varying graphs.

The second regularizer promotes estimates where the LTV filters remain constant over time except for a relatively small number of time instants $\mathcal{T}_{i,j} := \{t : a_{ij,t}^{(\ell)} \neq a_{ij,t-1}^{(\ell)} \text{ for some } \ell\}$ denoted as *local breakpoints*. This variant of total-variation regularizer, together with the notion of local breakpoints, constitutes one of the major novelties of this paper and contrasts with the notion of structural (or global) breakpoints, defined as $\mathcal{T} := \{t : \mathbf{A}_t^{(\ell)} \neq \mathbf{A}_{t-1}^{(\ell)} \text{for some } \ell\}$ and adopted in the literature; see e.g. [79, 42, 75, 76, 78]. These works promote solutions with few global breakpoints, and therefore all the LTV filter estimates change simultaneously at the same time for all nodes. In contrast, this work advocates promoting solutions with few *local* breakpoints, since it is expected that changes in the underlying dynamic system take place locally. For instance, in a chemical process, closing a valve between tank A and B affects the future interactions between their pressures, but does not generally affect interactions between the pressure of tanks C and D.

D.2.3 Data windowing

In practice, the time series are expected to evolve at a faster time scale than the underlying system that generates them. In many applications, such as control of industrial processes, the opposite would imply that the sampling rate needs to be increased. If this is the case, it may be beneficial to assume that $\mathbf{A}_t^{(\ell)}$ remain constant within a certain window since this would decrease the number of coefficients to estimate and therefore would improve estimation performance.

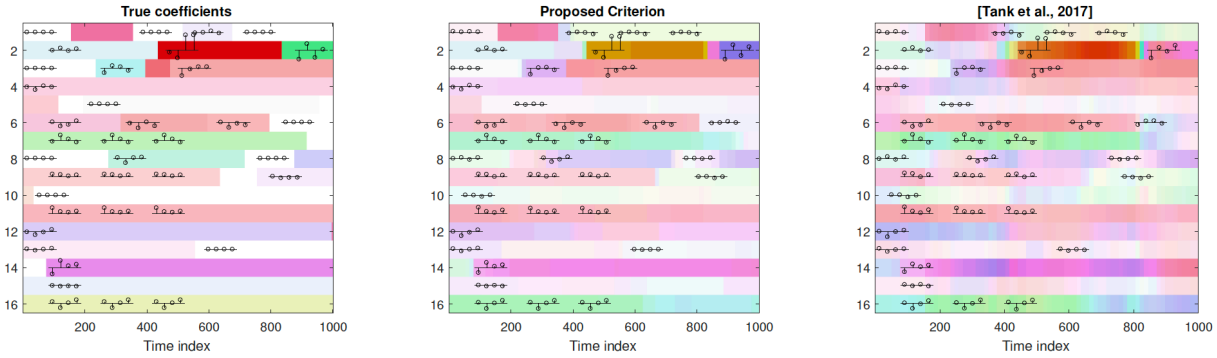


Figure D.2: Comparison between the estimates of the proposed criterion and the one in (Tank et al., 2017) [42]

To introduce this windowing technique let $\{\mathcal{W}_n\}_{n=1}^N$ be a partition of [L+1,T] into N sub-intervals (windows), and let $n(t)$ denote for each t the (unique) index such that $t \in \mathcal{W}_{n(t)}$. If $\mathbf{A}_t^{(\ell)} = \tilde{\mathbf{A}}_{n(t)}^{(\ell)}$ $\forall t$, then (D.3) becomes

$$\min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{t=L+1}^T \left\| \mathbf{y}_t - \sum_{\ell=1}^L \tilde{\mathbf{A}}_{n(t)}^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 + \sum_{(i,j)} \left(\lambda \sum_{t=L+1}^T \left\| \tilde{\mathbf{a}}_{ij,n(t)} \right\|_2 + \gamma \sum_{t=L+2}^T \left\| \tilde{\mathbf{a}}_{ij,n(t)} - \tilde{\mathbf{a}}_{ij,n(t-1)} \right\|_2 \right) \tag{D.4}$$

where $\tilde{\mathbf{a}}_{ij,t}$ is correspondingly defined in terms of $\tilde{\mathbf{A}}_t^{(\ell)}$. Absorbing scaling factors in the regularization parameters, (D.4) boils down to

$$\min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{n=1}^N \sum_{t \in \mathcal{W}_n} \left\| \mathbf{y}_t - \sum_{\ell=1}^L \tilde{\mathbf{A}}_n^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 + \sum_{(i,j)} \left(\tilde{\lambda} \sum_{n=1}^N \left\| \tilde{\mathbf{a}}_{ij,n} \right\|_2 + \tilde{\gamma} \sum_{n=2}^N \left\| \tilde{\mathbf{a}}_{ij,n} - \tilde{\mathbf{a}}_{ij,n-1} \right\|_2 \right). \tag{D.5}$$

Note that, while $LP^2(T-L)$ coefficients need to be estimated in (D.3), this number reduces to LP^2N in (D.5).

Besides an improvement in the estimation performance (D.5) when the length of the windows is attuned to the dynamics of the system, it can be shown that the objective function becomes strongly convex if windows are sufficiently large, which speeds up the convergence of the algorithm in Sec. D.2.5 (convergence becomes linear). The caveat here is a loss of temporal resolution: if one wishes to detect local breakpoints and two or more changes are produced in the same LTV filter within a single window, then the algorithm will only detect at most a single breakpoint. This effect can be counteracted by applying a screening techniques along the lines of [79].

D.2.4 Choice of parameters

Regularization parameters, in this case λ and γ, are conventionally set through cross-validation. However, such a task may be challenging when dealing with non-stationary

data. If one decides to carry out M-fold cross validation, forming M sets of consecutive samples is not appealing since the estimate of the fitting error in the validation set will become artificially high and not informative about whether the algorithm is learning changes in the VAR coefficients.

To circumvent this limitation, the proposed technique forms the aforementioned sets by skipping one out of M time samples. The estimator for the m-th fold becomes

$$\min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{n=1}^{N} \sum_{\substack{t\in\mathcal{W}_n \\ t \bmod M\neq m}} \left\| \mathbf{y}_t - \sum_{\ell=1}^{L} \tilde{\mathbf{A}}_n^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 + \sum_{(i,j)} \left(\tilde{\lambda} \sum_{n=1}^{N} \|\tilde{\mathbf{a}}_{ij,n}\|_2 + \tilde{\gamma} \sum_{n=2}^{N} \|\tilde{\mathbf{a}}_{ij,n} - \tilde{\mathbf{a}}_{ij,n-1}\|_2 \right). \tag{D.6}$$

Admittedly, all vectors $\{\mathbf{y}_t\}_t$ will still show up in all folds, but only as regressors in those folds where they are not target vectors. Indeed, this does not cause any problem from a theoretical standpoint and the performance observed in the numerical tests supports this approach.

D.2.5 Iterative solver

This section outlines the derivation of an ADMM-based algorithm proposed to solve (D.3). Define $\mathbf{Z} := \text{blkdiag}(\mathbf{x}_{q+1}^\top, \mathbf{x}_{q+2}^\top, \ldots, \mathbf{x}_T^\top)$, with $\mathbf{x}_t^\top := [\mathbf{y}_{t-1}^\top \ldots \mathbf{y}_{t-q}^\top]$; $\mathbf{B} := [\mathbf{B}_{q+1}^\top, \ldots, \mathbf{B}_T^\top]$, with $\mathbf{B}_t := \left[\mathbf{A}_t^{(1)}, \mathbf{A}_t^{(2)}, \ldots, \mathbf{A}_t^{(q)}\right]^\top$; and $\mathbf{Y} := [\mathbf{y}_{q+1}, \ldots, \mathbf{y}_T]^\top$. Then (D.3) can be rewritten as

$$\arg\min_{\mathbf{B}} \frac{1}{2} \|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_F^2 + \lambda \Omega_{GL}(\mathbf{B}) + \gamma \Omega_{GTV}(\mathbf{B}) \tag{D.7}$$

where $\Omega_{GL}(\mathbf{B}) = \sum_{(i,j)} \sum_{t=L+1}^{T} \|\mathbf{a}_{ij,t}\|_2$, and $\Omega_{GTV}(\mathbf{B}) = \sum_{(i,j)} \sum_{t=L+1}^{T} \|\mathbf{a}_{ij,t+1} - \mathbf{a}_{ij,t}\|_2$. Upon defining

$$\mathbf{D} := \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ \mathbf{0} & & \ldots & -\mathbf{I} & \mathbf{I} \end{bmatrix},$$

$\Omega_{GTV}(\mathbf{B})$ can be expressed as $\Omega_{GL}(\mathbf{D}\mathbf{B})$. This allows to rewrite (D.7) along the lines of [123] for solving via ADMM

$$\begin{aligned} \arg\min_{\mathbf{B},\boldsymbol{\Theta},\mathbf{C}} & \frac{1}{2} \|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_F^2 + \lambda \Omega_{GL}(\boldsymbol{\Theta}) + \gamma \Omega_{GL}(\mathbf{C}), \\ & \text{s.to } \mathbf{D}\mathbf{B} = \boldsymbol{\Theta},\ \mathbf{B} = \mathbf{C} \end{aligned} \tag{D.8}$$

The ADMM for the ρ-augmented Lagrangian with scaled dual variables $\mathbf{U}$ and $\mathbf{V}$

Algorithm 12 ADMM solver for dynamic network ID

Input: λ, γ, data $\{\mathbf{y}_t\}_{t=1}^T$
Output: matrix $\mathbf{B}$ containing VAR coefficients

for $k = 1, \ldots$ until convergence **do**
  Update $\mathbf{B}_t$ via (D.9a)
  for $t \in [L+1, T]$ **do**
    for $(i, j) \in [1, P]^2$ **do**
      Update $\mathbf{c}_{ij,t}, \theta_{ij,t-1}$ via (D.9b,D.9c)
    end for
  end for
  Update $\mathbf{U}, \mathbf{V}$ via (D.9d,D.9e)
end for

computes for each iteration k:

$$\mathbf{B}^{[k+1]} := \left(\mathbf{Z}^\top\mathbf{Z}/\rho + \mathbf{I} + \mathbf{D}^\top\mathbf{D}\right)^\dagger \left(\mathbf{Z}^\top\mathbf{Y}/\rho + \mathbf{C}^{[k]} - \mathbf{V}^{[k]} + \mathbf{D}^\top(\mathbf{\Theta}^{[k]} - \mathbf{U}^{[k]})\right) \tag{D.9a}$$

$$\boldsymbol{\theta}_{ij,t}^{[k+1]} := \text{prox}_{\lambda/\rho\|\cdot\|_2}(\mathbf{b}_{ij,t}^{[k+1]} - \mathbf{b}_{ij,t-1}^{[k+1]} + \mathbf{u}_{ij,t-1}^{[k+1]}) \tag{D.9b}$$

$$\mathbf{c}_{ij,t}^{[k+1]} := \text{prox}_{\lambda/\rho\|\cdot\|_2}(\mathbf{b}_{ij,t}^{[k+1]} + \mathbf{v}_{ij,t}^{[k+1]}) \tag{D.9c}$$

$$\mathbf{U}^{[k+1]} := \mathbf{U}^{[k]} + (\mathbf{D}\mathbf{B}^{[k+1]} - \mathbf{\Theta}^{[k+1]}) \tag{D.9d}$$

$$\mathbf{V}^{[k+1]} := \mathbf{V}^{[k]} + (\mathbf{B}^{[k+1]} - \mathbf{C}^{[k+1]}) \tag{D.9e}$$

and it is summarized in Proc. 12. The update (D.9a) can be efficiently computed by exploiting the tri-diagonal strucutre of $\mathbf{Z}$ and $\mathbf{D}$. The updates in (D.9b) and (D.9c) exploit the fact that the resulting prox operators are separable per (i, j) and can be expressed in terms of a group-soft-thresholding operator [124].

D.3 Numerical experiments

A simple experiment is shown next to validate the proposed estimator. An Erdös-Rényi [13] random graph $\mathcal{G}_0$ is generated with $P = 4$ nodes and an edge probability of $P_0^{(i,j)} := 0.5$ if $i \neq j$ and $P_0^{(i,j)} := 0$ if $i = j$. An $(L = 4)$-order TVAR model is generated, with initial VAR coefficients $\{\mathbf{A}_{L+1}^{(\ell)}\}_{\ell=1}^L$ over $\mathcal{G}_0$ drawn from a standard normal distribution and scaled to ensure stability [43, chapter 1]. Local breakpoints are generated at $N_b = 100$ uniformly spaced time instants $\mathcal{T}_b := \{t_{b1}, t_{b2}, \ldots, t_{bN_b}\}$, and for each $t_b \in \mathcal{T}_b$ a pair of nodes (i_b, j_b) is selected uniformly at random, generating a local breakpoint at the triplet (t_b, i_b, j_b). For each breakpoint b, the VAR coefficients $\mathbf{a}_{i_b j_b, t_b}$ and the edge set $\mathcal{E}_t$ are changed as follows: if $(i_b, j_b) \in \mathcal{E}_{t_b-1}$, $\mathbf{a}_{i_b j_b, t_b}$ is set to $\mathbf{0}$ with probability $P_z := 0.4$; otherwise, a new standard Gaussian coefficient vector $\mathbf{a}_{i_b j_b, t_b}$ is generated and scaled to keep stability. A realization of this TVAR process is generated by drawing $\{\mathbf{y}_\ell\}_{\ell=1}^L$ and $\{\boldsymbol{\varepsilon}_t\}_{t=L+1}^T$ i.i.d from a zero-mean Gaussian distribution with variance $\sigma_\epsilon^2 := 0.03$.

Fig. D.2 compares the true coefficients with the estimates obtained by the proposed criterion and the one in [42]. The latter only detects global (but not local) breakpoints. The windowing described in Sec. D.2.3 selects subperiods of length $N = 21$, and λ and γ

have been selected using the cross-validation scheme described in Sec. D.2.4, both for the proposed algorithm and the one in [42] (which only uses λ).

In each subfigure, each horizontal band corresponds to a pair of nodes, and the horizontal axis represents time. The LTV impulse response vectors $\mathbf{a}_{ij,t}/\left\|\mathbf{a}_{ij,t}\right\|$ are mapped to colors in an HSV space, being assigned similar hue if their unitary counterparts $\mathbf{a}_{ij,t}/\left\|\mathbf{a}_{ij,t}\right\|$ are closeby. The value (brightness) is set proportional to $\left\|\mathbf{a}_{ij,t}\right\|$, so responses close to $\mathbf{0}$ appear close to white, whereas impulse responses with a larger ℓ_2-norm will appear in a darker color. The stems appearing between some pairs of breakpoints represent filter coefficients of $\mathbf{a}_{ij,t}$ during the segment they lie on.

It is observed that the proposed algorithm could detect most of the local breakpoints and correctly identifies segments of stationarity. On the other hand, the competing algorithm yields a high number of false positives as expected.

D.4 Conclusions

Dynamic networks can be identified using the notion of local breakpoints, when VAR coefficient changes appear in a small number of edges. The proposed technique involves three novelties: a regularized criterion, a windowing technique, and a cross-validation scheme. Simulation experiments encourage further research along these lines.

Appendix E

Paper E

Title:	Joint Topology Identification and Signal Recovery in Non-stationary Vector Autoregressive Processes
Authors:	**Bakht Zaman**, Luis M. Lopez-Ramos, and Baltasar Beferull-Lozano
Affiliation:	Center Intelligent Signal Processing and Wireless Networks (WISENET) Department of ICT, University of Agder, Grimstad, Norway
Journal:	Submitted to IEEE Trans. Signal Process.

Online Joint Topology Identification and Signal Estimation with Inexact Proximal Online Gradient Descent

Bakht Zaman, Luis M. Lopez-Ramos, and Baltasar Beferull-Lozano

***Abstract*— Identifying the topology that underlies a set of time series is useful for tasks such as prediction, denoising, and data completion. Vector autoregressive (VAR) model-based topologies capture dependencies among time series, and often inferred from observed spatio-temporal data. When the data are affected by noise and/or missing samples, the tasks of topology identification and signal recovery (reconstruction) have to be performed jointly. Additional challenges arise when i) the underlying topology is time-varying, ii) data become available sequentially, and iii) no delay is tolerated. To overcome these challenges, this paper proposes two algorithms to estimate the VAR model-based topologies. The proposed algorithms have complementary merits in terms of complexity and performance. A performance guarantee is derived for one of the algorithms in the form of a dynamic regret bound. Numerical tests are also presented, showcasing the ability of the proposed algorithms to track the time-varying topologies with missing data in an online fashion.**

E.1 Introduction

In many applications involving complex systems, causal relations among time series are computed. These relations form a causality graph, where each node corresponds to a time series, and oftentimes reveal the topology of e.g. an underlying social, biological, or brain network [13]. A causality graph provides insights about the complex system under analysis, and enables certain tasks such as forecasting [102], signal reconstruction [3], anomaly detection [2], and dimensionality reduction [9]. While most prior work assumes that the data are fully observable at every node and time-instant, this is not the case in certain real-world scenarios [81, 82], due to diverse reasons. For instance, in sensor networks, the data at a node may be partially observed due to faulty equipments/sensors, dropped data packages due to network congestion, or under-observation of certain signals with the purpose of saving energy (e.g. sporadic observations based on the variations of the measured signal). In social networks, available user data may be partial due to security or privacy reasons. In ecological networks, uncontrollable factors such as weather conditions limit the ability to have reliable counts of a certain species. This paper considers the problem of online topology identification with streaming noisy data where some values are missing.

Identifying graphs capturing the spatio-temporal "interactions" among time series has attracted great attention in the recent literature [13, 22]. Among the popular approaches, correlation [13], partial correlations, Markov random fields, or other approaches in

graph signal processing [14, 16, 15, 11, 19, 21] are adopted in the literature. For directed interactions, one may employ structural equation models (SEM) [24], [26] or Bayesian networks [11, Sec. 8.1]. However, these methods account only for *memoryless* interactions, meaning that they cannot accommodate delayed interactions where the value of a time series at a given time instant is related to the past values of other time series.

An important notion of causality among time series is due to Granger [28] based on optimal prediction error, which is generally difficult to determine optimally [60, p. 33], [61]. Thus, alternative causality definitions based on vector autoregressive (VAR) models are typically preferred [103, 31, 104]. VAR topologies are estimated assuming Gaussianity and stationarity in [30, 29] and assuming sparsity in [66, 105, 20, 106]. All these approaches assume that the graph does not change over time. Since this is not the case in many applications, approaches have been devised to identify time-varying topologies, both undirected [38, 107, 41] and directed piecewise-constant [47].

The complexity of all previously discussed approaches becomes prohibitive for long observation windows since they process the entire data set at once and cannot accommodate data arriving sequentially. The modern approach to tackle these issues is *online* optimization, where an estimate is refined with every new data instance. Existing online topology identification algorithms for memoryless interactions include [34, 26, 35, 36, 37], and [32].

The problem of topology identification under incomplete data is considered in [83]. However, the underlying topology is considered to be undirected and the proposed algorithm is a batch approach. For directed graphs, the works in [84] and [85] address the batch estimation of the VAR parameters in the presence of noisy data with missing values. Other batch approaches to tackle the problem of joint signal estimation and topology identification from noisy observations include [125], where a spatio-temporal smoothness-based graph learning algorithm is proposed. The problem of online time series prediction with missing data is considered in [86], where the goal is to predict the future values; and [87], where the missing values are imputed by their estimates. Theoretical guarantees in the form of static regret bounds are presented. However, those works adopt a univariate autoregressive (AR) process model, and thus do not extract information about the relations among multiple time series. Moreover, these works consider a static (stationary) model and analyze the static regret. An approach to jointly estimate the signal and topology is presented in [88] for a structural VAR model (SVARM) when the observations contain noisy and missing values; in that work, different batch and online algorithms are proposed, and an identifiability result is stated. However, no performance guarantees showing the tracking capabilities of the proposed online algorithm are presented.

The present work proposes online algorithms to estimate the *memory-aware* causality graphs associated with a collection of time series with noisy missing data. The contributions include two complementary algorithms that estimate (track) time-varying VAR causality graphs (and therefore capture memory-based interactions) from streaming data affected by noise and missing entries. Both algorithms have fixed computational complexity per sample, which renders them suitable for sequential and big-data scenarios. The proposed algorithms have complementary merits: the first one has very low computational complexity, and the second one has improved tracking capability and is supported

by dynamic regret analysis.

More specifically, the contributions are

- C1) A first algorithm, termed *Joint Signal and Topology Identification via Sparse Online learning* (JSTISO). At each iteration, the proposed algorithm simultaneously estimates the signal (from the noisy observations with missing values) and the topology, by minimizing a carefully chosen, sequential objective function. Such a function involves signal reconstruction mismatch (how far the estimated signal is from the signal predicted by the past values and the topology), and time-variation of the estimated topology (distance between parameter estimates that are adjacent in time). A scalar hyperparameter allows to trade off between the two aforementioned metrics.

- C2) A second algorithm, named *Joint Signal and Topology Identification via Recursive Sparse Online learning* (JSTIRSO). The difference with respect to JSTISO is that the cost function that JSTIRSO optimizes is augmented with an additional term based on the cost function proposed in [45], which is in turn inspired by the celebrated recursive least squares (RLS) algorithm.

- C3) To characterize the performance of JSTIRSO when the topology is time-varying, a dynamic regret bound is derived.

- C4) Finally, performance is empirically validated through extensive experiments with synthetic and real data sets.

The rest of the paper is organized as follows: Sec. E.2 presents the model and a batch problem formulation for inferring time-varying memory-aware causality graphs. Sec. E.3 introduces the online joint tracking and signal estimation, and explains the online convex optimization approach. To solve the problem of joint signal and topology estimation in an online fashion, an approximate loss function is derived in Sec. E.4. An alternative loss function is presented in Sec. E.5 and it is argued why it is expected to yield better tracking performance. The performance of the proposed algorithm (JSTIRSO) is evaluated in the form of the dynamic regret analysis in Sec. E.6. Numerical results are presented in Sec. E.7.

Notation. Bold lowercase (uppercase) letters denote column vectors (matrices). Operators $\mathbb{E}[\cdot]$, ∂, $(\cdot)^\top$, $\text{vec}(\cdot)$, $\lambda_{\max}(\cdot)$, and $\text{diag}(\cdot)$ respectively denote expectation, subdifferential, matrix transpose, vectorization, maximum eigenvalue, and diagonal of a matrix. The operator ∇ denotes gradient and ∇^s represents a subgradient. Symbols $\mathbf{0}_N$, $\mathbf{1}_N$, $\mathbf{0}_{N\times N}$, and $\boldsymbol{I}_N$ respectively represent the all-zero vector of size N, the all-ones vector of size N, the all-zero matrix of size $N \times N$, and the size-N identity matrix. Also, $[\cdot]_+ = \max(\cdot, 0)$. Finally, $\mathbb{1}$ is the indicator satisfying $\mathbb{1}\{x\} = 1$ if x is true and $\mathbb{1}\{x\} = 0$ otherwise.

E.2 Model and Problem Formulation

Consider a collection of N time series, where $y_n[t]$, $t = 0, 1, \ldots, T-1$, denotes the value

of the n-th time series at time t. A causality graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ is a graph where the n-th vertex in $\mathcal{V} = \{1, \ldots, N\}$ is identified with the n-th time series $y_n[t]$ and there is an edge (or arc) from n' to n ($(n, n') \in \mathcal{E}$) if and only if (iff) $y_{n'}[t]$ *causes* $y_n[t]$ according to a certain causality notion. A prominent notion of causality can be defined using VAR models. To this end, consider the order-P time-varying VAR model [43]:

$$\boldsymbol{y}[t] = \sum_{p=1}^{P} \mathbf{A}_p^{(t)} \boldsymbol{y}[t-p] + \boldsymbol{u}[t], \tag{E.1}$$

where $\boldsymbol{y}[t] \triangleq [y_1[t], \ldots, y_N[t]]^\top$, $\boldsymbol{A}_p^{(t)} \in \mathbb{R}^{N \times N}, p = 1, \ldots, P$, are the matrices of time-varying VAR parameters and $\boldsymbol{u}[t] \triangleq [u_1[t], \ldots, u_N[t]]^\top$ is the *innovation process*, generally assumed to be a temporally white, zero-mean stochastic process, i.e., $\mathbb{E}[\boldsymbol{u}[t]] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{u}[t]\boldsymbol{u}^\top[\tau]] = \mathbf{0}_{N \times N}$ for $t \neq \tau$. Yet, the results in the regret analysis in Sec. E.6 hold independently of such assumptions, which benefits its generality.

With this model we can introduce the concept of *VAR causality* [49], which embodies a similar spirit to that of Granger causality, but is much less challenging to compute: given a process order P, it is said that time series $y_i[t]$ *VAR-causes* time series $y_j[t]$ iff the P most recent values of $y_i[t]$ carry information that allows to reduce the prediction MSE of $y_j[t]$ (compared to the optimal prediction based on all other time series in the set under consideration). While in the definition of Granger causality the definition of an optimal prediction is not clear, the VAR model allows a clear definition of an optimal predictor. Before continuing this discussion, let us introduce some extra notation for brevity purpose: With $a_{n,n'}^{(p)}$ the n, n'-th entry of $\boldsymbol{A}_p^{(t)}$ [1], (E.1) takes the following form

$$\begin{aligned} y_n[t] &= \sum_{n'=1}^{N} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p] + u_n[t] \\ &= \sum_{n' \in \mathcal{N}(n)} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p] + u_n[t], \end{aligned} \tag{E.2}$$

for $n = 1, \ldots, N$, where $\mathcal{N}(n) \triangleq \{n' : \boldsymbol{a}_{n,n'} \neq \mathbf{0}\}$ and

$$\boldsymbol{a}_{n,n'} \triangleq \left[a_{n,n'}^{(1)}, \ldots, a_{n,n'}^{(P)}\right]^\top. \tag{E.3}$$

When $\boldsymbol{u}[t]$ is a zero-mean and temporally white stochastic process, the term $\hat{y}_n[t] \triangleq \sum_{n' \in \mathcal{N}(n)} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p]$ in (E.2) is the *minimum mean square error estimator* of $y_n[t]$ given the previous values of all time series $\{y_{n'}[\tau],\ n' = 1, \ldots, N,\ \tau < t\}$; see e.g. [61, Sec. 12.7]. The set $\mathcal{N}(n)$ therefore collects the indices of those time series that participate in this optimal predictor of $y_n[t]$ or, alternatively, the information provided by time series $y_{n'}[t]$ with $n' \notin \mathcal{N}(n)$ is not informative to predict $y_n[t]$. This allows us to express the definition of VAR-causality in a clearer and more compact way: $y_{n'}[t]$ *VAR-causes* $y_n[t]$ whenever $n' \in \mathcal{N}(n)$. Equivalently, $y_{n'}[t]$ *VAR-causes* $y_n[t]$ if $\boldsymbol{a}_{n,n'} \neq \mathbf{0}$. VAR causality relations among the N time series can be represented using a causality graph where

[1] For brevity purpose, we drop the t for each element of $\boldsymbol{A}_p^{(t)}$

$\mathcal{E} \triangleq \{(n, n') : \boldsymbol{a}_{n,n'} \neq \boldsymbol{0}\}$. Clearly, in such a graph, $\mathcal{N}(n)$ is the in-neighborhood of node n. To quantify the strength of these causality relations, a weighted graph can be constructed by assigning e.g. the weight $\|\boldsymbol{a}_{n,n'}\|_2$ to the edge (n, n').

With these definitions, given a set of time series data in *batch* form, the problem of identifying a *time-varying* VAR causality graph for each time instant is a tracking problem. It involves more unknown variables than data and, thus, it is necessary to incorporate certain assumptions in order to aim for a solution.

More formally, the problem statement in batch form is: given the observations $\boldsymbol{y}[t]$, $t = 0, \ldots, T-1$ and the VAR process order, P, find the time-varying VAR coefficients $\{\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P\}_{t=P}^{T-1}$ such that it yields sparse topology at each time instant. It is common to make an assumption on the variations of the topologies. In this case, we assume that the variations in the topology are constrained, so that the sum of the squared norms of the difference between every two consecutive sets of parameters do not exceed a given budget B. Given the observations $\{\boldsymbol{y}[\tau]\}_{\tau=0}^{T-1}$, to estimate the time-varying sparse topologies in batch form, following [66], a batch problem can be formulated by solving the following minimization problem:

$$\underset{\{\{\boldsymbol{A}_p^{(\tau)}\}_{p=1}^P\}_{\tau=P}^{T-1}}{\arg\min} \frac{1}{2(T-P)} \sum_{t=P}^{T-1} \left\| \boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)} \boldsymbol{y}[t-p] \right\|_2^2 + \sum_{t=P}^{T-1} \Omega\left(\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P\right) \tag{E.4a}$$

$$\text{s. t.} \sum_{t=P+1}^{T-1} \left\| \text{vec}\left(\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P\right) - \text{vec}\left(\{\boldsymbol{A}_p^{(t-1)}\}_{p=1}^P\right) \right\|_2^2 \leq B, \tag{E.4b}$$

where the first term in the cost function is the least-squares loss, and the second term is a group sparsity-promoting regularization function defined as

$$\Omega\left(\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P\right) \triangleq \lambda \sum_{n=1}^{N} \sum_{n'=1}^{N} \mathbb{1}\{n' \neq n\} \left\| \boldsymbol{a}_{n,n'}^{(t)} \right\|_2, \tag{E.5}$$

where $\boldsymbol{a}_{n,n'}^{(t)}$ has the same structure as (E.3) with time-varying VAR parameters. The regularization function Ω promotes sparse edges in the causality graphs. The parameter λ is a user-defined constant that controls the sparsity in the edges of the graph. In the constraint (E.4b), the cumulative variation in consecutive solutions is bounded by a budget B. This restricts the amount of variations that the VAR model suffers during the time lapse captured by the batch data, and is necessary for the problem to have a meaningful solution (otherwise it would be very ill-posed). In this work, we consider that some data values will be missing (for reasons already stated in the introduction), and the observed values will be affected (corrupted) by measurement noise.

To formulate the problem of estimating the causality graphs with missing values and noise in the observation vector, consider a subset of $\mathcal{V}$ where the signal is observed, given by $\mathcal{M}_t \subseteq \mathcal{V}$. The (random) pattern of missing values is collected in the N-by-N diagonal matrix $\boldsymbol{M}_t$ where $M_{nn}[t], n = 1, \ldots, N$, are i.i.d. Bernoulli random variables taking value 1 with probability ρ and zero with probability $1 - \rho$. $\boldsymbol{M}_t$ is a diagonal matrix with the n-th diagonal entry being zero whenever the value at the n-th node is missing, otherwise one. Let $\tilde{\boldsymbol{y}}[t]$ be the observation obtained at time t, given by

$$\tilde{\boldsymbol{y}}[t] = \boldsymbol{M}_t \boldsymbol{y}[t] + \boldsymbol{M}_t \boldsymbol{\epsilon}[t], \tag{E.6}$$

where $\boldsymbol{\epsilon}[t]$ is the observation noise vector.
In batch setting, the problem of estimating time-varying topologies with missing values is stated as: given the noisy observations $\{\tilde{\boldsymbol{y}}[t]\}_{t=0}^{T-1}$ with missing values, and the VAR process order P, find the coefficients $\{\{\hat{\mathbf{A}}_p^{(t)}\}_{p=1}^{P}\}_{t=P}^{T-1}$ such that it yields a sparse topology. However, thanks to VAR model, is easier to estimate the topology from the observation vector directly if the missing values are reconstructed (imputed), and the topology (VAR parameters) helps in such reconstruction. Thus, a natural approach is to jointly estimate the signal and the topology.
In batch setting, the approach advocated in [88] is to solve the following problem, which includes joint estimation of the signal and the VAR coefficients:

$$\left\{\hat{\boldsymbol{y}}[t], \left\{\hat{\mathbf{A}}_p^{(t)}\right\}_{p=1}^{P}\right\}_{t=P}^{T-1} = \underset{\left\{\boldsymbol{y}[t], \{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\right\}_{t=P}^{T-1}}{\arg\min} \frac{1}{2}\sum_{t=P}^{T-1}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P}\boldsymbol{A}_p^{(t)}\,\boldsymbol{y}[t-p]\right\|_2^2$$
$$+ \frac{\nu}{2|\mathcal{M}_t|}\sum_{t=P}^{T-1}\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t\boldsymbol{y}[t]\|_2^2 + \sum_{t=P}^{T-1}\Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \beta\sum_{t=P}^{T-1}\sum_{p=1}^{P}\|\boldsymbol{A}_p^{(t)} - \boldsymbol{A}_p^{(t-1)}\|_{\mathrm{F}}^2, \quad \text{(E.7)}$$

where the first term is a least-squares (LS) fitting error for all time instants (where the t-th term in the summation fits the signal based on the P previous observations and the VAR coefficients at time t), the second term penalizes the mismatch between the observation vector and the reconstructed signal (recall that $|\mathcal{M}_t|$ is the number of nodes where the signal is observed), the third term is a regularization function that promotes sparsity in the edges, and the fourth term limits the variations in the coefficients (comes from the dualization of the constraint in (E.4)). The parameter $\nu > 0$ is a constant to control the trade-off between the prediction error based on the VAR coefficients and the mismatch between the measured samples and the signal reported after the reconstruction. The parameter λ controls the sparsity in the edges while β controls the magnitude of the cumulative norm of the difference between consecutive coefficients. The resulting problem in (E.7) is (separately) convex in $\{\boldsymbol{y}[t]\}_{t=P}^{T-1}$ and in $\{\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\}_{t=P}^{T-1}\}$, but not jointly convex. The problem in (E.7) can be solved via alternating minimization. Each problem in alternating minimization can be solved via proximal gradient descent.

In the next section, we describe solving this problem in an online fashion where the data are coming sequentially and are partially observable.

E.3 Online Signal Reconstruction and Topology Inference

The batch formulation in (E.7) uses information from all time instants to produce a sequence of reconstructed signal values and VAR parameter (topology) estimates. On the other hand, an online formulation should allow us to produce such a sequence with minimum delay and with fixed complexity (at the price of lower accuracy). Specifically, here we are interested in an algorithm that, at each time instant t, produces an estimate of $\boldsymbol{y}[t]$ and $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}$ as soon as the partial observation $\tilde{\boldsymbol{y}}[t]$ is received.

To this end, we design an online criterion such that its sum over time matches the batch objective in (E.7). As a preliminary step, define

$$\ell_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \triangleq \frac{1}{2}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \boldsymbol{y}[t-p]\right\|_2^2 + \frac{\nu}{2|\mathcal{M}_t|}\left\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t \boldsymbol{y}[t]\right\|_2^2. \quad \text{(E.8)}$$

Now we can use the expression above[2], and the definition of $\Omega(\cdot)$ from (E.5), to define the dynamic cost function:

$$c_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}, \left\{\boldsymbol{A}_p^{(t)}, \boldsymbol{A}_p^{(t-1)}\right\}_{p=1}^{P}\right) \triangleq \ell_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \beta \sum_{t=P}^{T-1}\sum_{p=1}^{P} \|\boldsymbol{A}_p^{(t)} - \boldsymbol{A}_p^{(t-1)}\|_{\mathrm{F}}^2, \quad \text{(E.9)}$$

The objective function in (E.7) can be rewritten as $\sum_t c_t(\dots)$. It becomes clear that producing an estimate of $\boldsymbol{y}[t]$ and $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$ does not only have an impact on $c_t(\cdot)$, but also on $\{c_\tau(\cdot)\}_{\tau=t}^{t+P}$. Such a coupling in time is taken into account in the framework of dynamic programming (or reinforcement learning), where the goal is to find a policy π of the form

$$\begin{aligned} \pi &: \mathbb{R}^{PN} \times \mathbb{R}^{N^2P} \times \mathbb{R}^{N} \times \mathbb{R}^{N^2} \quad \to \quad \mathbb{R}^{N} \times \mathbb{R}^{N^2P} \\ \pi&\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \left\{\hat{\boldsymbol{A}}_p^{(t-1)}\right\}_{p=1}^{P}, \tilde{\boldsymbol{y}}[t], \boldsymbol{M}_t\right) \quad \rightsquigarrow \quad \hat{\boldsymbol{y}}[t], \left\{\hat{\boldsymbol{A}}_p^{(t)}\right\}_{p=1}^{P} \end{aligned} \quad \text{(E.10)}$$

such that the cumulative cost is minimized in expectation. Learning such a policy (via e.g., deep reinforcement learning) would require a high amount of computation, and it is left out of the scope of the present paper. Instead, we propose to approximate such a policy using the much more tractable framework of online convex optimization (reviewed next). Fortunately enough, the structure of (E.9) resembles that of the composite problems that can be efficiently dealt with via proximal online gradient descent (OGD). In the next section, an approximation of the cost function discussed above will be taken in a way such that we can derive an proximal OGD update over $\{\boldsymbol{A}_p^{(t-1)}\}_{p=1}^P$.

In the remainder of this section, the theoretical background of proximal OGD and inexact proximal OGD will be introduced. In the next section, we will explain the approximations we take in order to be able to apply the inexact proximal OGD (IP-OGD) framework [70] to the online problem at hand.

E.3.1 Theoretical background: composite problems

In the sequel, we present a framework to solve stochastic, composite-objective optimization problems in an online fashion.

[2]The splitting of the arguments of ℓ_t into present and past samples will become useful in subsequent sections

Consider a sequence of functions such that each element in the sequence can be split into two parts (a loss function and a regularization function). Generally, each function in the sequence is given by

$$h_t(\boldsymbol{a}) \triangleq f_t(\boldsymbol{a}) + \Omega_t(\boldsymbol{a}), \tag{E.11}$$

where $f_t : \mathcal{X} \to \mathbb{R}$ is a general convex loss function, $\Omega_t : \mathcal{X} \to \mathbb{R}$ is the convex regularization function where $\mathcal{X}$ is a convex set. Note that the function $\Omega_t(\cdot)$ can vary with time, however, in this work, it will remain constant.

Given such a sequence of functions, the online learning setting requests to generate, at each time t, a hypothesis or estimate $\boldsymbol{a}[t]$, given the previous functions $\{h_\tau\}_{\tau=0}^{t-1}$. The quality of the proposed estimate $\boldsymbol{a}[t]$ will be assessed by $h_t(\boldsymbol{a}[t])$. Since the estimate must be delivered before h_t is made available, the possibility of generating good estimates is subject to certain assumptions on how much the sequence of optimal estimates (which is only known in hindsight) changes over time. In the context of this work, $\boldsymbol{a}[t]$ corresponds to the topology, and the online learning task corresponds to the tracking of the time-varying topologies, subject to the assumption that the topology changes slowly over time.

The performance metric usually considered in online learning algorithms for static problems is the static regret which compares the algorithm's performance with a static (constant in time) hindsight solution. It has been shown, though, that the algorithms in [45] can estimate and track the slowly time-varying solutions, yet the static regret is less relevant for inference of time-varying models. Since the optimal solution also changes with time in tracking problems, the static regret cannot accommodate time-varying optimal solutions. Therefore, the static regret is not a robust measure of cumulative error when the optimal solution varies with time. To characterize the performance of online algorithms in time-varying scenarios, a dynamic regret is proposed in tracking scenarios, where the hindsight solution is also time-varying [56]. Mathematically, the dynamic regret is defined as

$$R_d[T] \triangleq \sum_{t=1}^{T} \big[h_t(\boldsymbol{a}[t]) - h_t(\boldsymbol{a}^*[t])\big], \tag{E.12}$$

where $\boldsymbol{a}[t]$ is the estimate of the online algorithm and $\boldsymbol{a}^*[t]$ is the optimal solution at time t, given by $\boldsymbol{a}^*[t] \triangleq \arg\min_{\boldsymbol{a}} h_t(\boldsymbol{a})$. Note that optimal solutions are time-varying. Next, we present an online algorithm to solve the composite problem given in (E.11). It is well known that composite problems can be efficiently solved via proximal methods [119], [126]. Before presenting the online algorithm based on proximal operator, we first briefly discuss proximal operators. The proximal operator of a scaled function $\lambda\Psi$ at point $\boldsymbol{v}$ is defined by [119]:

$$\mathbf{prox}_{\lambda\Psi}(\boldsymbol{v}) \triangleq \underset{\boldsymbol{x}\in\text{dom }\Psi}{\arg\min} \left[\Psi(\boldsymbol{x}) + \frac{1}{2\lambda}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2\right], \tag{E.13}$$

where the function is minimized together with a quadratic proximal term. The minimization objective inside (E.13) becomes strongly convex due to the quadratic proximal term. The proximal operator of a function at point $\boldsymbol{v}$ can be interpreted as minimizing the function while being close $\boldsymbol{v}$. The parameter λ controls the trade-off between minimizing $\Psi(\cdot)$ and being close to $\boldsymbol{v}$. Based on this proximal operator, there are various algorithms

which work under very general conditions. Usually, the proximal algorithms are used to solve composite problems (differentiable plus non-differentiable term) and they exhibit good convergence guarantees. Some of the existing algorithms such as gradient descent, projected gradient descent, etc. can be shown to be special cases of proximal algorithms.

An extremely popular algorithm in proximal methods is proximal gradient descent (PGD) [119]. In PGD, the objective is split into two terms, one that is differentiable and one that is not differentiable. At each iteration, a gradient descent step is performed on the differentiable component of the objective and then the proximal operator of the non-differentiable function at the resultant vector is performed. This process is repeated until convergence. In its online version, namely proximal (OGD), only one iteration of the proximal gradient is performed at each time instant based on the available data sample, instead of running until convergence. In many cases, the full information about the cost function is not available to the algorithm. To deal with this issue, the inexact proximal OGD [70] assumes that an inexact gradient is available and the analysis of the algorithm includes the error between the true gradient and the available inexact gradient. Inexact proximal OGD performs very well in tracking time-varying parameters supported by theoretical guarantees.

E.4 Deriving an approximate loss function

We can manipulate the expressions in the previous section to turn the joint signal and topology identification problem into a composite objective problem that can be solved using the approach described above.

Our approach consists in treating, at time t, the P previous reconstructed samples, $\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}$, as random variables. Although those variables are dependent of the estimated VAR parameters, we adopt the simplifying approximation of assuming that they are *independent*. After doing so, the deterministic function $c_t(\cdot)$ is replaced with a *random* function

$$C_t\left(\boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) = \ell_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \Omega\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \beta\sum_{p=1}^{P}\|\boldsymbol{A}_p^{(t)} - \hat{\boldsymbol{A}}_p^{(t-1)}\|_{\mathrm{F}}^2, \quad \text{(E.14)}$$

which is jointly convex in its arguments. Notice that, if $\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}$ and $\tilde{\boldsymbol{y}}[t]$ were equal to the true (unobservable) signals $\{\boldsymbol{y}[\tau]\}_{\tau=t-P}^{t}$, this setting would be the same that is dealt with in [45], by direct application of proximal OGD. Since the aforementioned signal estimates are inexact versions of the true signals, in the present work we will use the inexact proximal OGD framework discussed in [70] to analyze the regret of the resulting algorithm.

Before proceeding to the formulation of the online algorithm, two remarks are in order.

Remark 1: The cost function has as inputs the signal estimate and the VAR parameters. It is assumed that the VAR parameters change smoothly with time, but we cannot assume that the signals vary smoothly with time. Recall that in each proximal

OGD iteration, a minimization is solved involving a first-order approximation of the loss ℓ_t, the (non-linearized) regularizer Ω, and a proximal term that ensures that the variable estimated at time t is close in norm to its previous estimate at time $t-1$. This proximal term should involve $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$, but not $\boldsymbol{y}[t]$.

Remark 2: As a consequence of the simplifying assumption of random independent reconstructed samples, the function $C_t(\cdot)$ becomes separable across nodes.

Fortunately, the joint optimization over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$ and $\boldsymbol{y}[t]$ can be reformulated into an optimization only over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$ as follows. Since C_t is jointly convex in both of its arguments, minimizing it can be split into first minimizing over $\boldsymbol{y}$ and then over $\{\boldsymbol{A}_p^{(t)}\}_{p=1}^P$. Then, we can write

$$\min_{\boldsymbol{y}[t],\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P} C_t\left(\boldsymbol{y}[t],\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right) = \min_{\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P} \mathcal{L}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right), \tag{E.15}$$

where

$$\mathcal{L}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right) \triangleq \min_{\boldsymbol{y}[t]} \ell_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right), \tag{E.16}$$

and the minimization in (E.16) can be solved analytically, as shown in Sec. E.4.1. Once a closed form is available for $\mathcal{L}$, a composite objective online optimization algorithm (specifically inexact proximal OGD) can be applied; its theoretical background is described in Sec. E.3.1. Observe that the loss function in (E.16) is separable across nodes, i.e.,

$$\mathcal{L}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^P\right) = \sum_{n=1}^N \mathcal{L}_t^{(n)}(\boldsymbol{a}_n[t]) = \sum_{n=1}^N \min_{y_n[t]} \ell_t^{(n)}(\hat{\boldsymbol{g}}[t], y_n[t], \boldsymbol{a}_n[t]), \tag{E.17}$$

where

$$\ell_t^{(n)}(\hat{\boldsymbol{g}}[t], y_n[t], \boldsymbol{a}_n[t]) \triangleq \frac{1}{2}\left((y_n[t] - \hat{\boldsymbol{g}}[t]^\top \boldsymbol{a}_n[t])^2 + \frac{\nu\, M_{nn}[t]}{|\mathcal{M}_t|}(y_n[t] - \tilde{y}_n[t])^2\right) \tag{E.18}$$

with

$$\hat{\boldsymbol{g}}[t] \triangleq \mathrm{vec}\Big(\big[\hat{\boldsymbol{y}}[t-1], \ldots, \hat{\boldsymbol{y}}[t-P]\big]^\top\Big), \tag{E.19}$$

and

$$\mathcal{L}_t^{(n)}(\boldsymbol{a}_n) \triangleq \min_{y_n[t]} \ell_t^{(n)}(\hat{\boldsymbol{g}}[t], y_n[t], \boldsymbol{a}_n[t]). \tag{E.20}$$

To derive the loss function, first, $y_n[t]$ is computed, i.e., signal reconstruction is performed. Then, a closed-form expression for $\mathcal{L}_t^{(n)}$ is derived.

E.4.1 Signal reconstruction

This section focuses on the (sub)problem of estimating the signal from a noisy observation vector with missing values, given a (fixed) topology. The resulting estimator is a convex combination of the signal prediction via the VAR process and signal estimation from the observation vector with missing values. More formally, the reconstruction subproblem consists in estimating $\boldsymbol{y}[t]$ given: the current data vector $\tilde{\boldsymbol{y}}[t]$, the masking matrix $\boldsymbol{M}_t$, the vector $\hat{\boldsymbol{g}}[t]$ collecting the estimates of the previous P data vectors, and the estimated VAR

coefficients $\{\mathbf{A}_p^{(t)}\}_{p=1}^P$. Notice from (E.8) that $\tilde{\boldsymbol{y}}[t]$ and $\boldsymbol{M}_t$ are implicit in the definition of $\ell_t(\cdot)$:

$$\hat{\boldsymbol{y}}[t] = \arg\min_{\boldsymbol{y}[t]} \ell_t(\hat{\boldsymbol{g}}[t], \boldsymbol{y}[t], \boldsymbol{a}_n[t]). \tag{E.21}$$

Since the above problem is separable across n, we can solve the problem separately. The solution for the n-th entry of $\hat{\boldsymbol{y}}[t]$ is $\hat{y}_n[t] = \arg\min_{y_n[t]} \ell_t^{(n)}(\hat{\boldsymbol{g}}[t], y_n[t], \boldsymbol{a}_n[t])$, which has a closed form given by

$$\hat{y}_n[t] = (1 - U_n[t])\, \hat{\boldsymbol{g}}[t]^\top \boldsymbol{a}_n[t] + U_n[t]\tilde{y}_n[t], \tag{E.22}$$

where

$$U_n[t] \triangleq \frac{\nu M_{nn}[t]}{|\mathcal{M}_t| + \nu M_{nn}[t]}. \tag{E.23}$$

Observe that $U_n[t]$ is zeros when $y_n[t]$ is missing, otherwise $U_n[t]$ is $\nu/(|\mathcal{M}_t| + \nu)$. When $y_n[t]$ is present, $U_n[t]$ is always less than 1.

The overall computational complexity for estimating $\hat{\boldsymbol{y}}[t]$ is $O(N^2P)$. This complexity can be reduced depending on the sparse structure of $\{\boldsymbol{a}_n[t]\}_{n=1}^N$. If, for instance, the number of edges is $\mathcal{O}(N)$, then the computational complexity for estimating $\hat{\boldsymbol{y}}[t]$ becomes $\mathcal{O}(NP)$ per t.

E.4.2 Loss function in closed form

Substituting the closed-form solution of $\hat{y}_n[t]$ from (E.22) into (E.20) and after simplification, we get

$$\mathcal{L}_t^{(n)}(\boldsymbol{a}[t]) = \frac{1}{2} U_n[t](\tilde{y}_n[t] - \hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t])^2. \tag{E.24}$$

The loss function we just derived will be used in Sec. E.4.3 to derive the IP-OGD iterates. Since proximal OGD involves linearizing part of the objective, in this case $\mathcal{L}_t^{(n)}(\cdot)$, which requires computing the gradient. The gradient of $\mathcal{L}_t^{(n)}$ w.r.t. $\boldsymbol{a}_n[t]$ is given by

$$\hat{\boldsymbol{v}}_n[t] \triangleq \nabla_{\boldsymbol{a}_n[t]} \mathcal{L}_t^{(n)} = U_n[t] \left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t] - \tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right). \tag{E.25}$$

E.4.3 Application of Inexact Proximal OGD to Joint Signal and Topology Estimation

Since the IP-OGD is a first-order method (i.e. only gradient is exploited), the error that comes from the observation noise and the missing values is only translated into the error in the gradient. Thus, the missing values and the noisy observations can be interpreted as we do not have access to the true gradient of the loss function.

For a general $f_t^{(n)}$ and $\Omega^{(n)}(\boldsymbol{a}_n) \triangleq \lambda \sum_{n'=1}^N \mathbb{1}\{n \neq n'\} \|\boldsymbol{a}_{n,n'}\|_2$, applying the online proximal gradient algorithm with a constant step size α yields:

$$\boldsymbol{a}_n[t+1] = \mathbf{prox}_{\Omega^{(n)}}^{\alpha} \left(\boldsymbol{a}_n[t] - \alpha \nabla f_t^{(n)}(\boldsymbol{a}_n[t])\right), \tag{E.26}$$

where the proximal operator of a function Ψ at point $\boldsymbol{v}$ is defined by [119]:

$$\mathbf{prox}_{\Psi}^{\eta}(\boldsymbol{v}) \triangleq \arg\min_{\boldsymbol{x} \in \mathrm{dom}\ \Psi} \left[\Psi(\boldsymbol{x}) + \frac{1}{2\eta} \|\boldsymbol{x} - \boldsymbol{v}\|_2^2\right]. \tag{E.27}$$

The parameter η controls the trade-off between minimizing $\Psi(\cdot)$ and being close to $\boldsymbol{v}$. When $\nabla f_t^{(n)}(\boldsymbol{a}_n[t])$ is not available and only an inexact version is accessible, the resulting algorithm is called inexact proximal OGD.

Let $\boldsymbol{a}_n^{\mathrm{f}}[t] \triangleq \boldsymbol{a}_n[t] - \alpha\nabla f_t^{(n)}(\boldsymbol{a}_n[t])$, and $\boldsymbol{a}_n^{\mathrm{f}}[t] = [(\boldsymbol{a}_{n,1}^{\mathrm{f}}[t])^\top, \ldots, (\boldsymbol{a}_{n,N}^{\mathrm{f}}[t])^\top]^\top$, which enables us to write the above update expression as

$$\begin{aligned}\boldsymbol{a}_n[t+1] &= \mathbf{prox}_{\Omega^{(n)}}^{\alpha}\left(\boldsymbol{a}_n^{\mathrm{f}}[t]\right)\\ &= \arg\min_{\boldsymbol{z}_n}\left(\Omega^{(n)}(\boldsymbol{z}_n) + \frac{1}{2\alpha}\left\|\boldsymbol{z}_n - \boldsymbol{a}_n^{\mathrm{f}}[t]\right\|_2^2\right)\\ &= \arg\min_{\{\boldsymbol{z}_{n,n'}\}_{n'=1}^N}\left(\lambda\sum_{n'=1}^N \mathbb{1}\{n\neq n'\}\left\|\boldsymbol{z}_{n,n'}\right\|_2 + \frac{1}{2\alpha}\sum_{n'=1}^N\left\|\boldsymbol{z}_{n,n'} - \boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\right\|_2^2\right).\end{aligned}$$

Observe that the above problem is separable and the solution to the n'-th problem is given by:

$$\begin{aligned}\boldsymbol{a}_{n,n'}[t+1] &= \arg\min_{\boldsymbol{z}_{n,n'}}\left[\mathbb{1}\{n\neq n'\}\left\|\boldsymbol{z}_{n,n'}\right\|_2 + \frac{1}{2\alpha\lambda}\left\|\boldsymbol{z}_{n,n'} - \boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\right\|_2^2\right]\\ &= \boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\left[1 - \frac{\alpha\lambda\mathbb{1}\{n\neq n'\}}{\left\|\boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\right\|_2}\right]_+.\end{aligned} \tag{E.28}$$

When $f_t^{(n)}$ is set to be $\mathcal{L}_t^{(n)}$, we recover JSTISO as tabulated in Algorithm 13.

Algorithm 13 Tracking time-varying topologies with missing data via JSTISO

Input: $P, \lambda, \sigma^2, \alpha, \{\hat{\boldsymbol{y}}[\tau]\}_{\tau=0}^{P-1}$
Output: $\{\boldsymbol{a}_n[t]\}_{n=1}^N$
Initialization: $\boldsymbol{a}_n[P] = \mathbf{0}, n = 1, ..., N,$

for $t = P, P+1, \ldots$ **do**
  Receive noisy data vector with missing values $\tilde{\boldsymbol{y}}[t]$
  Form $\hat{\boldsymbol{g}}[t]$ from the previously estimated $\{\hat{\boldsymbol{y}}[t-p]\}_{p=1}^P$ via (E.19)
  for $n = 1, \ldots, N$ **do**
    Compute $\hat{y}_n[t]$ using $\tilde{y}_n[t]$ via (E.22)
    $\hat{\boldsymbol{v}}_n[t] = U_n[t]\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t] - \tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right)$
    for $n' = 1, 2, \ldots, N$ **do**
      $\boldsymbol{a}_{n,n'}^{\mathrm{f}}[t] = \boldsymbol{a}_{n,n'}[t] - \alpha\hat{\boldsymbol{v}}_{n,n'}[t]$
      $\boldsymbol{a}_{n,n'}[t+1] = \boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\left[1 - \frac{\alpha\lambda\ \mathbb{1}\{n\neq n'\}}{\left\|\boldsymbol{a}_{n,n'}^{\mathrm{f}}[t]\right\|_2}\right]_+$
    end for
    $\boldsymbol{a}_n[t+1] = \left[\boldsymbol{a}_{n,1}^\top[t+1], \ldots, \boldsymbol{a}_{n,N}^\top[t+1]\right]^\top$
  end for
  Output $\{\boldsymbol{a}_n[t+1]\}_{n=1}^N$
end for

E.5 An Alternative Loss Function for Improved Tracking

The loss function in the previous approach is an instantaneous loss, which only depends on the current sample. While this keeps the complexity of the iterations very low, and may be sufficient for online estimation of a static VAR model, it is sensitive to noise and input variability, and thus it is expected to perform poorly when attempting at tracking a time-varying model. In [45], a running average loss function is designed drawing inspiration from the relation between least mean squares (LMS) and recursive least squares (RLS) to improve the tracking capabilities of the algorithm that is derived based on an instantaneous loss function. In this paper, we follow similar steps to propose a second approach, where a running average loss function is adopted, which depends on the past received signal values. In this second approach, we set the loss function as

$$\tilde{\ell}_t\left(\{\boldsymbol{y}[\tau]\}_{\tau=0}^{t-1}, \hat{\boldsymbol{y}}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) = \frac{1}{2}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[t-p]\right\|_2^2 + \frac{1}{2}\sum_{\tau=P}^{t-1} \gamma^{t-\tau}\left\|\hat{\boldsymbol{y}}[\tau] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[\tau-p]\right\|_2^2 + \frac{\nu}{2|\mathcal{M}_t|}\left\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t\boldsymbol{y}[t]\right\|_2^2, \quad \text{(E.29)}$$

where γ is a user-selected forgetting factor which controls the weight of past (reconstructed) samples of $\boldsymbol{y}[t]$. The procedure in the previous section (treating the previously reconstructed samples as a random variable, and minimizing over $\boldsymbol{y}[t]$) is applied to the alternative deterministic loss $\tilde{\ell}_t$, so we can write the random loss function $\tilde{\mathcal{L}}_t$ as

$$\tilde{\mathcal{L}}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \triangleq \min_{\boldsymbol{y}[t]} \tilde{\ell}_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=0}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right). \quad \text{(E.30)}$$

The above loss function can be written in terms of the previous loss function ℓ_t as

$$\tilde{\mathcal{L}}_t\left(\left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) = \min_{\boldsymbol{y}[t]} \ell_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=t-P}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) + \frac{1}{2}\sum_{\tau=P}^{t-1} \gamma^{t-\tau}\left\|\hat{\boldsymbol{y}}[\tau] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[\tau-p]\right\|_2^2. \quad \text{(E.31)}$$

Next, we follow the same steps as in the previous case. We start with the signal reconstruction as in Sec. E.4.1 for the present case. The minimizer of (E.30) is:

$$\begin{aligned}\hat{\boldsymbol{y}}[t] &= \arg\min_{\boldsymbol{y}[t]}\ \tilde{\ell}_t\left(\{\hat{\boldsymbol{y}}[\tau]\}_{\tau=0}^{t-1}, \boldsymbol{y}[t], \left\{\boldsymbol{A}_p^{(t)}\right\}_{p=1}^{P}\right) \\ &= \arg\min_{\boldsymbol{y}[t]} \frac{1}{2}\left\|\boldsymbol{y}[t] - \sum_{p=1}^{P} \boldsymbol{A}_p^{(t)}\, \hat{\boldsymbol{y}}[t-p]\right\|_2^2 + \frac{\nu}{2|\mathcal{M}_t|}\left\|\tilde{\boldsymbol{y}}[t] - \boldsymbol{M}_t\boldsymbol{y}[t]\right\|_2^2. \end{aligned} \quad \text{(E.32)}$$

Observe that (E.32) coincides with the reconstruction problem in (E.21) and, therefore, its solution is given by (E.22).

Next, we derive the closed-form solution for $\tilde{\mathcal{L}}_t$ used in this approach. To this end, substituting the closed-form expression of $\hat{\boldsymbol{y}}[t]$ from (E.22) into (E.30), we get

$$\tilde{\mathcal{L}}_t\left(\{\boldsymbol{A}_p^{(t)}\}_{p=1}^{P}\right) = \frac{1}{2}\sum_{n=1}^{N}\left[U_n[t](\tilde{y}_n[t]-\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t])^2\right] + \frac{1}{2}\sum_{n=1}^{N}\Big(\sum_{\tau=P}^{t-1}\gamma^{t-\tau}\hat{y}_n^2[\tau] + \gamma\boldsymbol{a}_n^\top[t]\hat{\boldsymbol{\Phi}}[t-1]\boldsymbol{a}_n[t] - 2\gamma\hat{\boldsymbol{r}}_n^\top[t-1]\boldsymbol{a}_n[t]\Big), \tag{E.33}$$

where

$$\hat{\boldsymbol{\Phi}}[t] \triangleq \sum_{\tau=P}^{t}\gamma^{t-\tau}\hat{\boldsymbol{g}}[\tau]\hat{\boldsymbol{g}}^\top[\tau], \tag{E.34a}$$

$$\hat{\boldsymbol{r}}_n[t] \triangleq \sum_{\tau=P}^{t}\gamma^{t-\tau}\hat{y}_n[\tau]\hat{\boldsymbol{g}}[\tau]. \tag{E.34b}$$

Note that $\tilde{\mathcal{L}}_t$ is separable across the nodes and can be written as:

$$\tilde{\mathcal{L}}_t(\cdot) = \sum_{n=1}^{N}\tilde{\mathcal{L}}_t^{(n)}(\cdot). \tag{E.35}$$

where

$$\tilde{\mathcal{L}}_t^{(n)}(\boldsymbol{a}_n) \triangleq \mathcal{L}_t^{(n)}(\boldsymbol{a}_n) + \sum_{\tau=P}^{t-1}\gamma^{t-\tau}\hat{y}_n^2[\tau] + \gamma\boldsymbol{a}_n^\top\hat{\boldsymbol{\Phi}}[t-1]\boldsymbol{a}_n - 2\gamma\hat{\boldsymbol{r}}_n^\top[t-1]\boldsymbol{a}_n \tag{E.36}$$

The proposed loss function will be used to derive the IP-OGD iterates, which requires computing its gradient. The gradient of $\tilde{\mathcal{L}}_t^{(n)}$ w.r.t. $\boldsymbol{a}_n[t]$ is given by

$$\nabla_{\boldsymbol{a}_n[t]}\tilde{\mathcal{L}}_t^{(n)} = U_n[t]\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t] - \tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right) + \gamma\hat{\boldsymbol{\Phi}}[t-1]\boldsymbol{a}_n[t] - \gamma\hat{\boldsymbol{r}}_n[t-1]. \tag{E.37}$$

Following similar steps to those in Sec. E.4.3, the gradient of the aforementioned loss function can be used to derive the JSTIRSO algorithm. The proposed JSTIRSO algorithm is tabulated in Alg. 14.

E.6 Performance analysis

To analyze the performance of JSTIRSO, we present analytical results in this section. First, the assumptions considered in the analysis are stated and then, two lemmas followed by the main theorem about the dynamic regret bound of JSTIRSO are presented. Moreover, a third lemma about bounding the error in the gradient is presented and discussed.

First, we define the following quantities:

$$\boldsymbol{\Phi}[t] \triangleq \sum_{\tau=P}^{t}\gamma^{t-\tau}\boldsymbol{g}[\tau]\boldsymbol{g}^\top[\tau], \tag{E.38a}$$

$$\boldsymbol{r}_n[t] \triangleq \sum_{\tau=P}^{t}\gamma^{t-\tau}\hat{y}_n[\tau]\boldsymbol{g}[\tau], \tag{E.38b}$$

Algorithm 14 Tracking time-varying topologies with missing data via JSTIRSO

Input: $\nu, \gamma, P, \lambda, \sigma^2, \alpha, \{\hat{\boldsymbol{y}}[\tau]\}_{\tau=0}^{P-1}$

Output: $\{\tilde{\boldsymbol{a}}_n[t]\}_{n=1}^{N}$

Initialization:

$\tilde{\boldsymbol{a}}_n[P] = \mathbf{0}, n = 1, ..., N$, $\boldsymbol{\Phi}[P-1] = \sigma^2 \boldsymbol{I}, \boldsymbol{r}_n[t] = \mathbf{0}, n = 1, ..., N$

for $t = P, P+1, \ldots$ **do**
  Receive noisy data vector with missing values $\tilde{\boldsymbol{y}}[t]$
  Form $\hat{\boldsymbol{g}}[t]$ from the previously estimated $\{\hat{\boldsymbol{y}}[t-p]\}_{p=1}^{P}$ via (E.19)
  $\hat{\boldsymbol{\Phi}}[t] = \gamma\, \hat{\boldsymbol{\Phi}}[t-1] + \hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^{\top}[t]$
  for $n = 1, \ldots, N$ **do**
    Compute $\hat{y}_n[t]$ using $\tilde{y}_n[t]$ via (E.22)
    $\hat{\boldsymbol{r}}_n[t] = \gamma\, \hat{\boldsymbol{r}}_n[t-1] + \tilde{y}_n[t]\, \hat{\boldsymbol{g}}[t]$
    $\hat{\boldsymbol{v}}_n[t] = U_n[t]\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^{\top}[t]\tilde{\boldsymbol{a}}_n[t] - \tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right) + \hat{\boldsymbol{\Phi}}[t-1]\tilde{\boldsymbol{a}}_n[t] - \hat{\boldsymbol{r}}_n[t-1]$
    for $n' = 1, 2, \ldots, N$ **do**
      $\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t] = \tilde{\boldsymbol{a}}_{n,n'}[t] - \alpha \hat{\boldsymbol{v}}_{n,n'}[t]$
      $\tilde{\boldsymbol{a}}_{n,n'}[t+1] = \tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t] \left[1 - \frac{\alpha\lambda\ \mathbb{1}\{n \neq n'\}}{\left\|\tilde{\boldsymbol{a}}^{\mathrm{f}}_{n,n'}[t]\right\|_2}\right]_+$
    end for
    $\tilde{\boldsymbol{a}}_n[t+1] = \left[\tilde{\boldsymbol{a}}^{\top}_{n,1}[t+1], \ldots, \tilde{\boldsymbol{a}}^{\top}_{n,N}[t+1]\right]^{\top}$
  end for
  Output $\{\tilde{\boldsymbol{a}}_n[t+1]\}_{n=1}^{N}$
end for

which can be thought as the true counterparts of $\hat{\boldsymbol{\Phi}}[t]$ and $\hat{\boldsymbol{r}}_n[t]$.

We consider the following assumptions for the results we present about the JSTIRSO algorithm.

A1. *Bounded samples:* There exists $B_y > 0$ such that $|y_n[t]|^2 \leq B_y$, $|\hat{y}_n[t]|^2 \leq B_y$, and $|\tilde{y}_n[t]|^2 \leq B_y \; \forall \, n, t$.

A2. *Bounded minimum eigenvalue of* $\boldsymbol{\Phi}[t]$ *and* $\hat{\boldsymbol{\Phi}}[t]$*:* There exists $\beta_{\tilde{\ell}} > 0$ such that $\lambda_{\min}(\boldsymbol{\Phi}[t]) \geq \beta_{\tilde{\ell}}$ and $\lambda_{\min}(\hat{\boldsymbol{\Phi}}[t]) \geq \beta_{\tilde{\ell}}, \; \forall \, t \geq P$.

A3. *Bounded maximum eigenvalue of* $\boldsymbol{\Phi}[t]$ *and* $\hat{\boldsymbol{\Phi}}[t]$*:* There exists $L > 0$ such that $\lambda_{\max}(\boldsymbol{\Phi}[t]) \leq L$ and $\lambda_{\max}(\hat{\boldsymbol{\Phi}}[t]) \leq L, \; \forall \, t \geq P$.

A4. *Bounds on the errors in* $\boldsymbol{\Phi}, \boldsymbol{r}_n, \boldsymbol{g}$ *due to noise and missing values:*

$$\left\| \hat{\boldsymbol{g}}[t] - \boldsymbol{g}[t] \right\|_2 \leq B_{\boldsymbol{g}} \quad \forall \; t \tag{E.39}$$

$$\lambda_{\max} \left(\hat{\boldsymbol{\Phi}}[t] - \boldsymbol{\Phi}[t] \right) \leq B_{\boldsymbol{\Phi}} \quad \forall \; t \tag{E.40}$$

$$\left\| \hat{\boldsymbol{r}}_n[t] - \boldsymbol{r}_n[t] \right\|_2 \leq B_{\boldsymbol{r}} \quad \forall \; t. \tag{E.41}$$

The forthcoming results depend on the error int the gradient, given by

$$\boldsymbol{e}^{(n)}[t] = \nabla \tilde{\mathcal{L}}_t^{(n)}(\boldsymbol{a}_n[t]) - \nabla \left[\min_{y_n[t]} \tilde{\ell}_t^{(n)} \left(\{\boldsymbol{y}[\tau]\}_{\tau=0}^{t-1}, y_n[t], \boldsymbol{a}_n[t] \right) \right], \tag{E.42}$$

where $\nabla \tilde{\mathcal{L}}_t^{(n)}(\boldsymbol{a}_n)$ is the gradient defined in (E.37) where the loss function is inexact due to the error in the reconstructed entries in $\hat{\boldsymbol{g}}$ (the error in $\hat{\boldsymbol{g}}$ comes in turn from the missing values and noisy observations), and therefore it is an inexact gradient. On the other hand, the term that is subtracted corresponds [cf. (E.33)] to the contribution from the n-th node to the loss function $\tilde{\mathcal{L}}_t^{(n)}$ when $\hat{y}_n[t]$ is replaced with the true signal $y_n[t]$, and it is therefore the exact gradient.

Dynamic regret analysis is generally expressed in terms of metrics that express how challenging tracking becomes, e.g., how fast the optimal parameters vary. Specifically in our case, the dynamic regret will be expressed in terms of the variation in consecutive optimal solutions (called path length) and the error in the gradient. If we define $\tilde{h}_t^{(n)} \triangleq \tilde{\mathcal{L}}_t^{(n)} + \Omega^{(n)}$, and $\tilde{\boldsymbol{a}}_n^{\circ}[t] \triangleq \arg\min_{\boldsymbol{a}_n} \tilde{h}_t^{(n)}(\boldsymbol{a}_n)$ is a (time-varying) hindsight solution, the path length is given by

$$W^{(n)}[T] \triangleq \sum_{t=P+1}^{T} \left\| \tilde{\boldsymbol{a}}_n^{\circ}[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t-1] \right\|_2 . \tag{E.43}$$

Also, we define the cumulative (norm of the) gradient error as

$$E^{(n)}[T] \triangleq \sum_{t=P}^{T} \left\| \boldsymbol{e}^{(n)}[t] \right\|_2 . \tag{E.44}$$

The dynamic regret for JSTIRSO corresponding to the n-th node is defined as

$$\tilde{R}_d^{(n)}[T] \triangleq \sum_{t=P}^{T} \left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t]) \right], \tag{E.45}$$

where $\tilde{\boldsymbol{a}}_n[t]$ is the JSTIRSO topology estimate. Next, we present two lemmas that will be instrumental to derive the dynamic regret of JSTIRSO.

Lemma 1. *Under assumptions A1 and A3, we have*

$$B_{\boldsymbol{v}} \triangleq \left\|\nabla\tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \leq \frac{\nu}{1+\nu}\left(PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2 + \gamma L\frac{1+\nu}{\nu}\right)\frac{1}{\beta_{\tilde{\ell}}}\left(\frac{\nu}{1+\nu}\sqrt{PN}B_y + \frac{\sqrt{PN}B_y}{1-\gamma}\right) + \left(\frac{\nu}{1+\nu} + \frac{\gamma}{1-\gamma}\right)\sqrt{PN}B_y \quad \text{(E.46)}$$

Proof. To bound $\|\nabla\tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2$, taking the norm on both sides of (E.37) and applying the triangular inequality yields

$$\left\|\nabla\tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \leq U_n[t]\lambda_{\max}\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\right)\|\boldsymbol{a}_n[t]\|_2 + U_n[t]\,\|\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\|_2 + \gamma\lambda_{\max}\left(\hat{\boldsymbol{\Phi}}[t-1]\right)\|\boldsymbol{a}_n[t]\|_2 + \|\gamma\hat{\boldsymbol{r}}_n[t-1]\|_2 \quad \text{(E.47)}$$

Next, using assumptions A1 and A4, it can be easily shown that $\lambda_{\max}(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]) \leq PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2$. Substituting this bound in the above expression and using assumption A3 yields

$$\left\|\nabla\tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \leq U_n[t]\left(PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\|\boldsymbol{a}_n[t]\|_2 + U_n[t]\sqrt{PN}B_y + \gamma L\,\|\boldsymbol{a}_n[t]\|_2 + \|\gamma\hat{\boldsymbol{r}}_n[t-1]\|_2\,. \quad \text{(E.48)}$$

Next, an upper bound of $\hat{\boldsymbol{r}}_n[t-1]$ is derived. By the definition of $\hat{\boldsymbol{r}}_n[t]$ and assumption A1, we have

$$\begin{aligned}
\|\hat{\boldsymbol{r}}_n[t-1]\|_2 &= \left\|\sum_{\tau=P}^{t-1}\gamma^{t-1-\tau}\hat{y}_n[\tau]\,\hat{\boldsymbol{g}}[\tau]\right\|_2 \\
&\leq \frac{1}{\gamma}\left\|\sum_{\tau=P}^{t-1}\gamma^{t-\tau}\sqrt{B_y}\sqrt{B_y}\mathbf{1}_{NP}\right\|_2 \qquad \text{(E.49a)} \\
&= \frac{1}{\gamma}B_y\sqrt{PN}\gamma^t\sum_{\tau=P}^{t-1}\left(\frac{1}{\gamma}\right)^\tau \\
&= \frac{1}{\gamma}B_y\sqrt{PN}\frac{\gamma(1-\gamma^{t-P})}{1-\gamma} \\
&\leq \frac{\sqrt{PN}B_y}{1-\gamma}. \qquad \text{(E.49b)}
\end{aligned}$$

Using the above bound in (E.48)

$$\left\|\nabla\tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \leq U_n[t]\left(PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\|\boldsymbol{a}_n[t]\|_2 + U_n[t]\sqrt{PN}B_y + \gamma L\|\boldsymbol{a}_n[t]\|_2 + \frac{\gamma\sqrt{PN}B_y}{1-\gamma} \tag{E.50}$$

$$\leq \frac{\nu}{1+\nu}\left(PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\|\boldsymbol{a}_n[t]\|_2 + \frac{\nu}{1+\nu}\sqrt{PN}B_y + \gamma L\|\boldsymbol{a}_n[t]\|_2 + \frac{\gamma\sqrt{PN}B_y}{1-\gamma} \tag{E.51}$$

$$= \frac{\nu}{1+\nu}\left(PNB_y + 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2 + \gamma L\frac{1+\nu}{\nu}\right)\|\boldsymbol{a}_n[t]\|_2 + \frac{\nu}{1+\nu}\sqrt{PN}B_y + \frac{\gamma\sqrt{PN}B_y}{1-\gamma}. \tag{E.52}$$

The next step is to derive a bound on $\|\boldsymbol{a}_n[t]\|_2$. To this end, from (E.28) and (E.25), it follows that

$$\|\boldsymbol{a}_n[t+1]\|_2 \tag{E.53}$$

$$\leq \left\|\boldsymbol{a}_n^{\mathrm{f}}[t]\right\|_2$$

$$= \|\boldsymbol{a}_n[t] - \alpha_t\hat{\boldsymbol{v}}_n[t]\|_2$$

$$= \Big\|\boldsymbol{a}_n[t] - \alpha_t\Big(U_n[t]\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t] - U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t] + \hat{\boldsymbol{\Phi}}[t-1]\boldsymbol{a}_n[t] - \hat{\boldsymbol{r}}_n[t-1]\Big)\Big\|_2 \tag{E.54}$$

$$= \Big\|\left(\boldsymbol{I} - \alpha_t\hat{\boldsymbol{\Phi}}[t-1] - \alpha_t U_n[t]\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\right)\boldsymbol{a}_n[t] + \alpha_t U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t] + \alpha_t\hat{\boldsymbol{r}}_n[t-1]\Big\|_2. \tag{E.55}$$

Applying triangular inequality and by assumption A2, we have

$$\|\boldsymbol{a}_n[t+1]\|_2 \tag{E.56}$$

$$\leq \lambda_{\max}\left(\boldsymbol{I} - \alpha_t\hat{\boldsymbol{\Phi}}[t-1] - \alpha_t U_n[t]\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\right)\|\boldsymbol{a}_n[t]\|_2 + \alpha_t\|U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\|_2 + \alpha_t\|\hat{\boldsymbol{r}}_n[t-1]\|_2 \tag{E.57}$$

$$= 1 - \alpha_t\lambda_{\min}\left(\hat{\boldsymbol{\Phi}}[t-1] + \alpha_t U_n[t]\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\right)\|\boldsymbol{a}_n[t]\|_2 + \alpha_t\|U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\|_2 + \alpha_t\|\hat{\boldsymbol{r}}_n[t-1]\|_2 \tag{E.58}$$

$$\leq 1 - \alpha_t\lambda_{\min}\left(\hat{\boldsymbol{\Phi}}[t-1]\right)\|\boldsymbol{a}_n[t]\|_2 + \alpha_t\|U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\|_2 + \alpha_t\|\hat{\boldsymbol{r}}_n[t-1]\|_2 \tag{E.59}$$

$$\leq (1 - \alpha_t\beta_{\tilde{\ell}})\|\boldsymbol{a}_n[t]\|_2 + \alpha_t\|U_n[t]\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\|_2 + \alpha_t\|\hat{\boldsymbol{r}}_n[t-1]\|_2. \tag{E.60}$$

Substituting the bound on $\|\hat{\boldsymbol{r}}_n[t-1]\|_2$ from (E.49b) into the above expression, we have

$$\|\boldsymbol{a}_n[t+1]\|_2 \leq (1 - \alpha_t\beta_{\tilde{\ell}})\|\boldsymbol{a}_n[t]\|_2 + \alpha_t\Big(U_n[t]\sqrt{PN}B_y + \frac{\sqrt{PN}B_y}{1-\gamma}\Big). \tag{E.61}$$

Setting $\alpha_t = \alpha$ and for $0 < \alpha \leq 1/L$, it can be proven by recursively substituting into (E.60) (similar steps to those in the proof of [45, Theorem 5]), that

$$\|\boldsymbol{a}_n[t+1]\|_2 \leq \frac{1}{\beta_{\tilde{\ell}}} \left(\frac{\nu}{1+\nu} \sqrt{PN} B_y + \frac{\sqrt{PN} B_y}{1-\gamma} \right) \quad \forall t. \tag{E.62}$$

Substituting the above bound into (E.52) completes the proof. □

Lemma 2. *All the subgradients of the regularization function $\Omega^{(n)}$ are bounded by $\lambda\sqrt{N}$, i.e., $\|\boldsymbol{u}_t\|_2 \leq \lambda\sqrt{N}$, where $\boldsymbol{u}_t \in \partial\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t])$.*

Proof. To find an upper bound on $\|\boldsymbol{u}_t\|_2$, we apply the result in [46, Lemma 2.6] to $\Omega^{(n)}$, which establishes that all the subgradients of $\Omega^{(n)}$ are bounded by its Lipschitz continuity parameter $L_{\Omega^{(n)}}$. In the following, we show that $L_{\Omega^{(n)}} = \lambda\sqrt{N}$. Lipschitz continuity of $\Omega^{(n)}$ means that there exists $L_{\Omega^{(n)}}$ such that

$$\left|\Omega^{(n)}(\boldsymbol{a}) - \Omega^{(n)}(\boldsymbol{b})\right| \leq L_{\Omega^{(n)}} \|\boldsymbol{a} - \boldsymbol{b}\|_2 , \tag{E.63}$$

$\forall\, \boldsymbol{a}, \boldsymbol{b}$. By definition, we have $\Omega^{(n)}(\boldsymbol{x}_n) = \lambda \sum_{n'=1, n' \neq n}^{N} \|\boldsymbol{x}_{n,n'}\|_2$ with $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^\top, ..., \boldsymbol{x}_{n,N}^\top]^\top$, $\boldsymbol{x}_{n,n'} \in \mathbb{R}^P, n' = 1, ..., N$. Let $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^\top, ..., \boldsymbol{z}_{n,N}^\top]^\top, \boldsymbol{z}_{n,n'} \in \mathbb{R}^P, n' = 1, ..., N$ and by taking the l.h.s. of (E.63), we have

$$\begin{aligned}
\left|\Omega^{(n)}(\boldsymbol{x}_n) - \Omega^{(n)}(\boldsymbol{z}_n)\right| &= \lambda \left| \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{x}_{n,n'}\|_2 - \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{z}_{n,n'}\|_2 \right| \\
&= \lambda \left| \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \left[\|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{z}_{n,n'}\|_2 \right] \right| \\
&\leq \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \left| \|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{z}_{n,n'}\|_2 \right| \qquad \text{(E.64a)} \\
&\leq \lambda \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \|\boldsymbol{x}_{n,n'} - \boldsymbol{z}_{n,n'}\|_2 \qquad \text{(E.64b)} \\
&\leq \lambda \sum_{n'=1}^{N} \|\boldsymbol{x}_{n,n'} - \boldsymbol{z}_{n,n'}\|_2 \\
&\leq \lambda\sqrt{N} \|\boldsymbol{x}_n - \boldsymbol{z}_n\|_2 , \qquad \text{(E.64c)}
\end{aligned}$$

where the inequality in (E.64a) holds due to the triangle inequality for scalars ($\|\boldsymbol{x}_{n,n'}\|_2 - \|\boldsymbol{y}_{n,n'}\|_2$ as scalars); (E.64b) holds due to the reverse triangle inequality (given by $|\|\boldsymbol{x}_1\|_2 - \|\boldsymbol{x}_2\|_2| \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$); and (E.64c) follows from the inequality $\|\boldsymbol{b}\|_1 \leq \sqrt{N}\|\boldsymbol{b}\|_2$ with $\boldsymbol{b} \in \mathbb{R}^N$ [118, Sec. 2.2.2]. The inequality in (E.64c) implies that (E.63) is satisfied with $L_{\Omega^{(n)}} = \lambda\sqrt{N}$, i.e., $\Omega^{(n)}$ is $\lambda\sqrt{N}$-Lipschitz continuous. □

Next, we present a bound on the dynamic regret of JSTIRSO.

Theorem 7. *Under assumptions A1, A2, and A3, let* $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^{T}$ *be generated by JSTIRSO (**Algorithm 2**) with a constant step size* $\alpha \in (0, 1/L]$. *If there exists* σ *such that*

$$\|\tilde{\boldsymbol{a}}_n^{\circ}[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t-1]\|_2 \leq \sigma, \ \forall\, t \geq P+1, \tag{E.65}$$

then the dynamic regret of JSTIRSO satisfies:

$$\tilde{R}_d^{(n)}[T] \leq \frac{1}{\alpha\beta_{\tilde{\ell}}}\Big[B_{\boldsymbol{v}} + \lambda\sqrt{N}\Big]\left(\|\tilde{\boldsymbol{a}}_n[P] - \tilde{\boldsymbol{a}}_n^{\circ}[P]\|_2 + W^{(n)}[T] + \alpha E^{(n)}[T]\right), \tag{E.66}$$

where $B_{\boldsymbol{v}}$ *is defined in* (E.46).

Proof. We derive the dynamic regret of JSTIRSO. To this end, since $\tilde{h}_t$ is convex, we have by definition

$$\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t]) \geq \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \left(\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right)^{\top}(\tilde{\boldsymbol{a}}_n^{\circ}[t] - \tilde{\boldsymbol{a}}_n[t]), \tag{E.67}$$

$\forall\, \tilde{\boldsymbol{a}}_n^{\circ}[t], \tilde{\boldsymbol{a}}_n[t]$, where a subgradient of $\tilde{h}_t^{(n)}$ is given by $\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) = \nabla \tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) + \boldsymbol{u}_t$ with $\boldsymbol{u}_t \in \partial\Omega^{(n)}(\tilde{\boldsymbol{a}}_n[t])$. Rearranging (E.67) and summing both sides of the inequality from $t = P$ to T results in:

$$\sum_{t=P}^{T}\left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t])\right] \leq \sum_{t=P}^{T}\left(\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right)^{\top} \cdot (\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]). \tag{E.68}$$

By applying the Cauchy-Schwarz inequality on each term of the summation in the r.h.s. of the above inequality, we obtain

$$\sum_{t=P}^{T}\left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t])\right] \leq \sum_{t=P}^{T}\left\|\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\right\|_2 \cdot \|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]\|_2. \tag{E.69}$$

The next step is to derive an upper bound on $\|\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2$. From the definition of $\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])$ and by the triangular inequality, we have

$$\|\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 \leq \|\nabla \tilde{\mathcal{L}}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 + \|\boldsymbol{u}_t\|_2. \tag{E.70}$$

From Lemma 1 and Lemma 2, we have $\|\nabla^s \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])\|_2 \leq B_{\boldsymbol{v}} + \lambda\sqrt{N}$. Substituting this bound into (E.69) leads to:

$$\sum_{t=P}^{T}\left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t]) - \tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^{\circ}[t])\right] \leq \sum_{t=P}^{T}\Big[B_{\boldsymbol{v}} + \lambda\sqrt{N}\Big]\,\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]\|_2. \tag{E.71}$$

Note that $U_n[t]$ is upper-bounded by $\nu/(1+\nu)$. Next, we apply Lemma 2 in [70] in order to bound $\sum_{t=P}^{T}\|\tilde{\boldsymbol{a}}_n[t] - \tilde{\boldsymbol{a}}_n^{\circ}[t]\|_2$ in (E.71). The hypotheses of Lemma 2 are Lipschitz smoothness of $\tilde{\mathcal{L}}_t^{(n)}$, Lipschitz continuity of $\Omega^{(n)}$, and strong convexity of $\tilde{\mathcal{L}}_t^{(n)}$. Lipschitz continuity of $\Omega^{(n)}$ is proved in (E.64c) whereas strong convexity of $\tilde{\mathcal{L}}_t^{(n)}$ is implied by the assumption A2. So we need to verify that $\tilde{\mathcal{L}}_t^{(n)}$ is Lipschitz-smooth. Since $\tilde{\mathcal{L}}_t^{(n)}$ is twice-differentiable, assumption A3 is equivalent to saying that $\tilde{\mathcal{L}}_t^{(n)}$ is L-Lipschitz smooth. To apply Lemma 2 in [70], one can set K in [70] as $T - P + 1$, g_k as $\Omega^{(n)}$, and f_k as $\tilde{\mathcal{L}}_{P+k-1}^{(n)}$,

it follows that $\boldsymbol{x}_k$ in [70] equals $\tilde{\boldsymbol{a}}_n[P+k-1]$ and $\boldsymbol{x}_k^\circ$ equals $\tilde{\boldsymbol{a}}_n^\circ[P+k-1]$. Then, since we have already shown above that the hypotheses of Lemma 2 in [70] hold in our case, applying it to bound $\|\tilde{\boldsymbol{a}}_n[t]-\tilde{\boldsymbol{a}}_n^\circ[t]\|_2$ in (E.71) yields:

$$\sum_{t=P}^{T}\left[\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n[t])-\tilde{h}_t^{(n)}(\tilde{\boldsymbol{a}}_n^\circ[t])\right]\leq \\ \frac{1}{\alpha\beta_{\tilde{\ell}}}\left[B_{\boldsymbol{v}}+\lambda\sqrt{N}\right]\left(\|\tilde{\boldsymbol{a}}_n[P]-\tilde{\boldsymbol{a}}_n^\circ[P]\|_2+W^{(n)}[T]+\alpha E^{(n)}[T]\right). \tag{E.72}$$

This concludes the proof (note that initializing $\tilde{\boldsymbol{a}}_n[P]=\mathbf{0}_{NP}$ can lead to further simplification). □

The bound that has been presented depends on the cumulative error $E^{(n)}[T]$, which can be bounded as a function of the quantities introduced in A4 (related to the inexactness of the reconstructed samples). The following lemma provides a bound on $\left\|\boldsymbol{e}^{(n)}[t]\right\|$.

Lemma 3. *Under assumptions A1 and A4, let $\{\tilde{\boldsymbol{a}}_n[t]\}_{t=P}^T$ be generated by JSTIRSO (**Algorithm 2**) with a constant step size $\alpha\in(0,1/L]$. Then, the error associated with the inexact gradient [cf. (E.42)] is bounded in norm as*

$$\left\|\boldsymbol{e}^{(n)}[t]\right\|_2\leq\left(\gamma B_{\boldsymbol{\Phi}}+\left(\frac{\nu}{1+\nu}\right)\left(2\sqrt{PNB_y}B_{\boldsymbol{g}}+B_{\boldsymbol{g}}^2\right)\right) \\ \times\frac{\sqrt{PN}B_y}{\beta_{\tilde{\ell}}}\left(\frac{\nu}{1+\nu}+\frac{1}{1-\gamma}\right)+\gamma B_{\boldsymbol{r}}+\left(\frac{\nu}{1+\nu}\right)B_{\boldsymbol{g}}\sqrt{B_y}. \tag{E.73}$$

Proof. Next, we analyze the error in the gradient for JSTIRSO. This error can be proved to be bounded under certain assumptions. The error in the gradient is given by (E.42) and can be written as:

$$\begin{aligned}&\boldsymbol{e}^{(n)}[t]\\&=U_n[t]\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]\boldsymbol{a}_n[t]-\tilde{y}_n[t]\hat{\boldsymbol{g}}[t]\right)+\gamma\hat{\boldsymbol{\Phi}}[t-1]\boldsymbol{a}_n[t]\\&\quad-\gamma\hat{\boldsymbol{r}}_n[t-1]-U_n[t]\left(\boldsymbol{g}[t]\boldsymbol{g}^\top[t]\boldsymbol{a}_n[t]-\tilde{y}_n[t]\boldsymbol{g}[t]\right)\\&\quad-\gamma\boldsymbol{\Phi}[t-1]\boldsymbol{a}_n[t]+\gamma\boldsymbol{r}_n[t-1] &\text{(E.74a)}\\&=\gamma(\hat{\boldsymbol{\Phi}}[t-1]-\boldsymbol{\Phi}[t-1])\boldsymbol{a}_n[t]+U_n[t](\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]-\boldsymbol{g}[t]\boldsymbol{g}^\top[t])\boldsymbol{a}_n[t]\\&\quad+\gamma(\boldsymbol{r}_n[t-1]-\hat{\boldsymbol{r}}_n[t-1])+U_n[t]\tilde{y}_n[t](\boldsymbol{g}[t]-\hat{\boldsymbol{g}}[t]). &\text{(E.74b)}\end{aligned}$$

Next, we take the norm on both sides of the above equation

$$\begin{aligned}&\left\|\boldsymbol{e}^{(n)}[t]\right\|_2\\&\leq\left\|\gamma(\hat{\boldsymbol{\Phi}}[t-1]-\boldsymbol{\Phi}[t-1])\boldsymbol{a}_n[t]\right\|_2+\left\|U_n[t](\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]-\boldsymbol{g}[t]\boldsymbol{g}^\top[t])\boldsymbol{a}_n[t]\right\|_2\\&\quad+\|\gamma(\boldsymbol{r}_n[t-1]-\hat{\boldsymbol{r}}_n[t-1])\|_2+\|U_n[t]y_n[t](\boldsymbol{g}[t]-\hat{\boldsymbol{g}}[t])\|_2 &\text{(E.75a)}\\&\leq\gamma\lambda_{\max}\left(\hat{\boldsymbol{\Phi}}[t-1]-\boldsymbol{\Phi}[t-1]\right)\|\boldsymbol{a}_n[t]\|_2+U_n[t]\lambda_{\max}\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^\top[t]-\boldsymbol{g}[t]\boldsymbol{g}^\top[t]\right)\|\boldsymbol{a}_n[t]\|_2\\&\quad+\gamma B_{\boldsymbol{r}}+U_n[t]\,|\tilde{y}_n[t]|\,\|\boldsymbol{g}[t]-\hat{\boldsymbol{g}}[t]\|_2\,, &\text{(E.75b)}\end{aligned}$$

where the first inequality holds because of the triangular inequality and the second inequality holds because of Cauchy-Schwarz inequality. The next step is to use the bounds defined in assumptions A1 and A4. Combining A1 and (E.39) it can be proven that

$$\lambda_{\max}\left(\hat{\boldsymbol{g}}[t]\hat{\boldsymbol{g}}^{\top}[t]-\boldsymbol{g}[t]\boldsymbol{g}^{\top}[t]\right) \leq 2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2. \tag{E.76}$$

Therefore substituting (E.40), (E.76), (E.41), and (E.39) into (E.75b), we obtain

$$\begin{aligned}
&\left\|\boldsymbol{e}^{(n)}[t]\right\|_2 \\
&\leq \gamma B_{\boldsymbol{\Phi}} \left\|\boldsymbol{a}_n[t]\right\|_2 + U_n[t]\left(2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\left\|\boldsymbol{a}_n[t]\right\|_2 + \gamma B_{\boldsymbol{r}} + U_n[t]B_{\boldsymbol{g}}\sqrt{B_y} && \text{(E.77a)}\\
&\leq \gamma B_{\boldsymbol{\Phi}} \left\|\boldsymbol{a}_n[t]\right\|_2 + \left(\frac{\nu}{1+\nu}\right)\left(2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\left\|\boldsymbol{a}_n[t]\right\|_2 + \gamma B_{\boldsymbol{r}} + \left(\frac{\nu}{1+\nu}\right)B_{\boldsymbol{g}}\sqrt{B_y} && \text{(E.77b)}\\
&= \left(\gamma B_{\boldsymbol{\Phi}} + \left(\frac{\nu}{1+\nu}\right)\left(2\sqrt{PNB_y}B_{\boldsymbol{g}} + B_{\boldsymbol{g}}^2\right)\right)\left\|\boldsymbol{a}_n[t]\right\|_2 + \gamma B_{\boldsymbol{r}} + \left(\frac{\nu}{1+\nu}\right)B_{\boldsymbol{g}}\sqrt{B_y}, && \text{(E.77c)}
\end{aligned}$$

where the final result comes from substituting an upper bound on $U_n[t]$ and rearranging terms. We can use here the same bound on $\|\boldsymbol{a}_n[t]\|_2$ that was derived in the proof of Lemma 1 [cf. (E.62)]:

$$\left\|\boldsymbol{a}_n[t+1]\right\|_2 \leq \frac{\sqrt{PN}B_y}{\beta_{\tilde{\ell}}}\left(\frac{\nu}{1+\nu}+\frac{1}{1-\gamma}\right) \quad \forall\ t; \tag{E.78}$$

substituting the above bound into (E.77c) completes the proof. □

It can be observed that this bound depends on the parameters of the data, the parameters ν, and γ in the estimation algorithm. The bound on $\|\boldsymbol{e}^{(n)}[t]\|_2$ means that under certain assumptions, the error in the gradient is always bounded.

Remark. Since $\|\boldsymbol{e}^{(n)}[t]\|_2$ and $E^{(n)}[T]$ are related via (E.44), the above bound can be used to replace $E^{(n)}[T]$ in the regret bound in (E.66) with an expression that depends on the quantities expressed in A4.

The bound on the dynamic regret for JSTIRSO depends on $W^{(n)}[T]$ and $E^{(n)}[T]$. If both the path length $W^{(n)}[T]$ and the cumulative error $E^{(n)}[T]$ are sublinear, then the dynamic regret bound becomes sublinear.

E.7 Numerical Tests

We analyze the performance of proposed algorithm by presenting normalized mean squared deviation for both the signal and the topology. The NMSD for the signal is given by

$$\mathrm{NMSD}_s[t] = \frac{\mathbb{E}\left[\,\|\boldsymbol{y}[t]-\hat{\boldsymbol{y}}[t]\|_2^2\,\right]}{\mathbb{E}\left[\|\boldsymbol{y}[t]\|_2^2\right]}, \tag{E.79}$$

where $\boldsymbol{y}[t]$ is the true signal while $\hat{\boldsymbol{y}}[t]$ is the estimated signal from the noisy observations with missing values. The NMSD for the topology is defined as:

$$\mathrm{NMSD}_g[t] \triangleq \frac{\mathbb{E}\left[\sum_{n=1}^{N}\|\hat{\boldsymbol{a}}_n[t]-\boldsymbol{a}_n^{\mathrm{true}}[t]\|_2^2\right]}{\mathbb{E}\left[\sum_{n=1}^{N}\|\boldsymbol{a}_n^{\mathrm{true}}[t]\|_2^2\right]}, \tag{E.80}$$

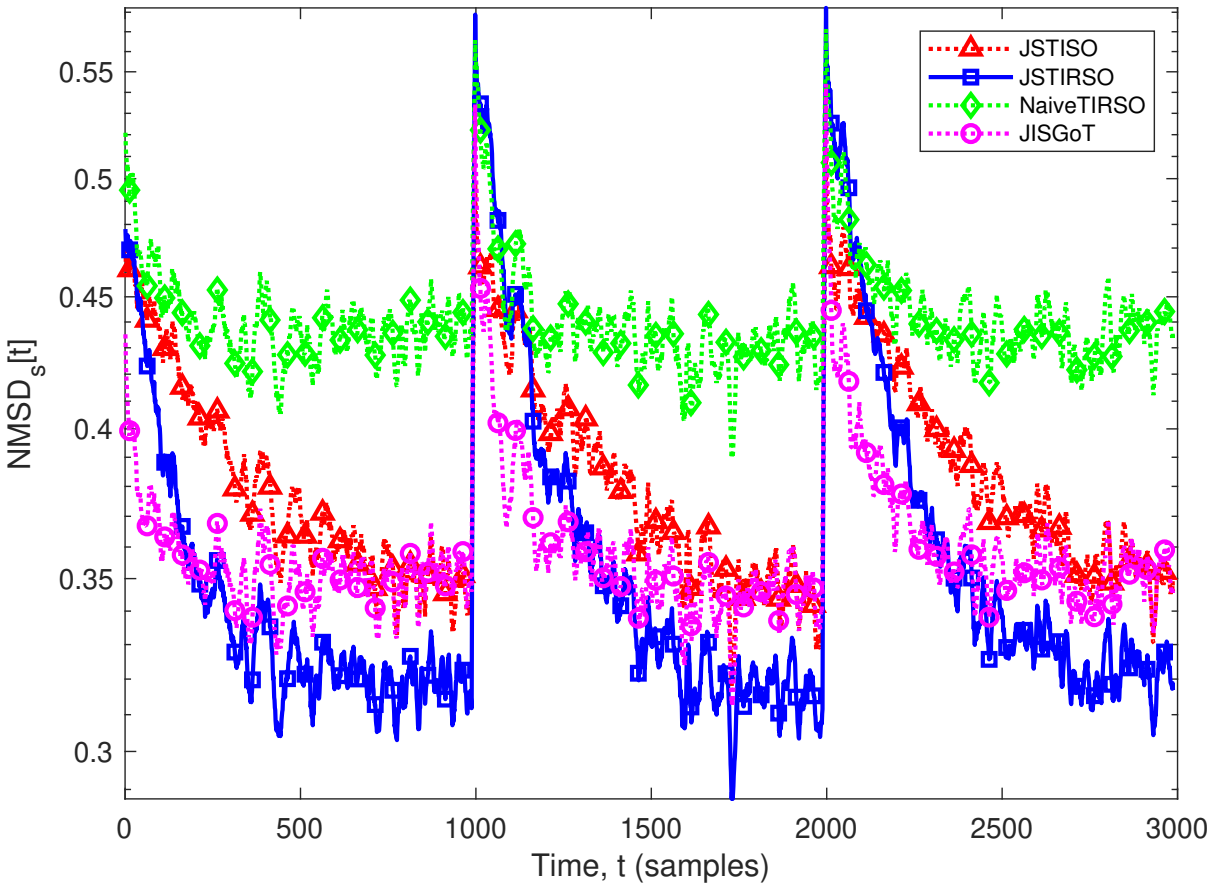


Figure E.1: NMSD for signal estimation vs. time. Simulation parameters: $N = 10, P = 3, T = 3000, \sigma_u = 0.01, \sigma_\epsilon = 0.01, \gamma = 0.99, \rho = 0.75, p_e = 0.25, \alpha = 0.1/L$, no. of Monte Carlo iterations = 500, JISGoT iterations = 10.

which measures the difference between the estimates $\{\hat{\boldsymbol{a}}_n[t]\}_t$ and the time-varying true VAR coefficients $\{\boldsymbol{a}_n^{\text{true}}[t]\}_t$. The letter g in NMSD_g stands for the graph.

We consider a dynamic VAR model, where the coefficients change abruptly. There are two time instants where the VAR coefficients are changed. To generate the synthetic data, an Erdős-Rényi random graph is generated with edge probability p_e and self-loop probability 1. This gives us a binary adjacency matrix of the underlying graph. This graph determines which entries of the matrices $\{\boldsymbol{A}_p\}_{p=0}^P$ are zero. The rest of entries are drawn i.i.d. from a standard normal distribution. Matrices $\{\boldsymbol{A}_p\}_{p=0}^P$ are scaled down afterwards by a constant that ensures that the VAR process is stable [43]. The innovation process samples are drawn independently as $\boldsymbol{u}[t] \sim \mathcal{N}(\boldsymbol{0}, \sigma_u^2 \boldsymbol{I}_N)$. At $t = T/3$ and $t = 2T/3$, the model changes abruptly from one model to another model. This is performed by generating a new set of VAR coefficients while keeping the binary adjacency matrix fixed.

The performance of JSTISO (Algorithm 13) and JSTIRSO (Algorithm 14) is evaluated for the signal and the topology estimation. The values for the parameter ν in JSTISO and JSTIRSO is selected via a grid search method where the optimal values of the parameter is selected based on minimum squared deviation for the signal. The values of the regularization parameters for the proposed algorithms are also selected via grid search

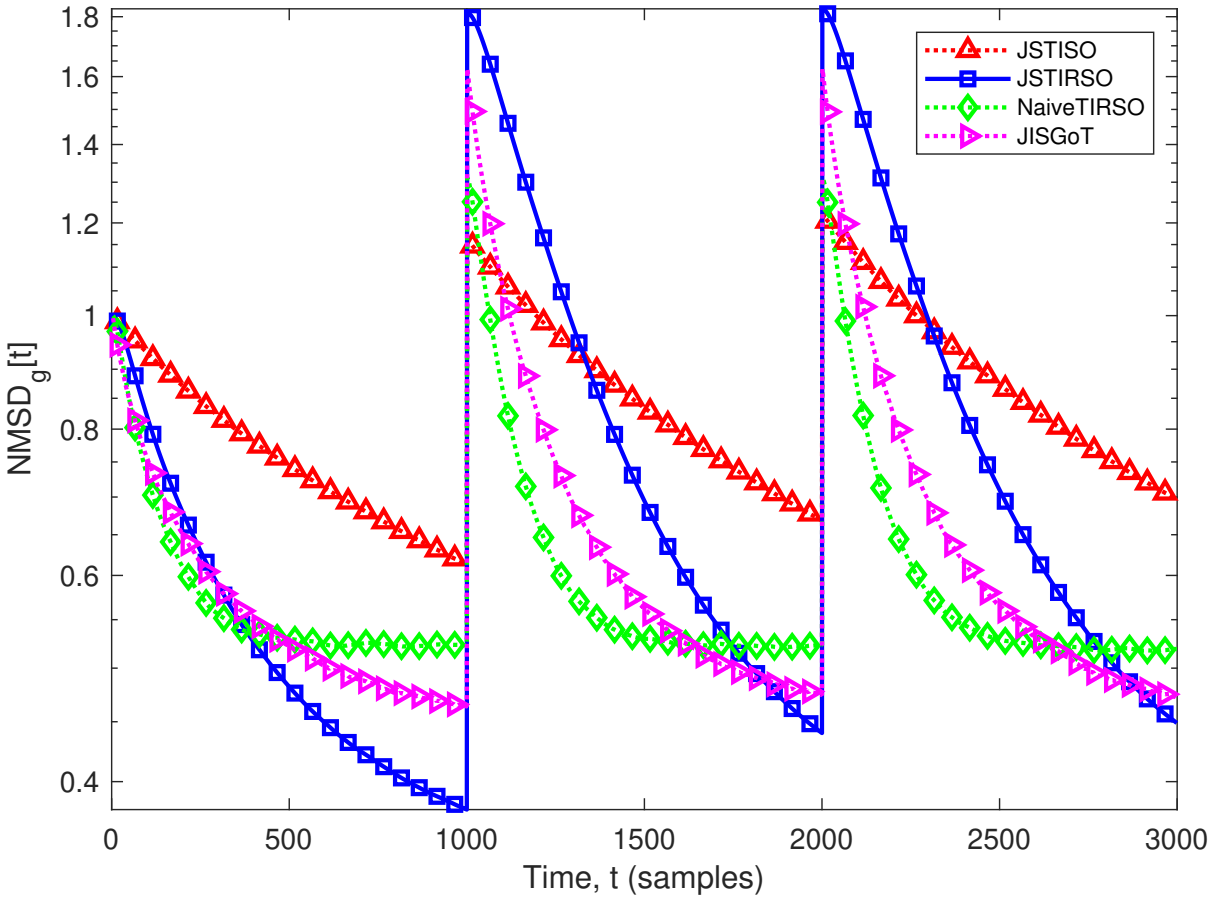


Figure E.2: NMSD for topology estimation vs. time. Simulation parameters: $N = 10, P = 3, T = 3000, \sigma_u = 0.01, \sigma_\epsilon = 0.01, \gamma = 0.99, \rho = 0.75, p_e = 0.25, \alpha = 0.1/L$, no. of Monte Carlo iterations = 500, JISGoT iterations = 10.

for true data (without noise and missing values). In Fig. E.1, the NMSD for the signal estimation is presented for JSTISO and JSTIRSO. The signal is estimated from noisy observations with missing data via (E.22). The estimated signal is different for both the algorithm because (E.22) depends on the estimated coefficients. Hence, the corresponding estimated coefficients from JSTISO and JSTIRSO are used in the signal estimation. We compare our proposed algorithms with JISGoT [88, Algorithm 4]. Despite refining the the previous P signal estimates at the cost of high computational complexity and running more than one iteration at each time instant, JSTIRSO eventually achieves lower error than JISGoT. Moreover, the performance of a third algorithm named as 'NaiveTIRSO' is also presented. NaiveTIRSO is an extension of the TIRSO presented in [45]. To deal with missing values, the prediction via the VAR process is used in NaiveTIRSO. The result in Fig. E.1 show that JSTISO and JSTIRSO can estimate the signal from the noisy observation having missing values. It can be observed that at the time when the model changes, the error starts to be increasing. This is due to the fact that the estimated coefficients at that time instant have just changed and the topology estimate by the algorithm is not accurate. Moreover, note that JSTIRSO outperforms JSTISO and NaiveTIRSO for the signal estimation.

The NMSD corresponding to the topology for the proposed algorithms is presented in Fig. E.2. JSTIRSO tracks the time-varying topologies with a lower final NMSD than JSTISO, JISGoT, and NaiveTIRSO. The rationale is the special loss function plus jointly estimating the signal from the observations with noise and missing values for JSTIRSO. Moreover, JSTIRSO is also supported by theoretical guarantees for the time-varying scenarios.

E.8 Conclusions

The problem of tracking time-varying topologies from noisy observations in the presence of missing data is investigated. Initially, the batch problem for the missing values is presented. Due to tractability issues of the batch problem, an approximated problem is solved in the online scenario to track the variations in the topologies with missing data. Two online algorithms JSTISO and JSTIRSO are proposed, where the problem is solved by deriving a joint approach for the estimation of the signal from the noisy data and estimation of the topology. To evaluate the performance of JSTIRSO, a dynamic regret bound is derived. Numerical results showcase the tracking capabilities of the proposed algorithms.

Bibliography

[1] A. Natali, E. Isufi, and G. Leus, "Forecasting multi-dimensional processes over graphs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 5575–5579.

[2] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised spatiotemporal graphical modeling approach to anomaly detection in distributed CPS," in *ACM/IEEE Int. Conf. Cyber-Physical Syst.*, Apr. 2016, pp. 1–10.

[3] P. D. Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. Signal Info. Process. Netw.*, vol. 2, no. 4, pp. 555–568, Dec. 2016.

[4] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[5] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[6] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1085–1089.

[7] A. Gavili and X. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303–6318, 2017.

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

[9] Y. Shen, P. A. Traganitis, and G. B. Giannakis, "Nonlinear dimensionality reduction on graphs," in *Proc. IEEE Int. Workshop Comput. Advan. Multi-Sensor Adapt. Process.*, Curacao, Netherlands Antilles, Dec. 2017.

[10] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust pca on graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 740–756, 2016.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006. [Online]. Available: https://books.google.com/books?id=kTNoQgAACAAJ

[12] E. Dall'Anese, A. Simonetto, S. Becker, and L. Madden, "Optimization and learning with information streams: Time-varying algorithms and applications," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 71–83, 2020.

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

[14] D. Angelosante and G. B. Giannakis, "Sparse graphical modeling of piecewise-stationary time series," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, 2011, pp. 1960–1963.

[15] S. L. Lauritzen, *Graphical Models*. Clarendon Press, 1996, vol. 17.

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[17] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.

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

[19] 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.

[20] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

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

[22] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *arXiv preprint arXiv:1810.13066*, 2018.

[23] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2399–2413, May 2018.

[24] R. B. Kline, *Principles and Practice of Structural Equation Modeling.* Guilford Publications, 2015.

[25] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural equation models for network topology inference," in *Proc. Annu. Conf. Inform. Sci. Syst.*, Princeton, NJ, 2016, pp. 163–168.

[26] ——, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3675–3687, Jul. 2017.

[27] J. Pearl, *Causality: Models, Reasoning and Inference.* Cambridge university press, 2009.

[28] C. W. J. Granger, "Some recent development in a concept of causality," *J. Econometrics*, vol. 39, no. 1-2, pp. 199–211, Sep. 1988.

[29] J. Songsiri and L. Vandenberghe, "Topology selection in graphical models of autoregressive processes," *J. Mach. Learn. Res.*, vol. 11, pp. 2671–2705, Oct. 2010.

[30] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2189–2199, Aug. 2004.

[31] S. Basu, A. Shojaie, and G. Michailidis, "Network Granger causality with inherent grouping structure." *J. Mach. Learn. Res.*, vol. 16, no. 2, pp. 417–453, Mar. 2015.

[32] R. Shafipour, A. Hashemi, G. Mateos, and H. Vikalo, "Online topology inference from streaming stationary graph signals," in *IEEE Data Sci. Workshop*, Jun. 2019, pp. 140–144.

[33] R. Shafipour and G. Mateos, "Online network topology inference with partial connectivity informatio," in *Proc. IEEE Int. Workshop Comput. Advan. Multi-Sensor Adapt. Process.*, 2019, pp. 226–230.

[34] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 205–213.

[35] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 563–575, Aug. 2014.

[36] M. G. Rodriguez, J. Leskovec, and B. Schölkopf, "Structure and dynamics of information pathways in online media," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2013, pp. 23–32.

[37] B. Zaman, L. M. Lopez-Ramos, and B. Beferull-Lozano, "Dynamic regret analysis for online tracking of time-varying structural equation model topologies," *arXiv preprint arXiv:2003.08145*, 2020.

[38] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, "Estimating time-varying networks," *Ann. Appl. Statist*, pp. 94–123, 2010.

[39] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[40] B. Baingana and G. B. Giannakis, "Tracking switched dynamic network topologies from information cascades," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 985–997, 2017.

[41] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning with constraints on graph temporal variation," 2020.

[42] A. Tank, E. B. Fox, and A. Shojaie, "An efficient admm algorithm for structural break detection in multivariate time series," *arXiv preprint arXiv:1711.08392*, 2017.

[43] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Springer, 2005.

[44] B. Zaman, L. M. López-Ramos, D. Romero, and B. Beferull-Lozano, "Online topology estimation for vector autoregressive processes in data networks," in *Proc. IEEE Int. Workshop Comput. Advan. Multi-Sensor Adapt. Process.*, Curaçao, Dutch Antilles, Dec. 2017.

[45] B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *Submitted to IEEE Trans. Signal Process., arXiv preprint arXiv:1904.01864*, Apr. 2019.

[46] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.

[47] L. M. Lopez-Ramos, D. Romero, B. Zaman, and B. Beferull-Lozano, "Dynamic network identification from non-stationary vector autoregressive time series," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2018, pp. 773–777.

[48] B. Zaman, L. M. Lopez-Ramos, and B. Beferull-Lozano, "Online joint topology identification and signal estimation with inexact proximal online gradient descent," *submitted to IEEE Trans. Signal Process.*, 2020.

[49] P. Geiger, K. Zhang, B. Schoelkopf, M. Gong, and D. Janzing, "Causal inference by identification of vector autoregressive processes with hidden components," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1917–1925.

[50] A. Roebroeck, E. Formisano, and R. Goebel, "Mapping directed influence over the brain using granger causality and fmri," *Neuroimage*, vol. 25, no. 1, pp. 230–242, 2005.

[51] M. Besserve, B. Schölkopf, N. K. Logothetis, and S. Panzeri, "Causal relationships between frequency bands of extracellular signals in visual cortex revealed by an information theoretic analysis," *Journal of computational neuroscience*, vol. 29, no. 3, pp. 547–566, 2010.

[52] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Computational Biology*, vol. 9, no. 5, 2013.

[53] J. Englin, D. Lambert, and W. D. Shaw, "A structural equations approach to modeling consumptive recreation demand," *Journal of Environmental Economics and Management*, vol. 33, no. 1, pp. 33–43, 1997.

[54] E. Hazan, "Introduction to online convex optimization," *Found. Trends Mach. Learn.*, vol. 2, no. 3-4, pp. 157–325, 2016.

[55] F. Orabona, "A modern introduction to online learning," *arXiv preprint arXiv:1912.13213*, 2019.

[56] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[57] A. Mokhtari, S. Shahrampour, A.Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," *arXiv preprint arXiv:1603.04954*, 2016.

[58] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online Optimization : Competing with Dynamic Comparators," in *Proc. Mach. Learn. Res.*, San Diego, CA, May 2015, pp. 398–406.

[59] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.

[60] A. Zellner, "Causality and econometrics," in *Carnegie-Rochester Conference series on Public Policy*, vol. 10. Elsevier, 1979, pp. 9–54.

[61] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory*. Prentice-Hall, 1993.

[62] H. Lütkepohl, M. Krätzig, and P. C. Phillips, *Applied Time Series Econometrics*. Cambridge University Press, 2004.

[63] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira, "Modeling gene expression regulatory networks with the sparse vector autoregressive model," *BMC Syst. Bio.*, vol. 1, no. 1, p. 39, 2007.

[64] P. A. Valdés-Sosa, J. M. Sánchez-Bornot, A. Lage-Castellanos, M. Vega-Hernández, J. Bosch-Bayard, L. Melie-García, and E. Canales-Rodríguez, "Estimating brain functional connectivity with sparse multivariate autoregression," *Philosoph. Trans. Royal Soc. London B: Bio. Sci.*, vol. 360, no. 1457, pp. 969–981, 2005.

[65] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*. Springer Berlin, 1971, vol. 170.

[66] A. Bolstad, B. D. V. Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2628–2641, Jun. 2011.

[67] J. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent," in *Proc. of Annu. Conf. Computat. Learn. Theory*, 2010, pp. 14–26.

[68] J. A. Bazerque, B. Baingana, and G. B. Giannakis, "Identifiability of sparse structural equation models for directed and cyclic networks," in *IEEE Global Conference on Signal and Information Processing*, 2013, pp. 839–842.

[69] T. Asparouhov, E. L. Hamaker, and B. Muthén, "Dynamic structural equation models," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 25, no. 3, pp. 359–388, 2018.

[70] R. Dixit, A. S. Bedi, R. Tripathi, and K. Rajawat, "Online learning with inexact proximal online gradient descent algorithms," *IEEE Trans. Signal Process.*, vol. 67, no. 5, pp. 1338–1352, Mar. 2019.

[71] J. R. Sato, P. A. Morettin, P. R. Arantes, and E. Amaro Jr, "Wavelet based time-varying vector autoregressive modelling," *Computational Statistics & Data Analysis*, vol. 51, no. 12, pp. 5847–5866, 2007.

[72] R. Dahlhaus, "Locally stationary processes," in *Handbook of statistics*. Elsevier, 2012, vol. 30, pp. 351–413.

[73] M. Niedźwiecki, M. Ciołek, and Y. Kajikawa, "On adaptive covariance and spectrum estimation of locally stationary multivariate processes," *Automatica*, vol. 82, pp. 1–12, 2017.

[74] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Bayesian nonparametric inference of switching dynamic linear models," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1569–1585, April 2011.

[75] H. Ombao, R. Von Sachs, and W. Guo, "Slex analysis of multivariate nonstationary time series," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 519–531, 2005.

[76] A. Aue, S. Hörmann, L. Horváth, M. Reimherr *et al.*, "Break detection in the covariance structure of multivariate time series models," *The Annals of Statistics*, vol. 37, no. 6B, pp. 4046–4087, 2009.

[77] H. Cho and P. Fryzlewicz, "Multiple-change-point detection for high dimensional time series via sparsified binary segmentation," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 77, no. 2, pp. 475–507, 2015.

[78] H. Cho *et al.*, "Change-point detection in panel data via double cusum statistic," *Electronic Journal of Statistics*, vol. 10, no. 2, pp. 2000–2038, 2016.

[79] A. Safikhani and A. Shojaie, "Structural break detection in high-dimensional non-stationary var models," *arXiv preprint arXiv:1708.02736*, 2017.

[80] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of arx-models using sum-of-norms regularization," *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[81] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. USA: John Wiley & Sons, Inc., 2014.

[82] E. Pavez and A. Ortega, "Covariance matrix estimation with non uniform and data dependent missing observations," *arXiv preprint arXiv:1910.00667*, 2019.

[83] P. Berger, G. Hannak, and G. Matz, "Efficient graph learning from noisy and incomplete data," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 105–119, 2020.

[84] M. Rao, T. Javidi, Y. C. Eldar, and A. Goldsmith, "Estimation in autoregressive processes with partial observations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4212–4216.

[85] P.-L. Loh and M. J. Wainwright, "High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity," *The Annals of Statistics*, pp. 1637–1664, 2012.

[86] O. Anava, E. Hazan, and A. Zeevi, "Online time series prediction with missing data," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 2191–2199.

[87] H. Yang and Q. Pan, Z.and Tao, "Online learning for time series prediction of ar model with missing data," *Neural Processing Letters*, vol. 50, no. 3, pp. 2247–2263, 2019.

[88] 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.

[89] Y. I. Abramovich, N. K. Spencer, and M. D. E. Turley, "Time-varying autoregressive (TVAR) adaptive order and spectrum estimation," in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, 2005.

[90] ——, "Order estimation and discrimination between stationary and time-varying (TVAR) autoregressive models," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2861–2876, Jun. 2007.

[91] S. Lundbergh, T. Teräsvirta, and D. V. Dijk, "Time-varying smooth transition autoregressive models," *J. Bus. Econ. Stat.*, vol. 21, no. 1, pp. 104–121, Jan. 2003.

[92] T. Kanada, M. Onuki, and Y. Tanaka, "Low-rank sparse decomposition of graph adjacency matrices for extracting clean clusters," in *Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2018, pp. 1153–1159.

[93] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2017.

[94] A. Cutkosky and K. A. Boahen, "Stochastic and adversarial online learning without hyperparameters," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5059–5067.

[95] L. M. Lopez-Ramos and B. Beferull-Lozano, "Online hyperparameter search interleaved with proximal parameter updates," *arXiv preprint arXiv:2004.02769*, 2020.

[96] G. Lewenfus, W. A. Martins, S. Chatzinotas, and B. Ottersten, "Joint forecasting and interpolation of graph signals using deep learning," *arXiv preprint arXiv:2006.01536*, 2020.

[97] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, 2013.

[98] H. Yang, Z. Xu, I. King, and M. R. Lyu, "Online learning for group lasso," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010.

[99] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 470–473, Aug. 2011.

[100] Z. Qin, K. Scheinberg, and D. Goldfarb, "Efficient block-coordinate descent algorithms for the group lasso," *Math. Prog. Computat.*, vol. 5, no. 2, pp. 143–169, Jun. 2013.

[101] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[102] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with varma recursions on graphs," *arXiv preprint arXiv:1810.08581*, 2018.

[103] R. Goebel, A. Roebroeck, D. Kim, and E. Formisano, "Investigating directed cortical interactions in time-resolved fmri data using vector autoregressive modeling and Granger causality mapping," *Magnet. Reson. Imag.*, vol. 21, no. 10, pp. 1251–1261, 2003.

[104] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, "Neural Granger causality for nonlinear time series," *arXiv preprint arXiv:1802.05842*, 2018.

[105] J. Songsiri, "Sparse autoregressive model estimation for learning Granger causality in time series," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, May 2013, pp. 3198–3202.

[106] M. Ayazoglu, M. Sznaier, and N. Ozay, "Blind identification of sparse dynamic networks and applications," in *IEEE Conf. Decision Control Eur. Control Conf.*, 2011, pp. 2944–2950.

[107] J. Lee, G. Li, and J. D. Wilson, "Varying-coefficient models for dynamic networks," *arXiv preprint arXiv:1702.03632*, 2017.

[108] Y. Shen and G. B. Giannakis, "Online identification of directional graph topologies capturing dynamic and nonlinear dependencies," in *IEEE Data Sci. Workshop*, 2018, pp. 195–199.

[109] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Statist. Soc.: Series B (Statist. Method.)*, vol. 68, no. 1, pp. 49–67, 2006.

[110] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, 2010.

[111] A. T. Puig, A. Wiesel, G. Fleury, and A. O. Hero, "Multidimensional shrinkage-thresholding operator and group lasso penalties," *IEEE Signal Process. Lett.*, vol. 18, no. 6, pp. 363–366, Jun. 2011.

[112] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley & Sons, 2003.

[113] 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.

[114] L. Kilian and H. Lütkepohl, *Structural Vector Autoregressive Analysis*. Cambridge University Press, 2017.

[115] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[116] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Siam, 2000, vol. 71.

[117] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[118] G. H. Golub, C. F. Van Loan, C. F. Van Loan, and P. C. F. Van Loan, *Matrix Computations*. The Johns Hopkins Univ. Press, 1996.

[119] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[120] 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, 2018.

[121] S. Akhavan and H. Soltanian-Zadeh, "Topology tracking of static and dynamic networks based on structural equation models," in *IEEE Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, pp. 107–112.

[122] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity," *arXiv preprint arXiv:1610.06551*, 2016.

[123] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 83–88, 2012.

[124] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[125] Y. Liu, L. Yang, G. Wenbin, T. Peng, and W. Wang, "Spatiotemporal smoothness-based graph learning method for sensor networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6.

[126] A. Beck, *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.