Sparse Online Learning with Kernels using Random Features for Estimating Nonlinear Dynamic Graphs

SUBMISSION DATE / POSTED DATE

21-02-2022 / 05-04-2023

CITATION

Money, Rohan; Krishnan, Joshin P.; Beferull-Lozano, Baltasar (2022): Sparse Online Learning with Kernels using Random Features for Estimating Nonlinear Dynamic Graphs. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.19210092.v4

DOI

10.36227/techrxiv.19210092.v4

Sparse Online Learning with Kernels using Random Features for Estimating Nonlinear Dynamic Graphs

Rohan Money, *Student Member, IEEE*, Joshin Krishnan, *Member, IEEE*,
Baltasar Beferull-Lozano, *Senior Member, IEEE*.

***Abstract*—Online topology estimation of graph-connected time series is challenging in practice, especially because the dependencies between the time series in many real-world scenarios are nonlinear. In this paper, we propose an online kernel-based algorithm for graph topology estimation. The algorithm also performs a Fourier-based Random feature approximation to tackle the curse of dimensionality associated with the kernel representations. Exploiting the fact that real-world networks often exhibit sparse topologies, we propose a group-Lasso based optimization framework, which is solved using an iterative composite objective mirror descent method, yielding an online algorithm with fixed computational complexity per iteration. We provide theoretical guarantees for the proposed algorithm and prove that the algorithm can achieve sublinear dynamic regret under certain reasonable assumptions. The experiments on real and synthetic data show that the proposed method outperforms its state-of-the-art competitors.**

***Index Terms*—Online graph learning, nonlinear topology identification, regret analysis, random Fourier features**

I. Introduction

Many practical networks such as large-scale cyber-physical systems (CPS), financial networks, brain networks, etc., generate multivariate time series data. In such systems, the time series are interdependent and it is possible to represent the dependencies in the form of graphs, or we can say that the multivariate time series is graph connected. Some of these dependencies are often imperceptible by direct inspection. Inferring and exploiting the hidden graph structure of data can have a significant impact in many application fields. For instance, it can contribute to developing better control actions in CPS [1], explainable analysis in brain networks [2], and improved forecast in financial time series [3], to name a few.

Real-world networks often exhibit time-delayed and directed dependencies between their components. For instance, consider an example of an oil and gas processing platform, as shown in Fig. 1. The system consists of wells and separators. The raw oil is extracted from the well and is separated as oil, water, and gas in the separators. It is a highly dynamic and complex system with hundreds of sensors and actuators. If an event occurs in a well, its effect will be reflected in the separators after a delay. Similarly, the oil level in separator-2 depends on the pressure that is controlled by an actuator in separator-3. The data acquired from such a system form a multivariate time series, possibly having many directed time-lagged interactions, which can be represented using a graph structure. Any information related to these dependencies is highly beneficial since it helps to predict the evolution of sensor variables in the near future and the appropriate control actions in advance. Although a scenario related to the oil and gas platform is adopted here for illustration, such interactions have a vital role in many important networks, such as brain data, the stock market, and smart water networks (SWN), to name a few. Hereafter, we use the term *topology identification* to denote the estimation of such dependencies.

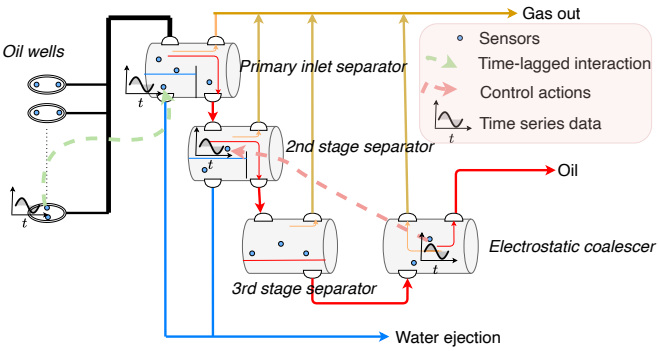


Figure 1: Schematic of processing stages in an oil and gas platform

A significant challenge associated with the aforementioned real-world graph-connected networks is the time-varying nature of the dependencies. There is extensive research in the field of online learning [4], [5], which outperforms classical batch solutions in terms of both computational complexity and ability to track changes. Such methods can be exploited and applied to topology identification in order to mitigate the problem of time-varying dependencies. For instance, [6] proposes a sparse online solution for topology identification using proximal updates, whereas [7] introduces a prediction-correction algorithm based on a time-varying convex optimization framework that exhibits an intrinsic temporal-regularization of the graph topology.

In addition to the dynamic nature, real-world systems such as the one shown in Fig. 1 are further complicated due to

This work was supported by the IKTPLUSS INDURB grant 270730/O70, the SFI Offshore Mechatronics grant 237896/O30.

Rohan T. Money is with the WISENET Lab, Dept. of ICT, University of Agder, Grimstad 4879, Norway (e-mail: rohantm@uia.no).
Joshin P. Krishnan is with the SIGIPRO Dept., Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (email: joshin@simula.no).
B. Beferull-Lozano is with the WISENET Center, Dept. of ICT, University of Agder, 4879 Grimstad, Norway, and also with the SIGIPRO Dept., Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: baltasar.beferull@uia.no).

the nonlinear nature of the dependencies. In CPSs such as Oil and Gas platforms or SWNs, this nonlinearity may arise from control mechanisms of the actuator, nonlinear liquid flows (see, e.g., [8]), a saturation of tanks, etc. Similarly, the interactions in stock market networks and network structured data related to brain imaging techniques, such as electroencephalography (EEG), electrocorticography (ECoG), positron emission tomography (PET), etc., also exhibit a high level of nonlinearities [9]. In such applications, topology estimation based on simple linear models [6], [7] is inadequate, since many of the inherent nonlinear interactions within the system are discarded.

An effective way to deal with the nonlinearity is by invoking kernel machines, which can approximate any nonlinear continuous function, provided enough training samples are available. For instance, in [10], a novel topology identification algorithm based on the nonlinear structural vector auto-regressive (SVAR) model using kernels is proposed. On the other hand, deep neural networks (DNNs) are powerful alternatives to kernels for modelling nonlinear interactions. Nonlinear dependencies are estimated in [11] using a temporal convolutional neural network and an attention mechanism, while [12] uses a vector autoregressive (VAR) model with an invertible neural network approach to capture dependencies, and [13] applies a group-Lasso regularizer on neural weights to obtain sparse nonlinear dependencies. Although the above-mentioned kernel- and DNN-based methods are powerful tools to model the nonlinear dependencies, their batch-based (offline) nature makes them unsuitable for real-time applications that require online topology estimation with every new data sample to track changes in the system. In addition, such batch-based approaches also suffer from a high computational complexity since the algorithm must process the entire data batch together.

The above discussion motivates the need for algorithms that can learn nonlinear and dynamic topologies. Kernels are an ideal choice in this regard due to their interpretability and capability to learn functions online [14]–[16]. In kernel frameworks, the data points are transformed to a function space, where a linear relationship exists between them. However, working in a function space has some limitations in the context of online topology identification. First, the standard online convex optimization techniques cannot be readily used as the dimension of optimization variables is not fixed, and it increases with every new data sample. Second, the number of parameters required to express the function increases with the number of data samples, and the computational complexity becomes prohibitive at some point, which is typically known as the curse of dimensionality [17]. This dimensionality growth is circumvented in [16] by discarding the past data samples using a forgetting window. However, such an approach can lead to suboptimal function learning because it discards data samples without assessing their significance in representing the functions to be learned.

Sparse kernel dictionaries and random feature (RF) approximation are two popular techniques for tackling the curse of dimensionality associated with kernels. A parsimonious online learning algorithm for kernels has been developed in [18] using a functional stochastic gradient descent (FSGD) method featured by sparse function subspace projections. This is achieved by learning sparse kernel dictionaries using the kernel orthogonal matching pursuit (KOMP) technique. Despite its reported benefits [18] in terms of model complexity compared to RF-based techniques, the sparse FSGD method in [18] has two limitations that render it an unfitting choice for online topology identification of multivariate time series: i) the algorithm need to include several KOMP sub-iterations for every time series at each time instant, which results in high computational complexity, not being suitable for online algorithms, particularly when the number of time series exceeds a few hundred, as it is typical in real-world networks such as the one shown in Fig. 1, and ii) in a multivariate setting with N time series, the FSGD derivation in [18] results in identical functional dependencies between a time series n and all other time series $n' = 1, 2, \ldots, N$ (as observed in [19]), which prevents distinguishing the different functional dependencies. In [20], an alternative approach to reduce the dimensionality growth of the kernel method for multivariate topology inference is presented, which involves learning a sparse kernel dictionary based on coherence criteria. Nevertheless, this algorithm's convergence guarantees assume that optimal parameters (representing the topology) do not change over time, which is impractical for time-varying systems.

On the other hand, the RF approximation approach not only addresses the problem of kernel dimensionality growth but also provides greater mathematical flexibility for modelling and learning the nonlinear interaction among multivariate time series, in addition to enabling a theoretical analysis. RF approximation was originally proposed in [21], and the idea has recently gained popularity in large-scale machine learning problems [22]–[24]. In addition to providing a computational boost in large-scale data sets, RF allows working in fixed lower dimensional spaces, which is very convenient for many online convex optimization routines. It has been shown that the RF approximation in kernels can be also used to understand neural networks [25], [26], and some researchers have shown equivalence in function approximation between neural networks and RF approximations [25]. Multiple Random Fourier features can be also utilized to initialize the learning process, and the best one can be kept to avoid overfitting [27], [28].

In this work, we propose a kernel-based online nonlinear topology identification algorithm using RF approximation. We assume that the dependencies of the system can be modelled using nonlinear additive sparse model. Notice that the sparsity assumption is not restrictive, since the interactions in real-world systems are often sparse due to the dominant local interactions. In fact, this prior information helps to avoid overfitting during learning. The proposed algorithm estimates nonlinear topologies in an online manner by generating sparse iterates at each time instant, using a proximal optimization technique known as Composite objective mirror descent (COMID). The algorithm features incremental updates to the model upon the arrival of new data samples, making it suitable for applications characterized by topology drifts [29], [30]. Through a combination of theoretical guarantees based on dynamic regret analysis and multiple numerical evidence, we

show the effectiveness of our algorithm in tracking the changes in topology.

The main contributions of this work are listed below:

(**i**) This paper proposes an online algorithm with fixed computational complexity per iteration for nonlinear topology estimation. The proposed algorithm is termed *Random feature-based nonlinear topology identification via recursive sparse online learning* (RFNL-TIRSO). This work is significantly different from our previous work in [31], where we used an instantaneous loss function, which is susceptible to noise and converges slowly. RFNL-TIRSO replaces the instantaneous loss function with an average running loss inspired by recursive least square (RLS) formulation, and compared to [31], it significantly improves convergence speed and robustness to the input noise.

(**ii**) We also provide theoretical guarantees regarding the convergence of RFNL-TIRSO, whereas no such theoretical guarantees were provided in [31]. The paper derives an upper bound for dynamic regret of RFNL-TIRSO based on the strong convexity property of the RLS loss function. Dynamic regret characterizes the tracking capability of an online algorithm [32], and we achieve a sublinear dynamic regret under certain assumptions that are reasonable in real-world applications. Our dynamic regret analysis includes three key elements: an online kernel-based nonlinear algorithm, a non-differentiable objective function, and a model with multiple decoupled functions representing topological connections to enable interpretable topology identification. None of the existing related analyses [33]–[39] provides a complete coverage of all these three elements.

(**iii**) The performance of the proposed algorithm is tested with extensive experiments using both real and synthetic data. The algorithm estimates interpretable topologies using time series data collected from the sensors of an oil and gas plant. In addition to the CPS applications, we also demonstrate the capability of our algorithm in detecting epileptic seizure events using EEG signals.

The rest of the paper is organized as follows: Section II presents the system model, kernel formulation, and RF approximation. In Section III, we develop the RFNL-TIRSO algorithm. Theoretical analysis of RFNL-TIRSO is performed in Section IV, and the numerical results are provided in Section V. Section VI concludes the paper.

Notations: Bold lowercase and uppercase letters denote column vectors and matrices, respectively. The operators ∇, $(.)^\top$, $\mathbb{E}$, $\Lambda_{max}(.)$, $\Lambda_{min}(.)$, $<.,.>$ respectively denote gradient, transpose, expectation, maximum eigenvalue, minimum eigenvalue, and inner product operators. The symbols $\mathbf{1}_N$ and $\boldsymbol{I}_N$ represent all-one vector of dimension N and identity matrix of dimension $N \times N$, respectively.

II. Nonlinear topology identification

A. System Model

Consider a collection of N sensors (nodes) generating a multi-variate time series denoted by $\mathbf{y}[t] \in \mathbb{R}^N$, where $t = 0, 1, \ldots, T-1$ denotes the time index. We assume that the dynamics of the sensor network can be captured by a P-th order VAR model with additive nonlinear functional dependencies:

$$y_n[t] = \sum_{n'=1}^{N} \sum_{p=1}^{P} f_{n,n'}^{(p)}(y_{n'}[t-p]) + u_n[t], \tag{1}$$

where $y_n[t]$ is the value of time series at time t observed at node $1 \leq n \leq N$, $f_{n,n'}^{(p)}$ is a nonlinear function that captures the influence of the p-lagged data point of node n' on node n, and $u_n[t]$ is the process noise, which is assumed to be zero mean i.i.d. random process. With respect to model (1), we define topology identification as the estimation of the functional dependencies $\left\{f_{n,n'}^{(p)}(.)\right\}_{p=1}^{P}$, $\forall n, n'$, from the observed time series $\{y_{n'}[t]\}_{n'=1}^{N}$.

B. Kernel representation

Assume that the functions $f_{n,n'}^{(p)}$ in (1) belong to a reproducing kernel Hilbert space (RKHS):

$$\mathcal{H}_{n'}^{(p)} := \left\{ f_{n,n'}^{(p)} \mid f_{n,n'}^{(p)}(y) = \sum_{t=p}^{\infty} \beta_{n,n',(t-p)}^{(p)} \, \kappa_{n'}^{(p)}(y, y_{n'}[t-p]) \right\}, \tag{2}$$

where $\kappa_{n'}^{(p)} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a positive definite kernel, which characterizes the RKHS. The kernel is a function measuring the similarity between the data points y and $y_{n'}[t-p]$. The expression (2) follows from the fact that any function in RKHS can be expressed as an infinite combination of kernel evaluations [40], i.e., the function $f_{n,n'}^{(p)}(y)$ can be expressed as the linear combination of the similarities between y and the data points $\{y_{n'}[t-p]\}_{t=p}^{t=\infty}$, with weights $\beta_{n,n',(t-p)}^{(p)}$. Here, we consider a Hilbert space with the inner product $\langle \kappa_{n'}^{(p)}(y, x_1), \kappa_{n'}^{(p)}(y, x_2) \rangle := \sum_{t=0}^{\infty} \kappa_{n'}^{(p)}(y[t], x_1) \kappa_{n'}^{(p)}(y[t], x_2)$ using kernels with reproducible property $\langle \kappa_{n'}^{(p)}(y, x_1), \kappa_{n'}^{(p)}(y, x_2) \rangle = \kappa_{n'}^{(p)}(x_1, x_2)$. Such a Hilbert space with the reproducing kernels is termed as RKHS, and the inner product described above induces the RKHS norm, $\|f_{n,n'}^{(p)}\|_{\mathcal{H}_{n'}^{(p)}}^2 = \sum_{t=0}^{\infty} \sum_{t'=0}^{\infty} \beta_{n,n',t}^{(p)} \, \beta_{n,n',t'}^{(p)} \, \kappa_{n'}^{(p)}(y_n[t], y_n[t'])$. We refer to [41] for further reading on RKHS.

The required functions $\left\{f_{n,n'}^{(p)} \in \mathcal{H}_{n'}^{(p)}\right\}_{n',p}$ at a particular node n can be obtained by solving the following non-parametric optimization problem in batch form, considering all the samples at once:

$$\left\{\hat{f}_{n,n'}^{(p)}\right\}_{n',p} = \underset{\left\{f_{n,n'}^{(p)} \in \mathcal{H}_{n'}^{(p)}\right\}}{\arg\min} \frac{1}{2} \sum_{\tau=P}^{T-1} \Bigg[y_n[\tau] - \sum_{n'=1}^{N} \sum_{p=1}^{P} f_{n,n'}^{(p)}(y_{n'}[\tau-p]) \Bigg]^2 + \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \Omega(||f_{n,n'}^{(p)}||_{\mathcal{H}_{n'}^{(p)}}). \tag{3}$$

For a non-decreasing function Ω, the solution of (3), denoted as $\left\{\hat{f}_{n,n'}^{(p)}\right\}_{n',p}$ can be obtained in terms of finite kernel evaluation by invoking the Representer Theorem [42]:

$$\hat{f}_{n,n'}^{(p)}(y_{n'}[\tau-p]) = \sum_{t=p}^{p+T-1} \hat{\beta}_{n,n',(t-p)}^{(p)} \, \kappa_{n'}^{(p)}(y_{n'}[\tau-p], y_{n'}[t-p]) \, . \tag{4}$$

Although the solution (4) entails only a finite number (equal to T) of kernel evaluations, its computational complexity becomes prohibitively high for a large value of T. This is

a major drawback associated with the kernel formulations, which is commonly referred to as *the curse of dimensionality*. In alignment with [24], [14], we use RF approximation to solve the curse of dimensionality.

C. RF approximation

From Section II-B, we remark that the RKHS is characterized by an inner product. Resorting to the theory of RF approximation, the inner product can be expressed in a random Fourier space, which facilitates the approximation of an RKHS function to a function in a fixed low dimensional space, thereby preventing the dimensionality growth. In addition to tackling the curse of dimensionality, working on a fixed low dimensional space will enable us to use the standard convex optimization tools to solve the topology identification.

The RF approximation requires that the kernel defining the RKHS should be shift invariant, i.e., $\kappa_{n'}^{(p)}\left(y_{n'}[\tau-p], y_{n'}[t-p]\right) = \kappa_{n'}^{(p)}\left(y_{n'}[\tau-p] - y_{n'}[t-p]\right)$. There are many popular kernels that are shift-invariant, such as the Laplacian, the Cauchy, and the Gaussian kernels. By Bochner's Theorem [43], every shift-invariant kernel can be expressed as an inverse Fourier transform of a probability density function. Following this theorem, the kernel evaluation can be expressed as

$$\begin{aligned}\kappa_{n'}^{(p)}&\left(y_{n'}[\tau-p], y_{n'}[t-p]\right)\\ &= \int_{\mathbb{R}} \pi_{\kappa_{n'}^{(p)}}(v)\; e^{jv\left(y_{n'}[\tau-p]-y_{n'}[t-p]\right)}dv\\ &= \mathbb{E}_v[e^{jv\left(y_{n'}[\tau-p]-y_{n'}[t-p]\right)}],\end{aligned} \tag{5}$$

where $\mathbb{E}$ is the expectation operation, $\pi_{\kappa_{n'}^{(p)}}(v)$ is the probability density function corresponding to the kernel under consideration, and v is the random variable associated with the probability density function. Using a sufficient amount of i.i.d. samples $\{v_i\}_{i=1}^{D}$ from the distribution $\pi_{\kappa_{n'}^{(p)}}(v)$, we can approximate the expectation in (5) as a sample mean (weak law of large numbers):

$$\begin{aligned}&\hat{\kappa}_{n'}^{(p)}\left(y_{n'}[\tau-p], y_{n'}[t-p]\right) =\\ &\frac{1}{D}\sum_{i=1}^{D} e^{jv_i\left(y_{n'}[\tau-p]-y_{n'}[t-p]\right)},\end{aligned} \tag{6}$$

irrespective of the distribution $\pi_{\kappa_{n'}^{(p)}}(v)$. Notice that (6) is an unbiased estimator of the kernel evaluation in (5) [44]. Finding the probability distribution, which is the inverse Fourier transform of a kernel, is a difficult task in general. However, for a Gaussian kernel with variance σ^2, the Fourier transform is also a Gaussian with variance σ^{-2}. Hence, in this work, we restrict our choice of the kernel to Gaussian kernels. Further, the real part of (6) is also an unbiased estimator of the kernel evaluation [22], and (5) can be expressed in vector form using only the real components as

$$\hat{\kappa}_{n'}^{(p)}\left(y_{n'}[\tau-p], y_{n'}[t-p]\right) = \boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau)^{\top}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t), \tag{7}$$

where

$$\begin{aligned}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau) = \frac{1}{\sqrt{D}}\Big[&\sin\left(v_1 y_{n'}[\tau-p]\right),\ldots,\sin\left(v_D y_{n'}[\tau-p]\right),\\ &\cos\left(v_1 y_{n'}[\tau-p]\right),\ldots,\cos\left(v_D y_{n'}[\tau-p]\right)\Big]^{\top}.\end{aligned} \tag{8}$$

Substitute (7) in (4) to obtain an approximation of the function $\hat{f}_{n,n'}^{(p)}$ in a fixed dimension (2D):

$$\begin{aligned}\hat{\hat{f}}_{n,n'}^{(p)}\left(y_{n'}[\tau-p]\right)) &= \sum_{t=p}^{p+T-1}\hat{\beta}_{n,n',(t-p)}^{(p)}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau)^{\top}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t)\\ &= \boldsymbol{\alpha}_{n,n'}^{(p)\top}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau),\end{aligned} \tag{9}$$

where $\boldsymbol{\alpha}_{n,n'}^{(p)} = \sum_{t=p}^{p+T-1}\hat{\beta}_{n,n',(t-p)}^{(p)}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t)$. For the sake of simplicity, we define the following notations:

$$\boldsymbol{\alpha}_{n,n'}^{(p)} = [\alpha_{n,n',1}^{(p)},\ldots,\alpha_{n,n',2D}^{(p)}]^{\top} \in \mathbb{R}^{2D}, \tag{10}$$

$$\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau) = [z_{\boldsymbol{v},n',1}^{(p)}(\tau),\ldots z_{\boldsymbol{v},n',2D}^{(p)}(\tau)]^{\top} \in \mathbb{R}^{2D}, \tag{11}$$

$$z_{\boldsymbol{v},n',k}^{(p)}(\tau) = \begin{cases}\sin(v_k y_{n'}[\tau-p]), & \text{if } k \le D\\ \cos(v_{k-D} y_{n'}[\tau-p]), & \text{otherwise.}\end{cases}$$

The functional optimization (3) can be reformulated as a parametric optimization problem using (9). First, we define the parametric form of the loss function in (3):

$$\mathcal{L}^n\left(\boldsymbol{\alpha}_{n,n'}^{(p)}\right) := \sum_{\tau=P}^{T-1}\frac{1}{2}\left[y_n[\tau] - \sum_{n'=1}^{N}\sum_{p=1}^{P}\boldsymbol{\alpha}_{n,n'}^{(p)\top}\;\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau)\right]^2, \tag{12}$$

which can be expanded in terms of RF components as

$$\mathcal{L}^n\left(\alpha_{n,n',d}^{(p)}\right) := \sum_{\tau=P}^{T-1}\frac{1}{2}\left[y_n[\tau] - \sum_{n'=1}^{N}\sum_{p=1}^{P}\sum_{d=1}^{2D}\alpha_{n,n',d}^{(p)}\; z_{\boldsymbol{v},n',d}^{(p)}(\tau)\right]^2.$$

For convenience, the variables $\left\{\alpha_{n,n',d}^{(p)}\right\}$ and $\left\{z_{\boldsymbol{v},n',d}^{(p)}(\tau)\right\}$ are stacked in the lexicographic order of the indices p, n', and d to obtain the vectors $\boldsymbol{\alpha}_n \in \mathbb{R}^{2PND}$ and $\boldsymbol{z}_{\boldsymbol{v}}(\tau) \in \mathbb{R}^{2PND}$, respectively, and loss function can be compactly rewritten as:

$$\mathcal{L}^n(\boldsymbol{\alpha}_n) = \frac{1}{2}\sum_{\tau=P}^{T-1}\left[y_n[\tau] - \boldsymbol{\alpha}_n^{\top}\boldsymbol{z}_{\boldsymbol{v}}(\tau)\right]^2. \tag{13}$$

Following [24], the original regularization term in (3) can be converted to an equivalent parametric form as:

$$\begin{aligned}&\Omega(||f_{n,n'}^{(p)}||_{\mathcal{H}_{n,n'}^{(p)}})\\ &= \Omega\left(\sqrt{\sum_{\tau=p}^{p+T-1}\sum_{t=p}^{p+T-1}\hat{\beta}_{n,n',(\tau-p)}^{(p)}\;\hat{\beta}_{n,n',(t-p)}^{(p)}\;k_{n'}^{(p)}(y_n(\tau),y_n(t))}\right)\\ &= \Omega\left(\sqrt{\sum_{\tau=p}^{p+T-1}\sum_{t=p}^{p+T-1}\hat{\beta}_{n,n',(\tau-p)}^{(p)}\hat{\beta}_{n,n',(t-p)}^{(p)}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(\tau)^{\top}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t)}\right)\\ &= \Omega(||\boldsymbol{\alpha}_{n,n'}^{(p)}||_2).\end{aligned} \tag{14}$$

The function Ω in (14) is chosen to be $\Omega(.) = |.|$, where $|.|$ represents the absolute value function, in order to promote the group sparsity of $\boldsymbol{\alpha}_{n,n'}^{(p)}$ [10]. Such regularizers are typically known as *group-Lasso regularizers* (see, Fig. 2 for a visual representation of the Lasso groups). Note that the function $|.|$ is non-decreasing, thereby satisying the regularization criteria

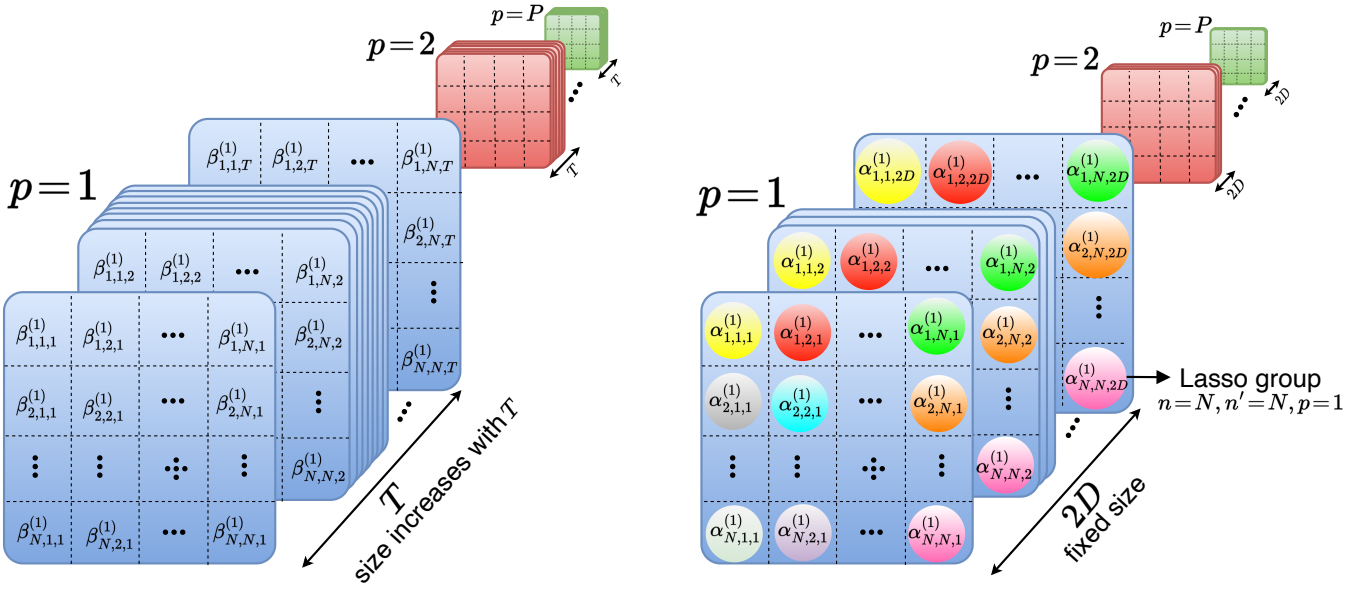


Figure 2: RKHS parameters (left) and fixed-size RF parameters (right). The Lasso groups of RF parameters are indicated in different colours.

to apply the Representer Theorem. Using (13) and (14), a parametric form of (3) can be constructed as follows:

$$\{\widehat{\boldsymbol{\alpha}}_n\}_{n'} = \arg\min_{\{\boldsymbol{\alpha}_n\}} \mathcal{L}^n(\boldsymbol{\alpha}_n) + \lambda \sum_{n'=1}^{N}\sum_{p=1}^{P} ||\boldsymbol{\alpha}_{n,n'}^{(p)}||_2. \tag{15}$$

Although the topology can be estimated by solving (15), this approach has several drawbacks since it is a batch formulation, meaning that (15) requires the entire batch of the time series samples $y_n[t],\ t = 0, 1, \ldots, T-1$ from all the nodes. In addition, the batch formulation is not useful when the data is available in a streaming manner and cannot be used to track the instantaneous time-varying topologies. Moreover, since the batch optimization computes the solutions using an entire batch of data, the computational complexity can often become prohibitively high, especially when batch size is huge. Hence, motivated by the above factors, we propose an online topology estimation strategy with a lower computational complexity in the following section.

III. Online Learning

To formulate an an online optimization framework, we replace the batch loss function $\mathcal{L}^n(\boldsymbol{\alpha}_n)$ in (15) with a stochastic (instantaneous) loss function $\ell_t^n(\boldsymbol{\alpha}_n) = \frac{1}{2}[y_n[t] - \boldsymbol{\alpha}_n^\top \boldsymbol{z}_{\boldsymbol{v}}(t)]^2$:

$$\widehat{\boldsymbol{\alpha}}_n = \arg\min_{\boldsymbol{\alpha}_n} \ell_t^n(\boldsymbol{\alpha}_n) + \lambda \sum_{n'=1}^{N}\sum_{p=1}^{P} \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2. \tag{16}$$

The loss function $l_t^n(\boldsymbol{\alpha}_n)$ in (16) is analogous to a Least Mean Square (LMS) formulation. However, notice that the estimates of LMS are prone to observation noise and can be unstable in practice. To avoid this problem, we formulate (16) in a recursive least square (RLS) sense, which further provides necessary stability in addition to faster convergence:

$$\tilde{\ell}_t^n(\boldsymbol{\alpha}_n) = \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \ell_\tau^n(\boldsymbol{\alpha}_n). \tag{17}$$

In (17), we replace the instantaneous loss with a running average loss using an exponential window. The parameter $\gamma \in (0, 1)$ is the forgetting factor of the window, and $\mu = 1-\gamma$ is set to normalize the exponential weighting window. We expand the RLS loss function as follows:

$$\tilde{\ell}_t^n(\boldsymbol{\alpha}_n) = \frac{1}{2}\mu \sum_{\tau=P}^{t-1} \gamma^{t-\tau} \Big(y_n^2[\tau] + \boldsymbol{\alpha}_n^\top \boldsymbol{z}_{\boldsymbol{v}}(\tau)\boldsymbol{z}_{\boldsymbol{v}}(\tau)^\top \boldsymbol{\alpha}_n - 2y_n[\tau]\boldsymbol{z}_{\boldsymbol{v}}(\tau)^\top \boldsymbol{\alpha}_n \Big) \tag{18}$$

$$= \frac{1}{2}\mu \sum_{\tau=P}^{t-1} \gamma^{t-\tau} y_n^2[\tau] + \frac{1}{2}\boldsymbol{\alpha}_n^\top \boldsymbol{\Phi}[t]\boldsymbol{\alpha}_n - \boldsymbol{r}_n[t]^\top \boldsymbol{\alpha}_n, \tag{19}$$

where

$$\boldsymbol{\Phi}[t] = \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \boldsymbol{z}_{\boldsymbol{v}}(\tau)\boldsymbol{z}_{\boldsymbol{v}}(\tau)^\top, \tag{20}$$

$$\boldsymbol{r}_n[t] = \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau]\boldsymbol{z}_{\boldsymbol{v}}(\tau). \tag{21}$$

As in a typical RLS formulation, these quantities can be updated recursively as $\boldsymbol{\Phi}[t] = \gamma\boldsymbol{\Phi}[t-1] + \mu\boldsymbol{z}_{\boldsymbol{v}}(t)\boldsymbol{z}_{\boldsymbol{v}}(t)^\top$ and $\boldsymbol{r}_n[t] = \gamma\boldsymbol{r}_n[t-1] + \mu y_n[t]\boldsymbol{z}_{\boldsymbol{v}}(t)$. The gradient of the loss function can be obtained as

$$\nabla\tilde{\ell}_t^n(\boldsymbol{\alpha}_n) = \boldsymbol{\Phi}[t]\boldsymbol{\alpha}_n - \boldsymbol{r}_n[t]. \tag{22}$$

Finally, using the RLS loss function, the topology can be estimated by solving

$$\arg\min_{\boldsymbol{\alpha}_n} \tilde{\ell}_t^n(\boldsymbol{\alpha}_n) + \lambda \sum_{n'=1}^{N}\sum_{p=1}^{P} \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2. \tag{23}$$

The cost function in (23) consists of a differentiable loss function and a non-differentiable group-Lasso regularizer. The online subgradient descent (OSGD) or the mirror descent (MD) method can be used to solve (23) online. However, these methods work by linearizing the entire objective function in

(23) using a subgradient of it. If the group-Lasso regularizer is linearized, its ability to induce sparsity is compromised, resulting in non-sparse estimates. Hence, we choose an alternate optimization technique known as composite objective mirror descent (COMID) [45], a modified version of the MD algorithm, in which the differentiable part of the objective function is linearized, whereas the regularizer is kept intact.

The online COMID updates can be written as

$$\boldsymbol{\alpha}_n[t+1] = \arg\min_{\boldsymbol{\alpha}_n} J_t^{(n)}(\boldsymbol{\alpha}_n), \tag{24}$$

$$\text{where } J_t^{(n)}(\boldsymbol{\alpha}_n) \triangleq \nabla\tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])^\top(\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]) + \frac{1}{2a_t}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]\|_2^2 + \lambda\sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2, \tag{25}$$

where $\boldsymbol{\alpha}_n[t] \in \mathbb{R}^{2PND}$ is the estimate of $\boldsymbol{\alpha}_n$ at time t. The objective function $J_t^{(n)}$ in (25) consists of 3 parts: (i) gradient of loss function given by (22), (ii) a Bregman divergence term with a_t as the step size, and (iii) a sparsity enforcing group-Lasso regularizer. The Bregman divergence [46] improves the stability of the online algorithms by constraining the value of the new estimate $\boldsymbol{\alpha}_n[t+1]$ within the proximity of the previous estimate $\boldsymbol{\alpha}_n[t]$. The Bregman divergence $B(\boldsymbol{\alpha}_n, \boldsymbol{\alpha}_n[t]) = \frac{1}{2}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]\|_2^2$ is selected in such a way that the optimization problem (24) has a closed form solution [46]. For notational convenience, we denote the gradient in (25) as

$$\mathbf{v}_n[t] := \nabla\tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t]). \tag{26}$$

The objective function in (25) is expanded by omitting the constants leading to the following formulation:

$$J_t^{(n)}(\boldsymbol{\alpha}_n) \propto \frac{\boldsymbol{\alpha}_n^\top\boldsymbol{\alpha}_n}{2a_t} + \boldsymbol{\alpha}_n^\top\left(\mathbf{v}_n[t] - \frac{1}{a_t}\boldsymbol{\alpha}_n[t]\right) + \lambda\sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2$$

$$= \sum_{n'=1}^{N}\sum_{p=1}^{P}\Bigg[\frac{{\boldsymbol{\alpha}_{n,n'}^{(p)}}^\top\boldsymbol{\alpha}_{n,n'}^{(p)}}{2a_t} + {\boldsymbol{\alpha}_{n,n'}^{(p)}}^\top\left(\mathbf{v}_{n,n'}^{(p)}[t] - \frac{1}{a_t}\boldsymbol{\alpha}_{n,n'}^{(p)}[t]\right) + \lambda\|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2\Bigg]. \tag{27}$$

A closed form solution for (24) using (27) can be obtained via the multidimensional shrinkage-thresholding operator [47]:

$$\boldsymbol{\alpha}_{n,n'}^{(p)}[t+1] = \left(\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - a_t\mathbf{v}_{n,n'}^{(p)}[t]\right) \times \left[1 - \frac{a_t\lambda}{\|\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - a_t\mathbf{v}_{n,n'}^{(p)}[t]\|_2}\right]_+, \tag{28}$$

where $[{\mathbf{v}_{n,n'}^{(1)}}^\top, {\mathbf{v}_{n,n'}^{(2)}}^\top, \ldots, {\mathbf{v}_{n,n'}^{(P)}}^\top]^\top \triangleq \mathbf{v}_{n,n'}$ for $n' = 1\ldots N$, $[\mathbf{v}_{n,1}^\top, \mathbf{v}_{n,2}^\top, \ldots, \mathbf{v}_{n,N}^\top]^\top \triangleq \nabla\tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])$, and $[x]_+ = \max\{0, x\}$. The first part $\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - \gamma_t\mathbf{v}_{n,n'}^{(p)}[t]$ in (28) forces the stochastic gradient update of $\boldsymbol{\alpha}_{n,n'}^{(p)}$ in a way to descend the recursive loss function $\tilde{\ell}_t^n(\boldsymbol{\alpha}_n)$, and the second part in (28) enforces group sparsity of $\boldsymbol{\alpha}_{n,n'}^{(p)}$. This closed-form expression estimates the required dependency between the time series y_n and the p-th time lagged value of time series $y_{n'}$ at time instant $t+1$, in terms of the parameter vector $\boldsymbol{\alpha}_{n,n'}^{(p)}[t+1]$. We name the proposed algorithm as *Random feature based nonlinear topology identification via recursive sparse online learning* (RFNL-TIRSO), which is shown in **Algorithm 1**.

Algorithm 1: RFNL-TIRSO Algorithm

Result: $\left\{\boldsymbol{\alpha}_{n,n'}^{(p)}\right\}_{n,n',p}$
Store $\{\boldsymbol{y}_n[t]\}_{t=1}^P$,
Initialize $\lambda > 0$, $a_t > 0$, $\theta > 0$, D, σ_n and $\boldsymbol{\Phi}(P-1) = \theta\boldsymbol{I}_{2PND}$
for $t = P, P+1, \ldots$ **do**
  Get data samples $y_n[t]$, $\forall n$ and compute $\boldsymbol{z_v}(t)$
  $\boldsymbol{\Phi}[t] = \gamma\boldsymbol{\Phi}[t-1] + \mu\boldsymbol{z_v}(t)\boldsymbol{z_v}(t)^\top$
  for $n = 1, \ldots, N$ **do**
    $\boldsymbol{r}_n[t] = \gamma\boldsymbol{r}_n[t-1] + \mu y_n[t]\boldsymbol{z_v}(t)$
    compute $\mathbf{v}_n[t]$ using (22), (26)
    for $n' = 1, \ldots, N$ **do**
      compute $\boldsymbol{\alpha}_{n,n'}^{(p)}[t+1]$ using (28)
    end
  end
end

IV. Theoretical results

The performance analysis and convergence guarantee of RFNL-TIRSO are presented in this section using dynamic regret analysis. Regret is a popular metric to measure the performance of an online algorithm [48]. Despite being originally developed for static learning problems, numerous online algorithms involving dynamic regret analysis have been developed [33]–[36] to solve problems in a dynamic environment; however all of them belong to the class of linear algorithms. Moreover, [33]–[35] assume differentiable objective functions, and hence they cannot be leveraged in RFNL-TIRSO. Dynamic regret bounds for nonlinear algorithms are proposed in [37]–[39]. In [37], the problem under consideration is limited to positive functions, whereas our problem formulation does not have such a limitation. The regret analysis presented in [38] differs significantly from the proposed method for several reasons. First, the objective function used in [38] must be differentiable, while in our proposed method, the regularizer is non-differentiable. Second, in contrast to [38], the regret analysis in the proposed method involves multiple decoupled functions representing interpretable topological connections. Although [39] provides a logarithmic regret bound using second-order information, the objective function under consideration is differentiable.

Our theoretical analysis is based on the following assumptions:

- $\boldsymbol{A1}$: Bounded samples: For all the time series samples, there exists $B_y > 0$ such that $\left\{|y_n[t]|^2\right\}_{n,t} \leq B_y \leq \infty$.
- $\boldsymbol{A2}$: Shift-invariant kernels: kernels used are shift-invariant, i.e., $k(x_i, x_j) = k(x_i - x_j)$.
- $\boldsymbol{A3}$: Bounded minimum eigenvalue of $\boldsymbol{\Phi}[t]$: There exists $\rho_l > 0$ such that $\Lambda_{min}(\boldsymbol{\Phi}[t]) > \rho_l$, where $\Lambda_{min}(.)$ denotes the minimum eigenvalue.
- $\boldsymbol{A4}$: Bounded maximum eigenvalue of $\boldsymbol{\Phi}[t]$: There exists $L > 0$ such that $\Lambda_{max}(\boldsymbol{\Phi}[t]) < L < \infty$, where $\Lambda_{max}(.)$ denotes the maximum eigenvalue.

$\boldsymbol{A1}$ is reasonable in practice as the signals from real-world applications are bounded. $\boldsymbol{A2}$ is true for typical kernels such

as Gaussian, Laplacian, etc. Since $\boldsymbol{\Phi}(t)$ is a sum of rank one matrices formed using feature vectors, $\boldsymbol{A3}$ will hold as long as the feature vectors are linearly independent. This is a reasonable assumption in practice when a sufficient amount of data is available. Note that $\boldsymbol{A3}$ is important for the strong convexity assumption of the loss function, which is used in the sequel. $\boldsymbol{A4}$ can be obtained by combining $\boldsymbol{A1}$ and the fact that the sum of eigenvalues of $\boldsymbol{\Phi}[t]$ is equal to its trace.

A. Dynamic Regret Analysis

As a preliminary step to the regret analysis, we define the optimum RKHS and RF coefficients.

Optimum RKHS coefficients: Using the batch form solution (4), obtained using the Representer Theorem, a parametric autoregressive representation at time t can be obtained as

$$\widehat{y}_n[t] = \hat{\boldsymbol{\beta}}_n^{\top} \boldsymbol{\kappa}_t, \tag{29}$$

where $\hat{\boldsymbol{\beta}}_n \in \mathbb{R}^{NPt}$ and $\boldsymbol{\kappa}_t \in \mathbb{R}^{NPt}$ are respectively obtained by stacking the variables $\hat{\beta}^{(p)}_{n,n',(\tau-p)}$ and the kernel evaluations in (4) along the lexicographic order of the indices n',p, and the time index up to t. The optimum RKHS coefficients $\boldsymbol{\beta}^*_n[t]$ for each node n at time t can be obtained by solving

$$\boldsymbol{\beta}^*_n[t] = \arg\min_{\hat{\boldsymbol{\beta}}_n} h^n_t(\hat{\boldsymbol{\beta}}_n), \tag{30}$$

where the cost function $h^n_t(\hat{\boldsymbol{\beta}}_n)$ in (30) is composed of two terms: $h^n_t(\hat{\boldsymbol{\beta}}_n) = \tilde{\ell}^n_t(\hat{\boldsymbol{\beta}}_n) + \omega^n(\hat{\boldsymbol{\beta}}_n)$, where $\tilde{\ell}^n_t(.)$ is the RLS loss function defined in (17) with instantaneous losses computed as $\ell^n_t(\hat{\boldsymbol{\beta}}_n) = \frac{1}{2}[y_n[t] - \hat{\boldsymbol{\beta}}_n^{\top}\boldsymbol{\kappa}_t]^2$, and $\omega^n(.)$ is the group-Lasso regularizer defined as $\omega^n(\hat{\boldsymbol{\beta}}_n) = \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\hat{\boldsymbol{\beta}}^{(p)}_{n,n'}\|_2$.

Optimum RF coefficients: Following the same procedure, we define the optimum RF coefficients $\boldsymbol{\alpha}^*_n[t]$ at time $t > P$ as

$$\boldsymbol{\alpha}^*_n[t] = \arg\min_{\boldsymbol{\alpha}_n} h^n_t(\boldsymbol{\alpha}_n), \tag{31}$$

where $h^n_t(\boldsymbol{\alpha}_n) = \tilde{\ell}^n_t(\boldsymbol{\alpha}_n) + \omega^n(\boldsymbol{\alpha}_n)$, and $\tilde{\ell}^n_t(.)$ is the RLS loss defined in (17) and $\omega^n(\boldsymbol{\alpha}_n) = \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{\alpha}^{(p)}_{n,n'}\|_2$. It should be noticed that the optimum RF coefficients $\boldsymbol{\alpha}^*_n[t]$ is different from the RFNL-TIRSO estimate $\boldsymbol{\alpha}_n[t]$ obtained by the computationally light COMID algorithm, as RFNL-TIRSO only makes one COMID update per time instant.

Dynamic Regret: Dynamic Regret is defined as the cumulative sum of the difference between the estimated cost function and the optimal cost function over all time instants. In our framework, it can be expressed as

$$\boldsymbol{R}_n[T] = \sum_{t=P}^{T-1} \Big[h^n_t(\boldsymbol{\alpha}_n[t]) - h^n_t(\boldsymbol{\beta}^*_n[t]) \Big]. \tag{32}$$

Our aim is to find a theoretical bound for $\boldsymbol{R}_n[T]$. Since our online algorithm works in the RF space, we perform the regret analysis with reference to the optimal cost function in the RF space, i.e., $h^n_t(\boldsymbol{\alpha}^*_n[t])$. Notice that this is without loss of generality because there is a one-to-one mapping. Adding and subtracting $h^n_t(\boldsymbol{\alpha}^*_n[t])$ in (32) yields

$$\boldsymbol{R}_n[T] = \boldsymbol{R}^{\text{rf}}_n[T] + \boldsymbol{\xi}_n[T], \tag{33}$$

where $\boldsymbol{R}^{\text{rf}}_n[T] = \sum_{t=P}^{T-1} \left(h^n_t(\boldsymbol{\alpha}_n[t]) - h^n_t(\boldsymbol{\alpha}^*_n[t])\right)$ is the regret with respect to optimal cost in RF space and $\boldsymbol{\xi}_n[T] = \sum_{t=P}^{T-1} \left(h^n_t(\boldsymbol{\alpha}^*_n[t]) - h^n_t(\boldsymbol{\beta}^*_n[t])\right)$ is the cumulative RF approximation error caused by the dimensionality reduction.

1) Bounding the regret w.r.t. optimal cost function in RF space: **Theorem 1** bounds $\boldsymbol{R}^{\text{rf}}_n(T)$.

Theorem 1. *Under the assumptions of A1, A3, A4, and letting $a_t = \frac{1}{L}$, the dynamic regret of RFNL-TIRSO (**Algorithm 1**) w.r.t. the optimal cost function in the RF space satisfies*

$$\boldsymbol{R}^{rf}_n(T) \leq \Big(\Big(1 + \frac{L}{\rho_l}\Big) \sqrt{2PNDB_y} + \lambda\sqrt{PN} \Big) \times \Big(\|\boldsymbol{\alpha}^*_n[P]\|_2 + \boldsymbol{W}_n(T) \Big),$$

where $\boldsymbol{W}_n(T) = \sum_{t=P}^{T-1} \|\boldsymbol{\alpha}^*_n[t] - \boldsymbol{\alpha}^*_n[t-1]\|_2$ is the path length.

Proof: See Appendix A.

From **Theorem 1**, it can be readily seen that if $\boldsymbol{W}_n(T)$ is sublinear, then the regret will also be sublinear.

2) Bounding the cumulative RF approximation error: **Theorem 2** provides a bound for $\boldsymbol{\xi}_n(T)$.

Theorem 2. *Under assumptions A1 and A2, there exists $\epsilon \geq 0$ such that the cumulative approximation error $\boldsymbol{\xi}_n[T]$ of RFNL-TIRSO (**Algorithm 1**) satisfies*

$$\boldsymbol{\xi}_n(T) \leq \epsilon L_h T C,$$

where $L_h > 0$ is the Lipschitz continuity parameter of the cost function.

Proof: See Appendix B.

Finally, we bound the dynamic regret $\boldsymbol{R}_n(T)$ using **Theorem 1** and **Theorem 2**.

Theorem 3. *Under the assumptions of A1, A2, A3, and A4, the dynamic regret $\boldsymbol{R}_n(T)$ of RF-NLTIRSO (**Algorithm 1**) satisfies*

$$\boldsymbol{R}_n(T) \leq \Big(\Big(1 + \frac{L}{\rho_l}\Big) \sqrt{2PNDB_y} + \lambda\sqrt{PN} \Big) \times \Big(\|\boldsymbol{\alpha}^*_n[P]\|_2 + \boldsymbol{W}_n(T) \Big) + \epsilon L_h T C.$$

Proof: **Theorem 3** can be directly and readily proved by substituting **Theorem 1** and **Theorem 2** in (33).

Notice that if we have setting $\epsilon = \mathcal{O}(\frac{1}{\sqrt{T}})$, this results in a dynamic regret of $\mathcal{O}(\boldsymbol{W}_n(T) + \sqrt{T})$. In such cases, the dynamic regret is sublinear, if $\boldsymbol{W}_n(T)$ is sublinear. Ideally, an online algorithm must obtain a sublinear dynamic regret, which implies that $\boldsymbol{R}_n(T)/T \to 0$ as $T \to \infty$, or in the worst case, a linear regret which implies $\boldsymbol{R}_n(T)/T \to$ *constant*, where *constant* is known as the steady-state error. Notice that in our case, this steady state error when $\boldsymbol{W}_n(T)$ is sublinear is $\epsilon L_f C$. If ϵ is small, the resulting study state error will also be small. As shown in appendix $\boldsymbol{B}$, we can make ϵ sufficiently small by increasing the number of random features D by trading off with complexity [21].

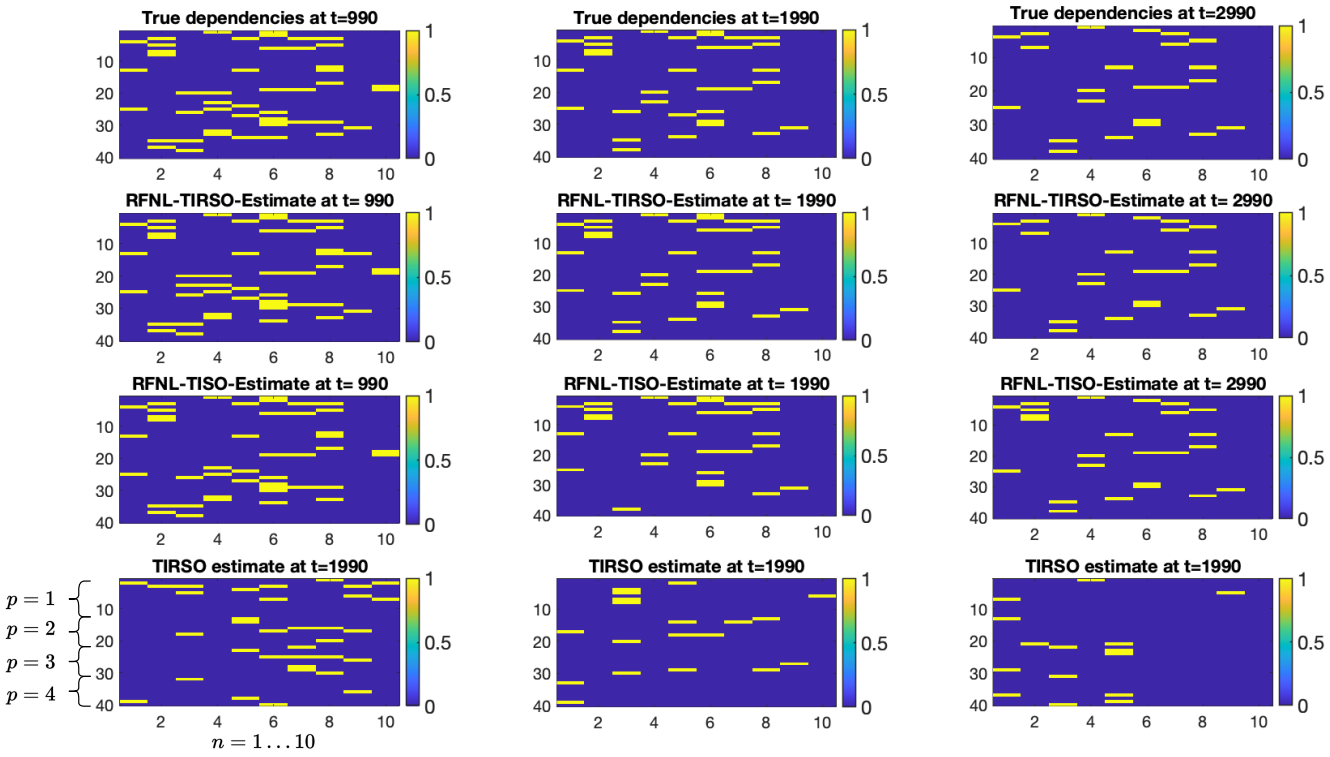
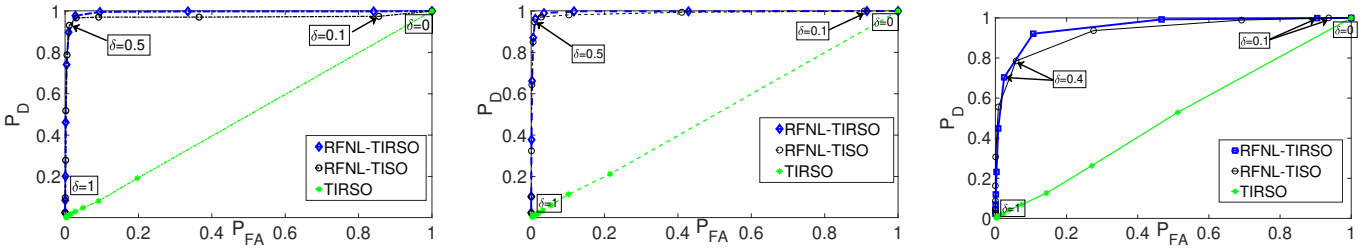


Figure 3: The true and estimated edges using various algorithms for $g(x) = g_1(x)$. In each subfigure, the x-axis corresponds to nodes $n = 1, \ldots, 10$, and the y-axis corresponds to nodes $n = 1, \ldots, 10$ for time lags $p = 1, \ldots, 4$. The edge values are indicated by the colour code.



(a) g_1 (b) g_2 (c) g_3

Figure 4: Receiver-Operating Curve for different realizations of the nonlinear function $g(x)$.

V. Experimental results

In this section, we analyze the performance of RFNL-TIRSO using extensive numerical experiments. We choose TIRSO [6], RFNL-TISO [31], and PDIS [20], [49], as the state-of-the-art competitors to compare the performance of RFNL-TIRSO. It is to be remarked that TIRSO is an online topology algorithm designed by assuming linear VAR models. TIRSO is selected in order to show the advantages of the proposed nonlinear algorithm RFNL-TIRSO, compared to its linear counterpart. The second algorithm RFNL-TISO is an online nonlinear topology estimation algorithm designed by considering an instantaneous least mean square loss function. Based on the discussions in Section III, RFNL-TIRSO is expected to show better performance compared to RFNL-TISO since it incorporates an RLS-based loss function. The third algorithm, PDIS [20], [49], is a recent online nonlinear topology identification algorithm using dictionaries of kernel functions based on partial-derivative-imposed sparsity. To the best of our knowledge, these three algorithms are the best benchmarks to compare the performance of RFNL-TIRSO, and although some other batch-based algorithms are available [10], [13], [12], they are not comparable to our algorithm, since they are offline algorithms.

The per node computational complexity of RFNL-TIRSO, RFNL-TISO, and TIRSO, are in the order of $\mathcal{O}\big(N^2P^2D^2\big)$, $\mathcal{O}\big(NPD\big)$, and $\mathcal{O}\big(N^2P^2\big)$, respectively. Although RFNL-TIRSO is computationally heavier than the competitors, it provides robustness, and theoretical performance guarantees, which is not the case for the competing algorithms and which we demonstrate through several numerical experiments in this section.

Experiments shown in this section are conducted using both synthetic and real data sets. The synthetic dataset includes graph-connected time series data generated by assuming dif-

ferent topology transition patterns to highlight the ability of the algorithms to track time-varying topologies. The real data sets include (i) time series data collected from Lundin's offshore oil and Gas platform[1] and (ii) Epileptic seizure data [50].

A. Experiments using Synthetic Data Sets

1) Piecewise stationary topology: We generate a multivariate time series using a nonlinear VAR model (1) with $N = 10, P = 4$. The nonlinear function in (1) is taken as $f^{(p)}_{n,n'}(x) = a^{(p)}_{n,n'}(x)g(x)$, where $g(x)$ is a nonlinear function and $a^{(p)}_{n,n'}(x) \in \{0, 1\}$. The experiments are conducted with three different realizations of $g(x)$: $g_1(x) = 0.25\sin(x^2)+0.25\sin(2x)+0.5\sin(x)$, $g_2(x) = 0.25\cos(x^2) + 0.25\cos(2x) + 0.5\cos(x)$, and with a Gaussian kernel, i.e., $g_3(x) = (1/\sqrt{2\pi})\exp(-x^2/2)$. We refer to $a^{(p)}_{n,n'}$ as an *edge*, and $a^{(p)}_{n,n'}(.) = 0/1$ means that the p-th time-lagged dependency between n and n' is disabled/enabled. A graph-connected time series is generated by restricting the number of active edges to be 30% of the total available edges. Further, we introduce abrupt changes in the topology after every 1000 time step by randomly cutting off 30% of the available active edges. Notice that the initial P data samples are generated randomly, and the rest of the data are generated using model (1). The hyperparameters of all the algorithms used in the experiments are tuned heuristically to get the maximum area under the receiver operating curve, which is explained below. The hyperparameter settings for RFNL-TIRSO are $(\sigma_n, \lambda, a_t) = (2.5, 0.01, 0.1/\Lambda_{max}(\boldsymbol{\phi}[t]))$, for g_1 and g_2, and $(1, 0.01, 0.1/\Lambda_{max}(\boldsymbol{\phi}[t]))$ for g_3. The top row of Fig. 3 contains the true edges $\left\{a^{(p)}_{n,n'}\right\}$ at different time steps, which are arranged in matrices of size $N \times N$, for $p = 1, 2, \ldots, P$, and stacked vertically, resulting in matrices of size $NP \times N$. The estimated dependencies $\left\{\hat{a}^{(p)}_{n,n'}\right\}$ using different algorithms are shown in the bottom rows. After computing the normalized ℓ_2 norms $b^{(p)}_{n,n'}[t] = \|\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\|_2/(\max_{n'}\|\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\|_2)$, the presence of an edge is detected using a threshold δ as $\hat{a}^{p}_{n,n'} = \mathbb{1}\{b^{(p)}_{n,n'}[t] < \delta\}$, where $\mathbb{1}\{x\} = 1/0$, if x is true/false. It is clear from Fig. 3 that the estimates of RFNL-TISO are very close to the ground truth, and they outperform others.

A numerical comparison of the performances of the algorithms is made using the probability of false alarm (P_{FA}) and the probability of detection (P_D). The probability of false alarm (P_{FA}) refers to the probability that the algorithm reports the presence of a dependency in the network that is not actually present. On the other hand, the probability of detection(P_D), refers to the probability that the algorithm detects a dependency that is truly present in the network. In our experiment, we assume there is a presence of a detected edge from the $p-th$ time-lagged value of $n'-th$ sensor to the present value of the $n-th$ sensor if the value of coefficient $b^{(p)}_{n,n'}[t]$ is greater than a threshold $\delta \in [0, 1]$, and define P_{FA} and P_D as

$$P_{\text{D}}[t] \triangleq 1 - \frac{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\mathbb{1}\{b^{(p)}_{n,n'}[t] < \delta\}\mathbb{1}\{a_{n,n'} = 1\}\right]}{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}[\mathbb{1}\{a_{n,n'} = 1\}]},$$
$$P_{\text{FA}}[t] \triangleq \frac{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\mathbb{1}\{b^{(p)}_{n,n'}[t] > \delta\}\mathbb{1}\{a_{n,n'} = 0\}\right]}{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\mathbb{1}\{a_{n,n'} = 0\}\right]}, \quad (34)$$

[1]https://www.lundin-energy.com/

where $\mathbb{1}\{x\} = 1/0$, if x is true/false and δ is a threshold. From (34), it is clear that when $\delta = 0$, both P_D and P_{FA} become one. With an increase in δ, both P_D and P_{FA} decrease, eventually reaching zero when δ equals one.

The Receiver-Operating curve (ROC) of the different algorithms at time $t = 990$ is plotted in Fig. 4 by varying δ from 0 to 1, with P_{FA} in the x-axis and P_D in the y-axis. The area under the ROC curve (AUC) is computed to evaluate the performance of the algorithm. A topology identification algorithm with a high AUC value is characterized by by a high P_D and low P_{FA}, indicating that it can accurately identify network topologies while minimizing the occurrence of false positives. From Fig. 4, it can be observed that the area under ROC (AUC) of the RFNL-TIRSO is substantially better than TIRSO and slightly better than RFNL-TISO for all three nonlinearity functions. These observations are more evident from Table I, where the computed AUC values are tabulated. We further analyze the AUC of RFNL-TIRSO for different RF space dimensions, i.e., $D \in \{20, 30, 50\}$, at different time instants in Table II, for the nonlinear function $g(x) = g_1(x)$. As expected, the AUC increases with D and the number of data samples. A similar AUC trend as in Table II was obtained for the other two nonlinear functions g_1 and g_2.

Table I: AUC for different algorithms.

AUC	g_1	g_2	g_3
$RFNL-TIRSO$	**0.9914**	**0.9949**	**0.9543**
$RFNL-TISO$	0.9741	0.9817	0.9317
$TIRSO$	0..4967	0.5	.5123

Table II: AUC curve for different values of D.

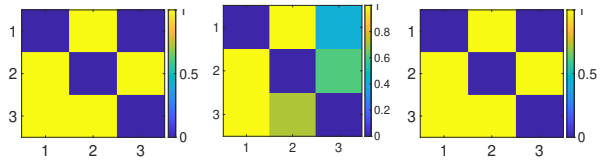
AUC	$t = 990$	$t = 1990$	$t = 2990$
$D = 20$	0.9500	0.9762	0.9732
$D = 30$	0.9568	0.9827	0.9835
$D = 50$	0.9721	0.9887	**0.9901**

2) Lorenz graph: We also present experiments with synthetic data sets generated using the Lorenz graph [51]. We consider a discretized version of the Lorenz graph involving 3 time series exhibiting the following nonlinear dependencies:

$$\begin{pmatrix} y_1[t+1] \\ y_2[t+1] \\ y_3[t+1] \end{pmatrix} = 0.01\begin{pmatrix} 10(y_2[t]-y_1[t]) \\ y_1[t](28-y_3[t])-y_2[t] \\ y_1[t]y_2[t]-\frac{8}{3}y_3[t] \end{pmatrix} + \begin{pmatrix} y_1[t] \\ y_2[t] \\ y_3[t] \end{pmatrix} \quad (35)$$

Compared to the model used in Section V-A1, the Lorenz graph model (35) involves only order one ($P = 1$) time lag dependencies among the nodes. Moreover, note that (35) involves nonadditive nonlinear interactions among the nodes, which is different from the VAR assumption in (1). The performance of the RFNL-TIRSO and the PDIS [49] algorithms are compared in this section, whereas TIRSO is omitted since the algorithm implementation assumes $P > 1$. To ensure a fair comparison, we follow exactly the same experiment set up as in [49], in which, the performance is measured using the *edge identification error rate* (EIER), defined as $EIER = \frac{\|\mathbf{A}-\hat{\mathbf{A}}\|_0}{N(N-1)} \times 100$, where $\mathbf{A}$ is the true dependency

matrix and $\hat{\mathbf{A}}$ is the estimated dependency matrix. For RFNL-TIRSO, $\hat{\mathbf{A}}$ is computed using $\widehat{b}^{(1)}_{n,n'}$. The hyperparameters are tuned heuristically to obtain minimum EEIR resulting in a setting $(\sigma_n, \lambda, a_t) = (1, .3, 1/(t\Lambda_{max}(\boldsymbol{\phi}[t])))$. The estimated and true binary adjacency matrix (excluding self-dependencies) are shown in Fig. 5, and the EIER till $t = 1750$ is plotted in Fig. 6. We remark that although the PDIS algorithm is designed by assuming non-additive nonlinear interactions, its performance lags behind the proposed RFNL-TIRSO algorithm, which assumes additive nonlinearities. This is because the RFNL-TIRSO algorithm employs an RLS loss function, which results in an improved convergence speed compared to the LMS loss used in PDIS.



(a) (b) (c)

Figure 5: Lorenz graph detection using RFNL-TIRSO: (a) True Binary dependency, (b) Estimated dependency, (c) Binary estimated dependency by stetting threshold as 0.5.

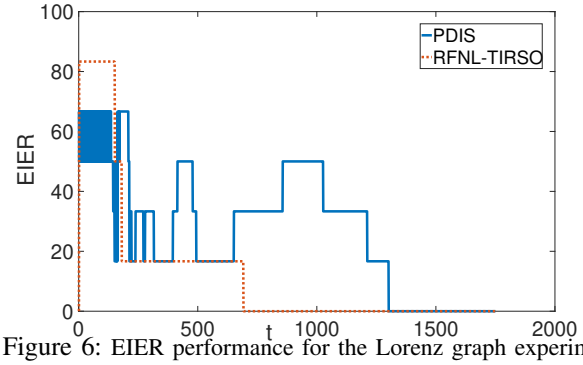


Figure 6: EIER performance for the Lorenz graph experiment.

3) Numerical Evaluation of Dynamic Regret: A theoretical bound of the dynamic regret $\boldsymbol{R}_n[T] = \boldsymbol{R}^{\text{rf}}_n[T] + \boldsymbol{\xi}_n[T]$ has been derived in Section IV-A. In this section, using experiments conducted on synthetic data, we numerically compute the dynamic regret of RFNL-TIRSO w.r.t. the optimal cost in the RF space, defined as $\boldsymbol{R}^{\text{rf}}_n[t] = \sum_{\tau=P}^{t-1} (h^n_\tau(\boldsymbol{\alpha}_n[\tau]) - h^n_\tau(\boldsymbol{\alpha}^*_n[\tau]))$, for $t = 1, \ldots, 1000$. This allows validating experimentally our theoretical results. Here, $\boldsymbol{\alpha}_n[\tau]$ is the RF coefficient estimated using RFNL-TIRSO at time τ, and $\boldsymbol{\alpha}^*_n[\tau]$ is the optimum RF coefficient, computed using a standard gradient descent algorithm until convergence. We remark that the estimation of $\boldsymbol{\alpha}^*_n[\tau]$ involves very high computational complexity compared to that of $\boldsymbol{\alpha}_n[\tau]$. In Fig. 7, we plot $\boldsymbol{R}^{\text{rf}}_n[t]$ and its rate of change w.r.t. time $\boldsymbol{R}^{\text{rf}}_n[t]/t$. In this experiment, we used the same data generation mechanism involving the nonlinear dependencies g_1 and g_2, as explained in Section V-A1, having topology change points at $t = 250$ and $t = 500$. Figure 7 shows that $\mathbf{R}^{\text{rf}}[t]$ is sublinear w.r.t. t and $\mathbf{R}^{\text{rf}}[t]/t$ is negligibly small, which is in agreement with the theoretical results stated in **Theorem 1**. We note that a numerical evaluation of the second component of the dynamic regret $\boldsymbol{\xi}_n[t]$ is a daunting, complex process since it involves finding the optimal parameters in a high dimensional RKHS. However, as shown in (67) we remark that $\boldsymbol{\xi}_n[t]/t$ is theoretically bounded by the value $\epsilon L_f C$, where ϵ is a user-controlled parameter. The value of $\boldsymbol{\xi}_n[t]/t$ can be made small to obtain a dynamic regret $\boldsymbol{R}_n[t]/t$ upper bounded by a small constant for $t \to \infty$.

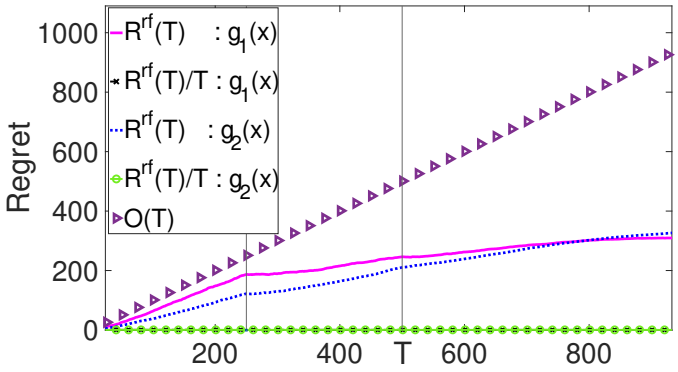


Figure 7: Regret w.r.t. optimal cost function in RF space. Vertical lines indicate the topology change points.

B. Experiments using Real Data Sets

1) Oil and Gas Platform Data: This section is dedicated to experiments using real data collected from Lundin's Offshore Oil and Gas (O and G) platform Edvard-Grieg[2]. We collected multivariate time series data from 24 nodes (numbered as $n = 1, 2, \ldots, 24.$) of the plant corresponding to various temperature (T), pressure (P), and oil-level (L) sensors. The sensors are placed in the separators of decantation tanks separating oil, gas and water. The time series are obtained by uniformly sampling the sensor readings with a sampling rate of 5 seconds. We assume that the hidden logic dependencies are present in the network due to the various existing physical connections and control actuators. The data obtained from the sensors are preprocessed by normalizing them to zero mean unit variance signals.

The dependencies are learned using RFNL-TIRSO ($D = 10$), RFNL-TISO, and TIRSO by assuming a VAR model of order P=12. A Gaussian kernel having a variance of 1 is used in all the experiments with hyperparameter setting $\lambda = 0.1$ and step size a_t=$1/\Lambda_{max}(\boldsymbol{\phi}[t])$ (tuned to obtain minimum NMSE). The estimated dependencies are visualized in Fig. 8 using the ℓ_2 norms $\|\boldsymbol{\alpha}_{n,n'}[t]\|_2$. RFNL-TIRSO identifies interpretable connections; for instance, two pressure sensors in the same separator are connected, and the oil level in separator-1 is connected to the pressure variation in separator-2. Further, as expected, most of the identified interactions are local (e.g., interactions inside a separator), with very few long-distance interactions (e.g., interactions between two separators). The strong local interactions among variables such as temperature, pressure, and oil level inside a container are directly linked to fluid dynamics of the oil and gas in the closed chamber as dictated by the differential equations governing these variables [52]. However, various control mechanisms governing the whole oil and gas platform and the physical connections

[2]https://www.lundin-energy.com/

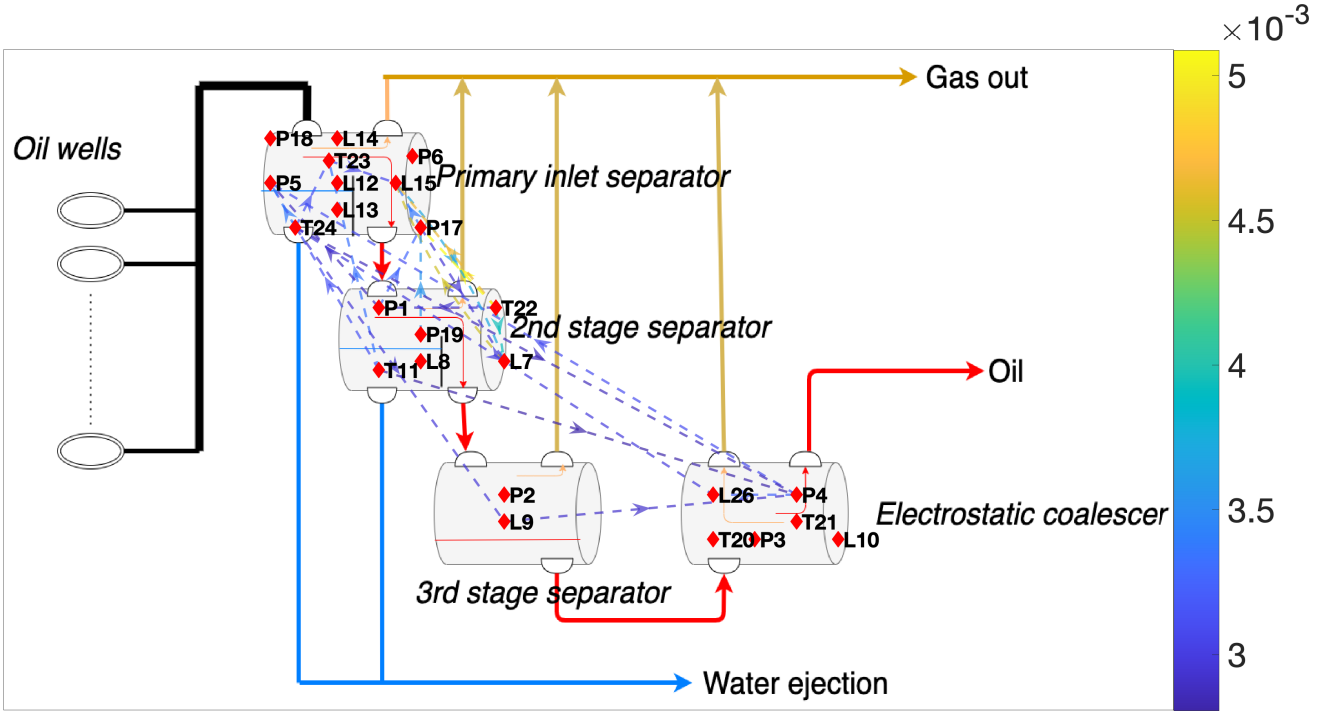


Figure 8: Topology estimated using RFNL-TIRSO for Oil and Gas platform. Temperature, pressure, and level sensors are denoted by the labels 'T','P', and 'L' in the node index, respectively.

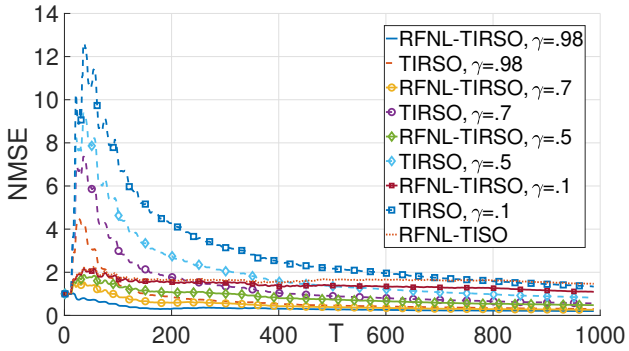


Figure 9: NMSE comparison: data from the Oil and Gas platform.

across different chambers can also cause some longer-distance non-trivial interactions, although they will not typically be as predominant as the local interactions. For instance, the primary inlet separator and the electrostatic coalescer can interact despite not being physically connected. When there are changes in the oil level within the coalescer, it can affect the head of the system, leading to changes in the pressure and oil level within the primary inlet separator that operates based on gravity.

We wish to note that the estimated dependencies can be interpreted as an abstract graph representation of various physics-based equations describing the space-temporal variation of the signals. Ground truth dependencies are not available in this experiment, and evaluating the estimated graph using the underlying differential physics-based equations governing the space-time system is a tedious procedure that is beyond the scope of this study. However, we demonstrate the ability of the algorithms to learn the dependencies based on the accuracy of time series forecasting using the learned VAR model. A good prediction accuracy implies that the estimated dependencies are close to the underlying unknown real dependencies. As a side note, we highlight that time series forecasting is a challenging problem having enormous applications in various fields such as financial engineering, traffic forecast, sensor networks, among others. The prediction accuracy is computed using normalized mean squared error (NMSE):

$$\text{NMSE (T)} = \frac{\sum_{t=1}^{T}(y_n[t+t_{step}] - \hat{y}_n[t+t_{step}])^2}{\sum_{t=1}^{T}(y_n[t+t_{step}])^2}, \quad (36)$$

where $\hat{y}_n[t+t_{step}]$ is the estimate of the time series generated by the n^{th} node at time instant $t+t_{step}$ based on the VAR model learned at time t. Figure 9 shows the NMSE of the estimated signals corresponding to a particular sensor $n=8$ using various algorithms. We discard the PDIS algorithm in this experiment since it is not designed for signal prediction. NMSE is calculated according to (36) with $t_{step}=12$, which refers to one minute ahead prediction. For RFNL-TIRSO and TIRSO, we conduct the experiments by varying the forgetting factor $\gamma \in \{0.1, 0.5, 0.7, 0.98\}$. We note that the best NMSE of the RFNL-TIRSO algorithm is obtained at $\gamma = .98$, and it outperforms all the competitors. It is interesting to observe that as γ reduces, the performance of RFNL-TIRSO becomes close to RFNL-TISO, as expected from (17). Additionally, we plot the dynamic regret and cumulative variation of the optimal parameter estimates in Fig. 10, which shows that our algorithm is able to track the topology even if the optimal topology is changing.

In section Section IV, we show that the RFNL-TIRSO converges if the learning rate is less than $1/L$, where L is the upper bound of $\Lambda_{max}(\boldsymbol{\phi}[T])$. The performance of RFNL-TIRSO under various learning rates is shown in Fig. 11. Intu-

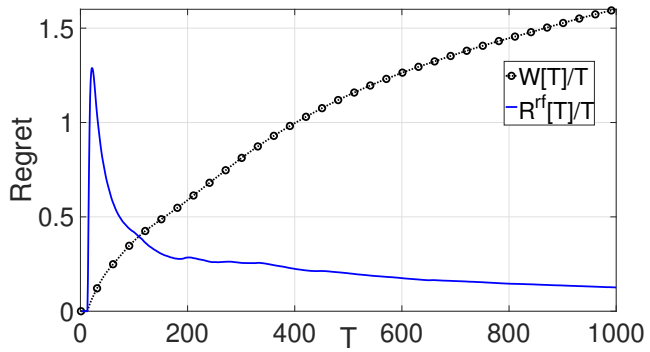


Figure 10: Rate of change of regret and path length: data from Edvard-Grieg Oil and Gas platform.

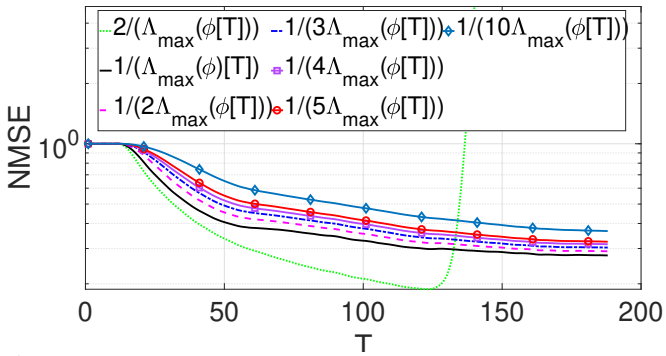


Figure 11: NMSE with different learning rate a_t: data from Edvard-Grieg Oil and Gas platform.

itively as the learning rate increases, RFNL-TIRSO converges faster; and when the learning rate is increased beyond $1/L$, convergence is not guaranteed, as evidenced in Fig. 11. Note that if the data has high variance, the value of $\Lambda_{max}(\boldsymbol{\phi}[T])$ will be obviously high, necessitating the use of a lower learning rate to ensure the algorithm convergence.

2) Epileptic data set: The dataset used for this experiment [50] is collected from the Children's Hospital Boston, and it consists of EEG recordings from two pediatric subjects with intractable seizures, labelled as P1 (age-11, gender-female) and P2 (age-10, gender-female). Subjects were monitored for several days following the withdrawal of anti-seizure medication to characterize their seizures and assess their candidacy for surgical intervention. The electrode positions and the nomenclature used during the EEG recordings were based on the well-known *International 10-20 system* standard. All signals were sampled at 64 samples per second, and there is a total of 23 Channels: FP1:F7, F7:T7, T7:P7, P7:O1, FP1:F3, F3:C3, C3:P3, P3:O1, FP2:F4, F4:C4, C4:P4, P4:O2, FP2:F8, F8:T8, T8:P8, P8:O2, FZ:CZ, CZ:PZ, P7:T7, T7:FT9, FT9:FT10, FT10:T8, and 2T8:P8, which measures the potential difference between the corresponding electrodes.

The estimated brain topology using RFNL-TIRSO ($P = 2, D = 20$) at various time instants (before seizure, during seizure, after seizure), visualized using the ℓ_2 norms $\|\boldsymbol{\alpha}_{n,n'}[t]\|_2$, are shown in Fig. 12. It is observed that the estimated topologies before and after the seizure are very similar, with connections concentrated across certain brain regions. However, during the seizure, the topologies get more disrupted, which agrees with the observations in [53]. This disruption can be attributed to an increase in pathogenic neural discharge during the seizure [54].

The brain can be divided into several regions, namely, temporal, frontal, occipital, parietal and central. Epilepsies are generally classified according to the region of the brain where they originate, with common classifications including temporal lobe (TL) epilepsy and frontal lobe (FL) epilepsy [55]. In TL epilepsy, more inter-region connections will originate from the temporal region, whereas in FL epilepsy, such connections are originated from the frontal region. To showcase this, we next present an experiment with the brain data of P1 and P2, respectively belonging to the TL and FL epilepsy categories [56]. To measure the activity level of different brain regions, we group all the channels connected to the 'temporal' region into one group (group-T) and the 'frontal' region into another group (group-F). Note that all the connections between the 'frontal' and the 'temporal' regions are excluded in this experiment. We define the activation level of a group as the sum of the degrees of all the nodes belonging to the group divided by the total number of nodes present in the group, where the degree of a node refers to the total number of edges connected to the node. The activation level of each group for P1 and P2 are shown in Fig. 13a and Fig. 13b, respectively. From the figures, it is observed that for P1 and P2, the activation levels of group-T and group-F, respectively, spike first, and then the activation spreads across the other brain region. These observations align with the characteristics of TL and FL epilepsies.

VI. Conclusion

An online nonlinear topology identification algorithm termed RFNL-TIRSO is proposed in this paper. The multivariate time series data received in the sequential form are processed online to estimate time-varying nonlinear dependencies. It has been proven that RFNL-TIRSO follows a sublinear dynamic regret, which guarantees the tracking capability of the algorithm in dynamic environments. The performance of RFNL-TIRSO is evaluated using real and synthetic data sets, and the algorithm outperforms the state-of-the-art online topology estimation methods.

Acknowledgment

The authors wish to thank Dr. Bakht Zaman and Dr. Mircea Moscu for providing codes of the TIRSO algorithm [6] and the PDIS algorithm [49].

Appendix A
Proof of Theorem 1

In this section, we derive a theoretical upper bound for $\boldsymbol{R}_n^{\text{rf}}(T)$. Since the function h_t^n is convex

$$\begin{aligned}\boldsymbol{R}_n^{\text{rf}}(T) &= \sum_{t=P}^{T-1} \left[h_t^n(\boldsymbol{\alpha}_n[t]) - h_t^n(\boldsymbol{\alpha}_n^*[t])\right] \\ &\leq \sum_{t=P}^{T-1} \nabla h_t^n(\boldsymbol{\alpha}_n[t])^\top (\boldsymbol{\alpha}_n[t] - \boldsymbol{\alpha}_n^*[t]).\end{aligned} \tag{37}$$

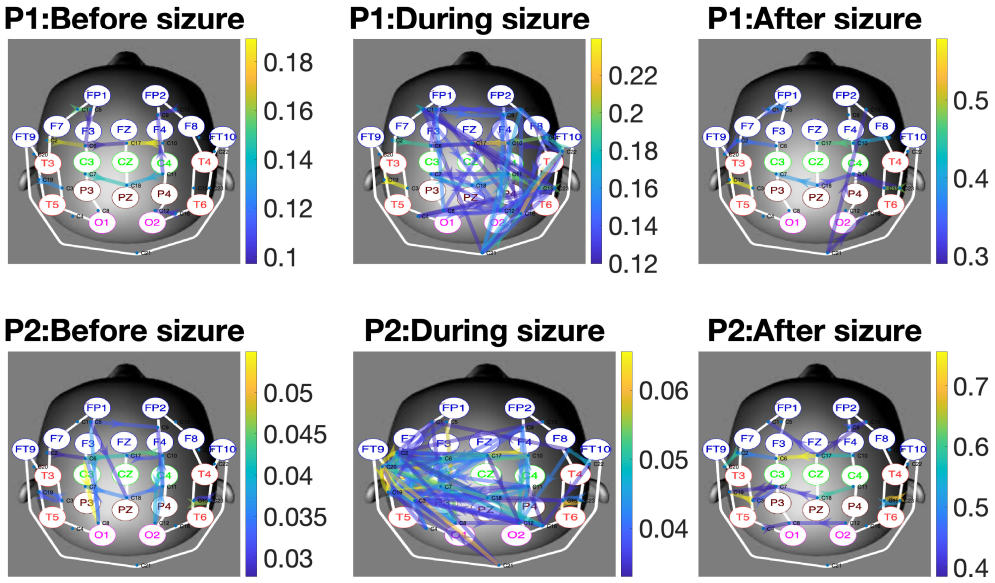
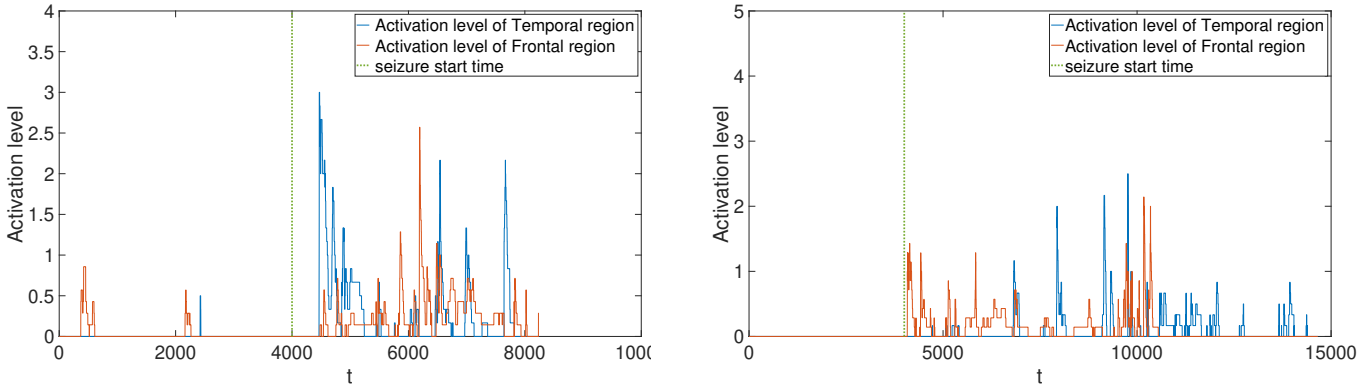


Figure 12: Estimated brain topology for the subjects P1 and P2 during various stages of seizure



(a) Subject: P1, Category: Temporal Lobe Epilepsy

(b) Subject: P2, Category: Frontal Lobe Epilepsy

Figure 13: Activation levels in 'T' and 'F' regions of the brain.

Apply Cauchy-Schwarz inequality on right hand side to get

$$\begin{aligned}\boldsymbol{R}_n^{\text{rf}}(T) &= \sum_{t=P}^{T-1}\Big[h_t^n(\boldsymbol{\alpha}_n[t]) - h_t^n(\boldsymbol{\alpha}_n^*[t])\Big] \\ &\leq \sum_{t=P}^{T-1} \|\nabla h_t^n(\boldsymbol{\alpha}_n[t])\|_2 \ \|\boldsymbol{\alpha}_n[t] - \boldsymbol{\alpha}_n^*[t]\|_2. \end{aligned} \tag{38}$$

The optimality gap of any proximal gradient descent algorithm with an objective function having 1) a strongly convex and Lipschitz smooth loss function and 2) a Lipschitz continuous regularizer is derived in [36]. We can show that RFNL-TIRSO is a proximal gradient descent algorithm by following the proofs provided in [6]. Hence, the cumulative optimality gap is bounded as

$$\sum_{t=P}^{T-1} \|\boldsymbol{\alpha}_n[t] - \boldsymbol{\alpha}_n^*[t]\|_2 = \|\boldsymbol{\alpha}_n^*[P]\|_2 + \boldsymbol{W}_n(T), \tag{39}$$

where $\boldsymbol{W}_n(T) = \sum_{t=P}^{T-1} \|\boldsymbol{\alpha}_n^*[t] - \boldsymbol{\alpha}_n^*[t-1]\|_2$ is the path length, which is a measure of the cumulative variation of the optimality gap. Next, we bound for the term $\|\nabla h_t^n(\boldsymbol{\alpha}_n[t])\|_2$ in (38).

Lemma 1. *Under the assumptions A1, A3 and A4,*

$$\|\nabla h_t^n(\boldsymbol{\alpha}_n[t])\|_2 \leq \left(\left(1 + \frac{L}{\rho_l}\right)\sqrt{2PNDB_y} + \lambda\sqrt{PN}\right).$$

Proof: The cost function consists of a differentiable loss function $\tilde{\ell}_t^n$ and a non-differentiable regularizer $\boldsymbol{\omega}^n$. We introduce the notation $\boldsymbol{u}^n$ to denote a subgradient of the regularizer $\boldsymbol{\omega}^n(\boldsymbol{\alpha}_n[t])$. The gradient of the entire cost function can be bounded by bounding the gradient of these two terms:

$$\|\nabla h_t^n(\boldsymbol{\alpha}_n[t])\|_2 \leq \|\nabla \tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])\|_2 + \|\boldsymbol{u}^n\|_2. \tag{40}$$

The term $\|\nabla \tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])\|_2$ is bounded in **Lemma 1.2** using **Lemma 1.1**, and the term $\|\boldsymbol{u}^n\|_2$ is bounded in **Lemma 1.3**.

Lemma 1.1. *Under assumptions A1 and A3*

$$\|\boldsymbol{\alpha}_n[t+1]\|_2 \leq (1 - a_t\rho_l)\|\boldsymbol{\alpha}_n[t]\|_2 + a_t\sqrt{2PNDB_y}.$$

Proof: From **Lemma 7** in [6] we have,

$$\|\boldsymbol{\alpha}_n[t+1]\|_2 \leq (1 - a_t\rho_l)\|\boldsymbol{\alpha}_n[t]\|_2 + a_t\|\boldsymbol{r}_n[t]\|_2. \tag{41}$$

Using (21), we can bound $\|\boldsymbol{r}_n[t]\|_2$ as

$$\begin{aligned}\|\boldsymbol{r}_n[t]\|_2 &= \|\mu \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau] \boldsymbol{z_v}(\tau)\|_2 \\ &\leq \mu \| \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau] \mathbf{1}_{2PND}\|_2 \quad (42)\\ &\leq \mu \sqrt{2PNDB_y} \gamma^t \sum_{\tau=P}^{t} (\frac{1}{\gamma})^\tau \quad (43)\\ &= \sqrt{2PNDB_y}(1-\gamma^{t-P+1}) \quad (44)\\ &\leq \sqrt{2PNDB_y}. \quad (45)\end{aligned}$$

Inequality (42) is obtained by replacing the RF vector (sinusoidal components) with an all-one vector having a higher norm, (43) is obtained using the assumption A1, (44) follows from $u = 1-\gamma$, and (45) follows from $\gamma \leq 1$. **Lemma 1.1** is proved by substituting (45) in (41).

Lemma 1.2. *Under assumptions A1, A3, and A4, the RFNL-TIRSO algorithm with step size parameter $a_t = \frac{1}{L}$ satisfies*

$$\|\nabla \tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])\|_2 \leq \Big(1 + \frac{L}{\rho_l}\Big)\sqrt{2PNDB_y}.$$

Proof: Invoke **Lemma 1.1**, set $a_t = a$, and let $\delta = (1 - a\rho_l)$ and $0 \leq \delta \leq 1$, to get

$$\|\boldsymbol{\alpha}_n[t+1]\|_2 \leq \delta \|\boldsymbol{\alpha}_n[t]\|_2 + a_t\sqrt{2PNDB_y} \quad (46)$$

The bound of $\|\boldsymbol{\alpha}_n[t+1]\|_2$ in terms of the norm of the initial estimate $\|\boldsymbol{\alpha}_n[P]\|_2$ is obtained by $t-P+1$ recursion of (46):

$$\begin{aligned}\|\boldsymbol{\alpha}_n[t+1]\|_2 &\leq \delta^{t-P+1}\|\boldsymbol{\alpha}_n[P]\|_2 + a\sqrt{2PNDB_y}\sum_{i=0}^{t-P}\delta^i \\ &= \frac{a\sqrt{2PNDB_y}(1-\delta^{t-P+1})}{1-\delta} \quad (47)\\ &\leq \frac{a\sqrt{2PNDB_y}}{1-(1-a\rho_l)}) = \frac{1}{\rho_l}\sqrt{2PNDB_y} \quad (48)\end{aligned}$$

In (47), we assumed that the RF coefficients are initialized with zeros, i.e., $\boldsymbol{\alpha}_n[P] = \mathbf{0}_{2PND}$.

Using (48) and (45), we can bound gradient:

$$\begin{aligned}\|\nabla \tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])\|_2 &= \|\boldsymbol{\phi}[t]\boldsymbol{\alpha}_n[t] - \boldsymbol{r}_n[t]\|_2 \text{ (from (22))} \\ &\leq \|\boldsymbol{\phi}[t]\boldsymbol{\alpha}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2 \\ &\leq \Lambda_{max}(\boldsymbol{\phi}[t])\|\boldsymbol{\alpha}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2 \quad (49)\\ &= L\frac{\sqrt{2PNDB_y}}{\rho_l} + \sqrt{2PNDB_y} \quad (50)\\ &\leq \Big(1+\frac{L}{\rho_l}\Big)\sqrt{2PNDB_y} \quad (51)\end{aligned}$$

Inequality (49) holds since spectral norm of $\boldsymbol{\phi}[t] = \Lambda_{max}(\boldsymbol{\phi}[t])$, whereas (50) is obtained by combining the Assumption A4, (48), and (45). Next, we bound $\|\boldsymbol{u}^n\|_2$.

Lemma 1.3. *The norm of a subgradient of the regularizer can be bounded as*

$$\|\boldsymbol{u}^n\|_2 \leq \lambda\sqrt{PN}.$$

Proof: To prove **Lemma 1.3**, we apply **Lemma 2.6** from [4] which states that every subgradient of $\omega^n(.)$ is bounded by its Lipschitz continuity parameter L_{ω^n}. In the following, we show that $L_{\omega^n} = \lambda\sqrt{PN}$.

Lipschitz continuity of ω^n means there exists $L_{\omega^n} > 0$ such that

$$|\omega^n(\boldsymbol{a}) - \omega^n(\boldsymbol{b})| \leq L_{\omega^n}\|\boldsymbol{a}-\boldsymbol{b}\|_2 \quad (52)$$

for all real $\boldsymbol{a}$ and $\boldsymbol{b}$. From the group-Lasso regularizer, we have

$$\omega^n(\boldsymbol{x_n}) = \lambda \sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{x}_{n,n'}^{(p)}\|_2. \quad (53)$$

Expanding the left-hand side of (52) using (53) yields

$$\begin{aligned}&|\omega^n(\boldsymbol{a}_n) - \omega^n(\boldsymbol{b}_n)| \quad (54)\\ &= \lambda\Big|\sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{b}_{n,n'}^{(p)}\|_2\Big| \quad (55)\\ &= \lambda\Big|\sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \|\boldsymbol{b}_{n,n'}^{(p)}\|_2\Big| \quad (56)\\ &\leq \lambda\sum_{n'=1}^{N}\sum_{p=1}^{P}\Big|\|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \|\boldsymbol{b}_{n,n'}^{(p)}\|_2\Big| \quad (57)\\ &\leq \lambda\sum_{n'=1}^{N}\sum_{p=1}^{P}\|\boldsymbol{a}_{n,n'}^{(p)} - \boldsymbol{b}_{n,n'}^{(p)}\|_2 \quad (58)\\ &\leq \lambda\sqrt{PN}\|\boldsymbol{a}_n - \boldsymbol{b}_n\|_2. \quad (59)\end{aligned}$$

In the above derivation, inequality (57) follows from the triangle inequality, inequality (58) from the reverse triangle inequality and (59) from the basic inequality $\|q\|_1 \leq \sqrt{M}\|q\|_2,\ q \in R^M$. From (59), we obtain the required Lipschitz parameter to be $\lambda\sqrt{PN}$.

Substitute the bounds of $\|\nabla l_t^n(\boldsymbol{\alpha}_n[t])\|_2$ given by **Lemma 1.2** and $\|\boldsymbol{u}^n\|_2$ given by **Lemma 1.3** in (40) to complete the proof of **Lemma 1**. Finally, the proof of **Theorem 1** can be completed by substituting **Lemma 1** and (39) in (38).

APPENDIX B
PROOF OF THEOREM 2

The cumulative approximation error due to the RF approximation is

$$\boldsymbol{\xi}_n[T] \leq \Big|\sum_{t=P}^{T-1}\Big[h_t^n(\boldsymbol{\alpha}_n^*[t]) - h_t^n(\boldsymbol{\beta}_n^*[t])\Big]\Big|. \quad (60)$$

Using the triangle inequality,

$$\begin{aligned}\boldsymbol{\xi}_n[T] &\leq \sum_{t=P}^{T-1}\Big|h_t^n(\boldsymbol{\alpha}_n^*[t]) - h_t^n(\boldsymbol{\beta}_n^*[t])\Big| \\ &\leq \sum_{t=P}^{T-1} L_h\Bigg|\sum_{n'=1}^{N}\sum_{p=1}^{P}\sum_{t'=P}^{t+p-1}\beta_{n,n',(t'-p)}^{(p)*}\boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t)^\top \boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t') \\ &\qquad - \beta_{n,n',(t'-p)}^{(p)*}k_{n'}^{(p)}\left(y_{n'}[t-p], y_{n'}[t'-p]\right)\Bigg| \quad (61)\end{aligned}$$

$$\leq \sum_{t=P}^{T-1} L_h \sum_{n'=1}^{N} \sum_{p=1}^{P} \sum_{t'=p}^{t+p-1} \left| \beta^{(p)*}_{n,n',(t'-p)} \right| \times \left| \boldsymbol{z}^{(p)}_{\boldsymbol{v},n'}(t)^{\top} \boldsymbol{z}^{(p)}_{\boldsymbol{v},n'}(t) - k^{(p)}_{n'}(y_{n'}[t-p], y_{n'}[t'-p]) \right|. \quad (62)$$

Inequality (61) is obtained from the Lipschitz continuity of the cost function ($L_h > 0$ is the Lipschitz continuity parameter) and (62) follows from Cauchy-Schwarz inequality. As shown in [21], it can be proved that for a given shift-invariant kernel $k^{(p)}_{n'}$ (assumption A2), the approximation error due to the random Fourier approximation is bounded by

$$\sup_{y_n(t)} \left| \boldsymbol{z}^{(p)}_{\boldsymbol{v},n'}(t)^{\top} \boldsymbol{z}^{(p)}_{\boldsymbol{v},n'}(t) - k^{(p)}_{n'}\left(y_{n'}[t-p], y_{n'}[t'-p]\right) \right| \leq \epsilon^{p}_{n'} \quad (63)$$

with a probability given by $1 - 2^8(\sigma^p_{n'}/\epsilon^p_{n'})^2 \exp(-D\epsilon^p_{n'}/12)$. Here, $\epsilon^p_{n'} \geq 0$ is a constant and $\sigma^p_{n'}$ is the variance of the random feature vector norm. Using (63),

$$\boldsymbol{\xi}_n[T] \leq \sum_{t=P}^{T-1} L_h \sum_{n'=1}^{N} \sum_{p=1}^{P} \sum_{t'=P}^{t+p-1} \left| \beta^{(p)*}_{n,n',(t'-p)} \right| \epsilon^{p}_{n'}. \quad (64)$$

Let $\epsilon = \max \epsilon^p_{n'}$, which leads to

$$\boldsymbol{\xi}(T) \leq \sum_{t=P}^{T-1} L_h \epsilon \sum_{n'=1}^{N} \sum_{p=1}^{P} \sum_{t'=P}^{t+p-1} \left| \beta^{(p)*}_{n,n',(t'-p)} \right| \quad (65)$$

$$\leq \sum_{t=P}^{T-1} \epsilon L_h C \quad (66)$$

$$\leq \epsilon L_h T C, \quad (67)$$

where C is a constant and (66) follows from the assumption A1: since $y_n(t)$ is bounded, the optimal parameters should also be bounded.

REFERENCES

[1] F. Youping, L. Jingjiao, and Z. Dai, "A method for identifying critical elements of a cyber-physical system under data attack," *IEEE Access*, vol. 6, 2018.

[2] H. Weiyu, G. Leah, W. Nicholas, G. Scott, B. Danielle, and R. Alejandro, "Graph frequency analysis of brain signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 7, 2016.

[3] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, 2022.

[4] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, 2012.

[5] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," 2020.

[6] B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Transactions on Signal Processing*, vol. 69, pp. 210–225, 2021.

[7] N. Alberto, C. Mario, I. Elvin, and L. Geert, "Online time-varying topology identification via prediction-correction algorithms," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[8] M. Singh and V. Kekatos, "Optimal scheduling of water distribution systems," *IEEE Transactions on Control of Network Systems*, 2019.

[9] C. Stam, *Nonlinear brain dynamics*. Nova Biomedical, 2006.

[10] Y. Shen, G. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Transactions on Signal Processing*, vol. 67, pp. 5325–5339, 2019.

[11] N. Meike, B. Doina, and S. Christin, "Causal discovery with attention-based convolutional neural networks," *Machine Learning and Knowledge Extraction*, vol. 1, 2019.

[12] L. Lopez-Ramos, K. Roy, and B. Beferull-Lozano, "Explainable nonlinear modelling of multiple time series with invertible neural networks," *INTAP*, 2021.

[13] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, "Neural granger causality for nonlinear time series," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[14] J. Lu, s. C.H. Hoi, J. Wang, P. Zhao, and Z. Liu, "Large scale online kernel learning," *Journal of Machine Learning Research*, 2016.

[15] L. Zhang, J. Yi, R. Jin, M. Lin, and X. He, "Online kernel learning with a near optimal sparsity bound," 2013.

[16] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online non-linear topology identification from graph-connected time series," in *2021 IEEE Data Science and Learning Workshop (DSLW)*, 2021, pp. 1–6.

[17] V. Michel and F. Damien, "The curse of dimensionality in data mining and time series prediction," in *Computational Intelligence and Bioinspired Systems*. Springer Berlin Heidelberg, 2005.

[18] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," in *2017 IEEE ICASSP*, March 2017, pp. 4671–4675.

[19] Y. Shen and G. B. Giannakis, "Online identification of directional graph topologies capturing dynamic and nonlinear dependencies†," in *2018 IEEE Data Science Workshop (DSW)*, 2018, pp. 195–199.

[20] M. Moscu, R. A. Borsoi, C. Richard, and J.-C. M. Bermudez, "Graph topology inference with derivative-reproducing property in rkhs: Algorithm and convergence analysis," *IEEE Transactions on Signal and Information Processing over Networks*, 2022.

[21] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NeurIPS*, ser. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., 2007, p. 1177–1184.

[22] J. Lu, S. Hoi, J. Wang, P. Zhao, and Z. Liu, "Large scale online kernel learning," *Journal of Machine Learning Research*, 2016.

[23] T. Nguyen, T. Le, H. Bui, and D. Phung, "Large-scale online kernel learning with random feature reparameterization," in *Proceedings of the Twenty-Sixth International Joint Conference on AI, IJCAI-17*.

[24] Y. Shen, T. Chen, and G. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *J. Mach. Learn. Res.*, vol. 20, no. 1, p. 773–808, Jan. 2019.

[25] G. Yehudai and O. Shamir, "On the power and limitations of random features for understanding neural networks," *arXiv:1904.00687*, 2020.

[26] R. Sato, M. Yamada, and H. Kashima, "Random features strengthen graph neural networks," *arXiv:2002.03155*, 2021.

[27] B. Can and H. Ozkan, "A neural network approach for online nonlinear neyman-pearson classification," *IEEE Access*, vol. 8, pp. 210 234–210 250, 2020.

[28] F. Porikli and H. Ozkan, "Data driven frequency mapping for computationally scalable object detection," *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 30–35, 2011.

[29] H. Ozkan, N. D. Vanli, and S. S. Kozat, "Online classification via self-organizing space partitioning," *IEEE Transactions on Signal Processing*, 2016.

[30] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in non-stationary environments: A survey," *IEEE Computational Intelligence Magazine*, 2015.

[31] R. Money, J. Krishnan, and B. Beferull-Lozano, "Random feature approximation for online nonlinear graph topology identification," *IEEE MLSP*, 2021.

[32] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, "Dynamic regret of strongly adaptive methods," *ICML*, 2018.

[33] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 7195–7201, 2016.

[34] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[35] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online Optimization : Competing with Dynamic Comparators," pp. 398–406, 09–12 May 2015.

[36] R. Dixit, A. S. Bedi, R. Tripathi, and K. Rajawat, "Online learning with inexact proximal online gradient descent algorithms," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1338–1352, 2019.

[37] A. Chakraborty, K. Rajawat, and A. Koppel, "Sparse representations of positive functions via first- and second-order pseudo-mirror descent," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3148–3164, 2022.

[38] A. S. Bedi, A. Koppel, K. Rajawat, and B. M. Sadler, "Trading dynamic regret for model complexity in nonstationary nonparametric optimization," *2020 American Control Conference (ACC)*, pp. 321–326, 2020.

[39] D. Calandriello, A. Lazaric, and M. Valko, "Second-order kernel online convex optimization with adaptive sketching," *International Conference on Machine Learning*, 2017.

[40] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[41] G. Wahba, *Spline Models for Observational Data*. SIAM Press, Society for Industrial and Applied Mathematics, 1990.

[42] B. Olkopf, R. Herbrich, A. Smola, and R. Williamson, "A generalized representer theorem," *Computational Learning Theory*, vol. 42, 06 2000.

[43] S. Bochner, "Lectures on fourier integrals," *Princeton University press*, vol. 42, 1959.

[44] L. Lopez-Ramos, D. Romero, B. Zaman, and B. Beferull-Lozano, "Dynamic network identification from non-stationary vector autoregressive time series," in *2018 IEEE GlobalSIP*, Nov 2018, pp. 773–777.

[45] J. Duchi, S. Shwartz, and A. Tewari, "Composite objective mirror descent," in *COLT'10*, 2010, pp. 14–26.

[46] M. Gutmann and J. Hirayama, "Bregman divergence as general framework to estimate unnormalized statistical models," in *Proceedings of UAI*, ser. UAI'11. Arlington, Virginia, USA: AUAI Press, 2011.

[47] A. Puig, A. Wiesel, and A. Hero, "A multidimensional shrinkage-thresholding operator," *Proceedings of the 15th Workshop on Statistical Signal Processing*, vol. 18, pp. 113 – 116, 10 2009.

[48] M. A. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on Machine Learning*, 2003.

[49] M. Moscu, R. Borsoi, and C. Richard, "Online kernel-based graph topology identification with partial-derivative-imposed sparsity," *2020 28th European Signal Processing Conference (EUSIPCO)*.

[50] A. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment." *PhD Thesis, MIT*, 2009.

[51] E. N. Lorenz, "Computational chaos-a prelude to computational instability," *Physica D: Nonlinear Phenomena*, 1989.

[52] S. Song, X. Liu, C. Li, Z. Li, S. Zhang, W. Wu, B. Shi, Q. Kang, H. Wu, and J. Gong, "Dynamic simulator for three-phase gravity separators in oil production facilities," *ACS Omega*, vol. 8, no. 6, pp. 6078–6089, 2023.

[53] Y. Hu, Q. Zhang, R. Li, T. Potter, and Y. Zhang, "Graph-based brain network analysis in epilepsy: an EEG study," *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2019.

[54] F. Pittau, F. Fahoum, R. Zelmann, F. Dubeau, and J. Gotman, "Negative bold response to interictal epileptic discharges in focal epilepsy," *Brain Topogr*, 2013.

[55] M. Manford, D. Fish, and S. Shorvon, "An analysis of clinical seizure patterns and their localizing value in frontal and temporal lobe epilepsies," *Brain : a journal of neurology*, vol. 1, 1996.

[56] N. Saadat and P. Hossein, "Epileptic seizure onset detection algorithm using dynamic cascade feed-forward neural networks," in *2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation*, 2011, pp. 196–199.