Sparse Online Learning With Kernels Using Random Features for Estimating Nonlinear Dynamic Graphs

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

***Abstract*—Online topology estimation of graph-connected time series is challenging in practice, particularly because the dependencies between the time series in many real-world scenarios are nonlinear. To address this challenge, we introduce a novel kernel-based algorithm for online graph topology estimation. Our proposed algorithm also performs a Fourier-based random feature approximation to tackle the curse of dimensionality associated with 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 our algorithm and prove that it can achieve sublinear dynamic regret under certain reasonable assumptions. In experiments conducted on both real and synthetic data, our method outperforms existing state-of-the-art competitors.**

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

I. Introduction

MULTIVARIATE time series data is generated by various real-world networks, including large-scale cyber-physical systems (CPS), financial networks and brain networks. In such systems, the time series are interdependent, and the dependencies can be represented as graphs; in other words, 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 provide valuable insights and better outcomes in many application fields. For instance, it can aid in developing better control actions in CPS [1], provide explainable analysis in brain networks [2], and improve the accuracy of forecasts in financial time series [3].

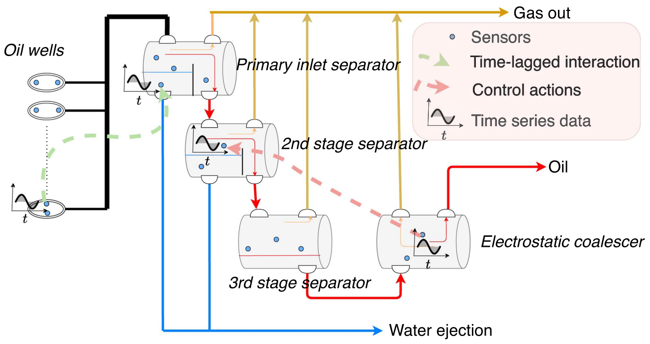


Fig. 1. Schematic of processing stages in an oil and gas platform

Real-world networks often exhibit time-delayed and directed dependencies among their components. For example, consider an oil and gas processing platform with multiple wells and separators, which uses hundreds of sensors and actuators to extract raw oil and separate it into oil, water and gas (Fig. 1). If an event occurs in a well, its impact 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. Understanding these dependencies is crucial as it aids in predicting the near-future evolution of sensor variables and identifying appropriate control actions. While we use an oil and gas platform scenario for illustration purposes, similar interactions exist in many important networks, such as the brain, the stock market, and smart water networks (SWN). Henceforth, we use the term *topology identification* to refer to the estimation of such dependencies.

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

Real-world systems, such as the one shown in Fig. 1, are not only dynamic but also further complex due to the nonlinear nature of their dependencies. In CPSs such as oil and gas platforms or SWNs, nonlinearity may arise from various sources, including control mechanisms of the actuator, nonlinear liquid flows (see,

Manuscript received 16 August 2022; revised 24 March 2023 and 15 May 2023; accepted 15 May 2023. Date of publication 1 June 2023; date of current version 14 June 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Subhro Das. This work was supported in part by the IKTPLUSS INDURB Under Grant 270730/O70, in part by the SFI Offshore Mechatronics Under Grant 237896/O30. *(Corresponding author: Rohan T. Money.)*

Rohan T. Money is with the WISENET Lab, Department of ICT, University of Agder, 4879 Grimstad, Norway (e-mail: rohantm@uia.no).

Joshin P. Krishnan is with the SIGIPRO Department, Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: joshin@simula.no).

Baltasar Beferull-Lozano is with the WISENET Center, Department of ICT, University of Agder, 4879 Grimstad, Norway, and also with the SIGIPRO Department, Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: baltasar.beferull@uia.no).

Digital Object Identifier 10.1109/TSP.2023.3282068

e.g., [8]), and saturation of tanks. 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), also exhibit a high level of nonlinearity, with multiple nonlinear effects contributing to the complexity of these systems [9]. Topology estimation based on simple linear models [6], [7] is inadequate for such applications, since many of the inherent nonlinear interactions within the system are discarded.

An effective way to deal with the nonlinearity is to invoke kernel machines, which can approximate any nonlinear continuous function, if 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. Deep neural networks (DNNs) are also 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 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. Moreover, such batch-based approaches suffer from a high computational complexity since the algorithm must process the entire data batch together.

The preceding discussion highlights the need for algorithms that can learn nonlinear and dynamic topologies. Kernels are an ideal choice because they offer the advantage of building interpretable models that can be learned online [14], [15], [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]. In [16], the dimensionality growth is mitigated by discarding the past data samples using a forgetting window. However, this 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. However, existing algorithms have limitations for online topology identification of multivariate time series. For instance, the sparse functional stochastic gradient descent (FSGD) method in [18] requires multiple kernel orthogonal matching pursuit (KOMP) sub-iterations, resulting in high computational complexity. Additionally, 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. An alternative approach, presented in [20], involves learning a sparse kernel dictionary based on coherence criteria. However, its convergence guarantees assume static optimal parameters (representing the topology), which is impractical for time-varying systems.

The RF approximation approach not only addresses kernel dimensionality growth but also provides greater mathematical flexibility for modelling and learning the nonlinear interaction among multivariate time series while enabling theoretical analysis. RF approximation was originally proposed in [21], and the idea has recently gained popularity in large-scale machine learning problems [22], [23], [24]. In addition to providing a computational boost in large-scale data sets, RF allows working in fixed lower dimensional spaces, making it convenient for online convex optimization routines. RF approximation in kernels can also be used to understand neural networks [25], [26], and researchers have shown equivalence in function approximation between neural networks and RF approximations [25]. Multiple Random Fourier features can be also used 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. The sparsity assumption is motivated by real-world systems that are often sparse due to the dominant local interactions, and it helps to avoid overfitting during learning. The algorithm estimates nonlinear sparse topologies in an online manner at each time instant, using a proximal optimization technique called composite objective mirror descent (COMID) and 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 theoretical guarantees based on dynamic regret analysis and 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) We propose an online nonlinear topology estimation algorithm with fixed computational complexity per iteration called *Random feature-based nonlinear topology identification via recursive sparse online learning* (RFNL-TIRSO). This algorithm differs significantly from our previous work in [31], by using a running average loss inspired by the recursive least square (RLS) formulation instead of an instantaneous loss function, which is susceptible to noise and converges slowly. Compared with [31], RFNL-TIRSO achieves significantly faster convergence speed and improved robustness to the input noise.

ii) We provide theoretical guarantees for the convergence of RFNL-TIRSO, which was lacking in [31]. The article 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 reasonable assumptions applicable to real-world applications. Our dynamic regret analysis encompasses three key elements: an online kernel-based nonlinear algorithm, a non-differentiable objective function, and a model with multiple decoupled functions for interpretable topology identification. Existing related works [33], [34], [35],

[36], [37], [38], [39] do not offer complete coverage of 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 from sensors in an oil and gas plant. Furthermore, the algorithm demonstrates its effectiveness in detecting epileptic seizure events using EEG signals.

The remainder of the article 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, followed by the numerical results in Section V. Finally, Section VI concludes the article.

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 $1 \le n \le N$, $f_{n,n'}^{(p)}$ 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 $\{f_{n,n'}^{(p)}(.)\}_{p=1}^{P}$, $\forall n, n'$, 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):

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

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, $\| f_{n,n'}^{(p)} \|_{\mathcal{H}_{n'}^{(p)}}^2 = \sum_{t=0}^{\infty} \sum_{t'=0}^{\infty} \beta_{n,n',t}^{(p)} \, \beta_{n,n',t'}^{(p)} \, \kappa_{n'}^{(p)} (y_n[t], y_n[t'])$. For further reading on RKHS, we recommend referring to [41].

The required functions $\{f_{n,n'}^{(p)} \in \mathcal{H}_{n'}^{(p)}\}_{n',p}$ 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:

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

For a non-decreasing function Ω, the solution of (3), denoted as $\{\hat{f}_{n,n'}^{(p)}\}_{n',p}$ 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 of kernel formulations, which is commonly referred to as *the curse of dimensionality.* In alignment with [14], [24], we use RF approximation to solve the curse of dimensionality.

C. RF Approximation

As observed in Section II-B, 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. By working in this fixed low-dimensional space, we can leverage standard convex optimization tools for solving the topology identification problem.

The RF approximation requires that the kernel defining the RKHS should be shift invariant, i.e., $\kappa_{n'}^{(p)}(y_{n'}[\tau-p], y_{n'}[t-p]) = \kappa_{n'}^{(p)}(y_{n'}[\tau-p] - y_{n'}[t-p])$. Many popular kernels 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

$$\kappa_{n'}^{(p)} \; (y_{n'}[\tau-p], y_{n'}[t-p]) = \int_{\mathbb{R}} \pi_{\kappa_{n'}^{(p)}}(v) \, e^{jv(y_{n'}[\tau-p] - y_{n'}[t-p])} dv$$

$$= \mathbb{E}_v \left[e^{jv(y_{n'}[\tau-p]-y_{n'}[t-p])} \right], \tag{5}$$

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

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

irrespective of the distribution $\pi_{\kappa^{(p)}_{n'}}(v)$. Notice that (6) is an unbiased estimator of the kernel evaluation in (5) [44]. In general, finding the probability distribution, which is the inverse Fourier transform of a kernel, is a challenging task. 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

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

where

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

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

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

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

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

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

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

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):

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

which can be expanded in terms of RF components as

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

For convenience, the variables $\{\alpha^{(p)}_{n,n',d}\}$ and $\{z^{(p)}_{\boldsymbol{v},n',d}(\tau)\}$ 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:

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

The function Ω in (14) is chosen to be $\Omega(.) = |.|$, where $|.|$ represents the absolute value function, in order to promote the group sparsity of $\boldsymbol{\alpha}^{(p)}_{n,n'}$ [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 satisfying the regularization criteria to apply the Representer Theorem. Using (13) and (14), a parametric form of (3) can be constructed as follows:

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

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. This batch formulation is not suitable for streaming data, where the data is available in a sequential manner and real-time tracking of time-varying topologies is required. Furthermore, the computational complexity of batch optimization can be prohibitively high, especially with large batch sizes. Motivated by the above factors, we propose an online topology estimation strategy in the following section, which addresses these limitations and has lower computational complexity.

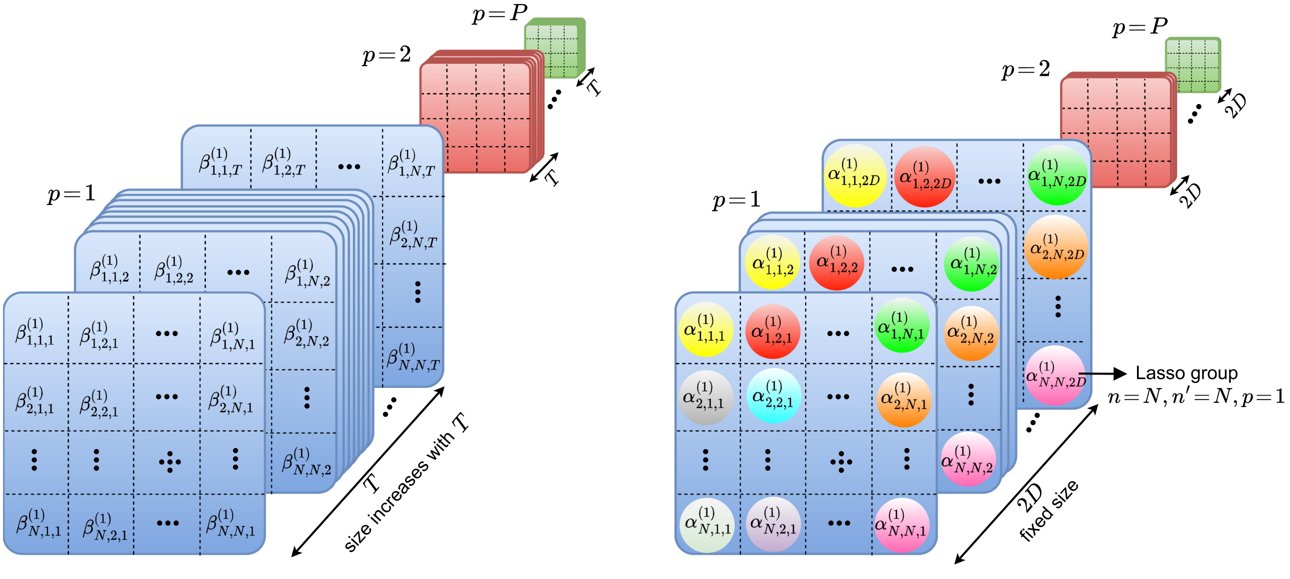


Fig. 2. RKHS parameters (left) and fixed-size RF parameters (right). The Lasso groups of RF parameters are indicated in different colours.

III. Online Learning

To formulate 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 overcome 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. To solve (23) online, we can use methods such as online subgradient descent (OSGD) or mirror descent (MD), which linearize the entire objective function using a subgradient. However, linearizing the group-Lasso regularizer, compromises its ability to induce sparsity, resulting in non-sparse estimates. To overcome this, we employ an alternate optimization technique – a modified version of the MD algorithm known as composite objective mirror descent (COMID) [45]. In COMID, the differentiable part of the objective function is linearized, while keeping the regularizer intact, preserving the sparsity-inducing property.

The online COMID updates can be written as

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

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

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:

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

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

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

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, which is shown in Algorithm 1, as *Random feature based nonlinear topology identification via recursive sparse online learning* (RFNL-TIRSO).

IV. Theoretical Results

In this section, we present the performance analysis and convergence guarantee of RFNL-TIRSO 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], [34], [35], [36] to solve problems in a dynamic environment; however, all of them belong to the class of linear algorithms. Additionally, [33], [34], [35] assume differentiable objective functions, which are not applicable to RFNL-TIRSO. Dynamic regret bounds for nonlinear algorithms have been proposed in [37], [38], [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 our proposed method for several reasons. First, the objective function used in [38] must be differentiable, whereas, in our proposed method, the regularizer is non-differentiable. Second, unlike [38], our regret analysis involves multiple decoupled functions that represent interpretable topological connections. Although [39] provides a logarithmic regret bound using second-order information, the objective function in their analysis is differentiable.

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 $\lambda > 0$, $a_t > 0$, $\theta > 0$, $D$, $\sigma_n$ 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 $\boldsymbol{z}_{\boldsymbol{v}}(t)$
    $\boldsymbol{\Phi}[t] = \gamma\boldsymbol{\Phi}[t-1] + \mu\boldsymbol{z}_{\boldsymbol{v}}(t)\boldsymbol{z}_{\boldsymbol{v}}(t)^\top$
    for $n = 1, \ldots, N$ do
        $\boldsymbol{r}_n[t] = \gamma\boldsymbol{r}_n[t-1] + \mu y_n[t]\boldsymbol{z}_{\boldsymbol{v}}(t)$
        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
```

Our theoretical analysis is based on the following assumptions:

- ***A1*** : Bounded samples: For all the time series samples, there exists $B_y > 0$ such that $\{|y_n[t]|^2\}_{n,t} \leq B_y \leq \infty$.
- ***A2*** : Shift-invariant kernels: kernels used are shift-invariant, i.e., $k(x_i, x_j) = k(x_i - x_j)$.
- ***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.
- ***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.

A1 is reasonable in practice as the signals from real-world applications are bounded. ***A2*** is true for typical kernels such as Gaussian and Laplacian. Since $\boldsymbol{\Phi}(t)$ is a sum of rank one matrices formed using feature vectors, ***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 ***A3*** is important for the strong convexity assumption of the loss function, which is used in the sequel. ***A4*** can be obtained by combining ***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

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

where the cost function $h^n_t(\hat{\boldsymbol{\beta}}_n, \boldsymbol{\kappa}_t)$ in (30) is composed of two terms: $h^n_t(\hat{\boldsymbol{\beta}}_n, \boldsymbol{\kappa}_t) = \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} \frac{1}{2}[y_n[t] - \hat{\boldsymbol{\beta}}_n^\top \boldsymbol{\kappa}_t]^2 + \omega^n(\hat{\boldsymbol{\beta}}_n)$, where the first term is RLS loss function, and the second term $\omega^n(.)$ 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

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

where $h^n_t(\boldsymbol{\alpha}_n, \boldsymbol{z}_{\boldsymbol{v}}(t)) = \tilde{\ell}^n_t(\boldsymbol{\alpha}_n) + \omega^n(\boldsymbol{\alpha}_n)$, and $\tilde{\ell}^n_t(.)$ is the RLS loss defined in (17) and $\omega^n(\boldsymbol{\alpha}_n) = \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{\alpha}^{(p)}_{n,n'}\|_2$. Note 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

$$\boldsymbol{R}_n[T] = \sum_{t=P}^{T-1} \left[h^n_t(\boldsymbol{\alpha}_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t)) - h^n_t(\boldsymbol{\beta}^*_n[t], \boldsymbol{\kappa}_t)\right]. \tag{32}$$

Our aim is to find a theoretical bound for $\boldsymbol{R}_n[T]$. Since our online algorithm works in the RF space, we conduct the regret analysis in relation to the optimal cost function in the RF space, i.e., $h^n_t(\boldsymbol{\alpha}^*_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t))$. It is worth noting that this choice is not arbitrary as there exists a one-to-one mapping between the two spaces, ensuring no loss of generality. Adding and subtracting $h^n_t(\boldsymbol{\alpha}^*_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t))$ in (32) yields

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

where $\boldsymbol{R}^{\text{rf}}_n[T] = \sum_{t=P}^{T-1}[h^n_t(\boldsymbol{\alpha}_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t)) - h^n_t(\boldsymbol{\alpha}^*_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t))]$ is the regret with respect to optimal cost in RF space and $\boldsymbol{\xi}_n[T] = \sum_{t=P}^{T-1}[h^n_t(\boldsymbol{\alpha}^*_n[t], \boldsymbol{z}_{\boldsymbol{v}}(t)) - h^n_t(\boldsymbol{\beta}^*_n[t], \boldsymbol{\kappa}_t)]$ 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

$$\boldsymbol{R}^{\text{rf}}_n(T) \leq \left(\left(1 + \frac{L}{\rho_l}\right)\sqrt{2PNDB_y} + \lambda\sqrt{PN}\right)$$

$$\times \left(\|\boldsymbol{\alpha}^*_n[P]\|_2 + \boldsymbol{W}_n(T)\right),$$

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

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

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 Theorems 1 and 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

$$\boldsymbol{R}_n(T) \leq \left(\left(1 + \frac{L}{\rho_l}\right)\sqrt{2PNDB_y} + \lambda\sqrt{PN}\right)$$

$$\times \left(\|\boldsymbol{\alpha}^*_n[P]\|_2 + \boldsymbol{W}_n(T)\right) + \epsilon L_h TC.$$

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

It is important to note that by setting $\epsilon = \mathcal{O}(\frac{1}{\sqrt{T}})$, we can achieve a dynamic regret of $\mathcal{O}(\boldsymbol{W}_n(T) + \sqrt{T})$. In such cases, if $\boldsymbol{W}_n(T)$ is sublinear, the dynamic regret becomes sublinear as well. Ideally, an online algorithm should aim for a sublinear dynamic regret, indicating that $\boldsymbol{R}_n(T)/T \rightarrow 0$ as $T \rightarrow \infty$, or in the worst case, a linear regret, which implies $\boldsymbol{R}_n(T)/T \rightarrow \mathit{constant}$, where $\mathit{constant}$ is known as the steady-state error. In our case, when $\boldsymbol{W}_n(T)$ is sublinear, the steady-state error is $\epsilon L_f C$. By choosing a small value for ϵ, we can ensure a small steady-state error. Appendix $\boldsymbol{B}$ demonstrates that we can make ϵ sufficiently small by increasing the number of random features D while considering the trade-off with complexity [21].

V. Experimental Results

In this section, we evaluate the performance of RFNL-TIRSO through extensive numerical experiments. We compare it with three state-of-the-art competitors: TIRSO [6], RFNL-TISO [31], and PDIS [20], [49]. TIRSO is a linear online topology algorithm, chosen to highlight the advantages of RFNL-TIRSO, which is a nonlinear algorithm. RFNL-TISO is another online nonlinear topology estimation algorithm that uses an instantaneous least mean square loss function. However, RFNL-TIRSO is expected to outperform RFNL-TISO due to its utilization of an RLS-based loss function, as discussed in Section III. The third algorithm, PDIS [20], [49], is a recent online nonlinear topology identification algorithm that uses dictionaries of kernel functions with partial-derivative-imposed sparsity. To the best of our knowledge, these three algorithms serve as the best benchmarks for comparing the performance of RFNL-TIRSO. While other batch-based algorithms are available [10], [12], [13], they are not directly comparable to our algorithm as they operate offline.

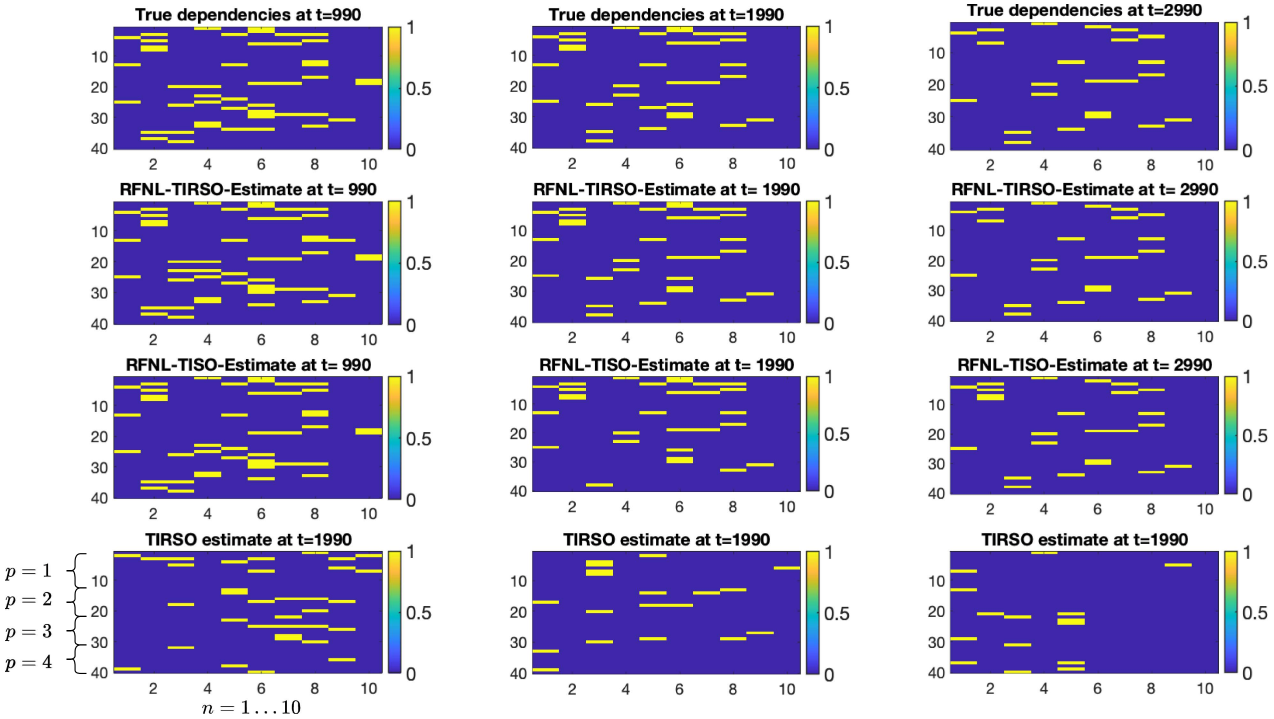


Fig. 3. The true and estimated edges using various algorithms for $g(x) = g_1(x)$. In each subfigure, the x-axis corresponds to nodes $n = 1, \dots, 10$, and the y-axis corresponds to nodes $n = 1, \dots, 10$ for time lags $p = 1, \dots, 4$. The edge values are indicated by the colour code.

The per node computational complexity of RFNL-TIRSO, RFNL-TISO, and TIRSO, are in the order of $\mathcal{O}(N^2P^2D^2)$, $\mathcal{O}(NPD)$, and $\mathcal{O}(N^2P^2)$, respectively. Although RFNL-TIRSO has a higher computational complexity than the other algorithms, it offers robustness, and theoretical performance guarantees, which are not provided by the competing algorithms. We demonstrate the robustness and performance of RFNL-TIRSO through several numerical experiments in this section.

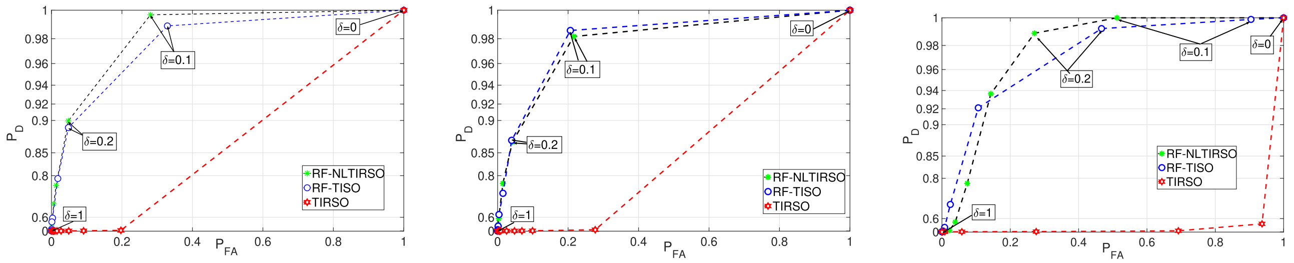
The experiments in this section involve both synthetic and real data sets. The synthetic dataset consists of graph-connected time series data generated with different topology transition patterns to highlight the ability of algorithm to track time-varying topologies. The real data set used include (i) 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. In the experiments, the initial P data samples are generated randomly, and the remaining data are generated using model (1). The hyperparameters for 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 $\{a^{(p)}_{n,n'}\}$ at different time steps, which are arranged in matrices of size $N \times N$, for $p = 1, 2, \dots, P$, and stacked vertically, resulting in matrices of size $NP \times N$. The estimated dependencies $\{\hat{a}^{(p)}_{n,n'}\}$ using different algorithms are shown in the bottom rows. After computing the normalized ℓ_2 norms $\widehat{b}^{(p)}_{n,n'}[t] = \|\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\|_2/(\max_{n'}\|\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\|_2)$, 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-TIRSO are very close to the ground truth, and they outperform the other algorithms.

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 $p - th$

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



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

Fig. 4. Receiver-Operating Curve for different realizations of the nonlinear function $g(x)$ (averaged over 50 experiment runs).

TABLE I
AUC for Different Algorithms (Computed Over 50 Experiment Runs). SD Indicates the Standard Deviation

Algorithm	g_1		g_2		g_3	
	AUC Mean	SD	AUC Mean	SD	AUC Mean	SD
$RFNL-TIRSO$	**0.9914**	**0.0022**	**0.9949**	**0.0024**	**0.9543**	**0.0143**
$RFNL-TISO$	0.9741	0.0045	0.9817	0.0053	0.9317	0.0214
$TIRSO$	0.4967	0.0421	0.5000	0.0532	0.5123	0.0451

TABLE II
AUC for Different Values of D (Computed Over 50 Experiment Runs). SD Indicates the Standard Deviation

No. of RF Features	$t = 990$		$t = 1990$		$t = 2990$	
	AUC Mean	SD	AUC Mean	SD	AUC Mean	SD
$D = 20$	0.9500	0.0050	0.9762	0.0033	0.9732	0.0031
$D = 30$	0.9568	0.0035	0.9827	0.0031	0.9835	0.0029
$D = 50$	0.9721	0.0031	0.9887	0.0025	**0.9901**	**0.0020**

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

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

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

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 $t = 2990$ 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 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.

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

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

In comparison with the model used in Section V-A1, the Lorenz graph model (35) introduces only first-order dependencies ($P = 1$) among the nodes. Additionally, it is important to note that (35) incorporates non-additive nonlinear interactions among the nodes, which differs from the VAR assumption in (1). In this

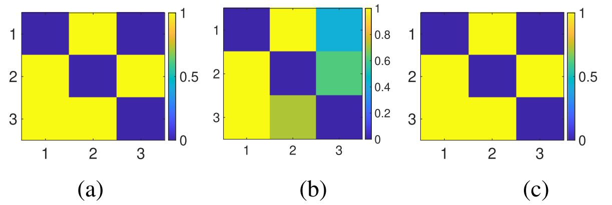


Fig. 5. Lorenz graph detection using RFNL-TIRSO: (a) True Binary dependency, (b) Estimated dependency, (c) Binary estimated dependency by stetting threshold as 0.5.

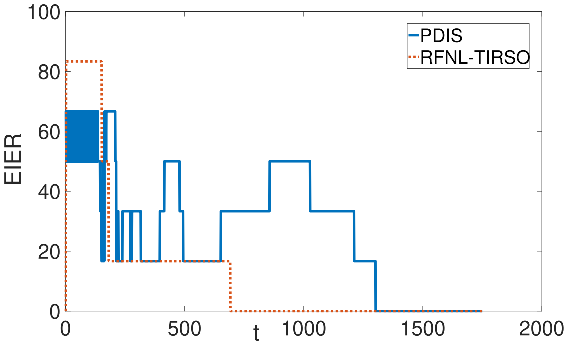


Fig. 6. EIER performance for the Lorenz graph experiment.

section, we compare the performance of the RFNL-TIRSO algorithm and the PDIS algorithm [49]. Since the TIRSO algorithm assumes $P > 1$ in its implementation, it is not included in this comparison. To ensure a fair comparison, we replicate the exact experimental setup as described in [49]. The performance evaluation is based on the *edge identification error rate* (EIER), defined as $EIER = \frac{\|\mathbf{A}-\hat{\mathbf{A}}\|_0}{N(N-1)} \times 100$, where $\mathbf{A}$ represents the true dependency matrix and $\hat{\mathbf{A}}$ represents the estimated dependency matrix. For RFNL-TIRSO, $\hat{\mathbf{A}}$ is computed using $\widehat{b}^{(1)}_{n,n'}$. The hyperparameters are heuristically tuned to minimize the EIER, resulting in the setting $(\sigma_n, \lambda, a_t) = (1, .3, 1/(t\Lambda_{\max}(\boldsymbol{\phi}[t])))$.

The estimated and true binary adjacency matrices (excluding self-dependencies) are shown in Fig. 5, and the EIER up to $t = 1750$ are 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 with the LMS loss used in PDIS.

3) Numerical Evaluation of Dynamic Regret: In Section IV-A, we derived a theoretical bound for the dynamic regret $\boldsymbol{R}_n[T] = \boldsymbol{R}^{\text{rf}}_n[T] + \boldsymbol{\xi}_n[T]$. 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 $\boldsymbol{R}^{\text{rf}}_n[T] = \sum_{\tau=P}^{T-1}[h^n_\tau(\boldsymbol{\alpha}_n[\tau], \boldsymbol{z}_{\boldsymbol{v}}(\tau)) - h^n_\tau(\boldsymbol{\alpha}^*_n[\tau], \boldsymbol{z}_{\boldsymbol{v}}(\tau))]$, for $T = 1, \ldots, 1000$. This allows us to experimentally validate 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. It is worth mentioning that the estimation of $\boldsymbol{\alpha}^*_n[\tau]$ involves very high computational complexity compared with that of $\boldsymbol{\alpha}_n[\tau]$. In Fig. 7, we plot $\boldsymbol{R}^{\text{rf}}_n[T]$ and $\boldsymbol{R}^{\text{rf}}_n[T]/T$. For this experiment, we use the same data generation mechanism with nonlinear dependencies g_1 and g_2, as explained in Section V-A1, having topology change points at $T = 250$ and $T = 500$. Fig. 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. It is important to note that numerically evaluating the second component of the dynamic regret $\boldsymbol{\xi}_n[T]$ is a complex task since it involves finding optimal parameters in a high dimensional RKHS. However, as shown in Theorem 2 we emphasize that $\boldsymbol{\xi}_n[T]/T$ is theoretically bounded by $\epsilon L_f C$, 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 $T \to \infty$.

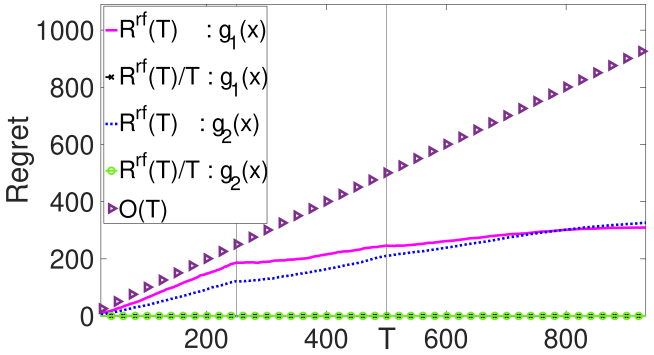


Fig. 7. Regret w.r.t. optimal cost function in RF space. Vertical lines indicate the topology change points.

B. Experiments Using Real Data Sets

1) Oil and Gas Platform Data: In this section, we describe experiments conducted using real data collected from Lundin's Offshore Oil and Gas 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. 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 within a

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

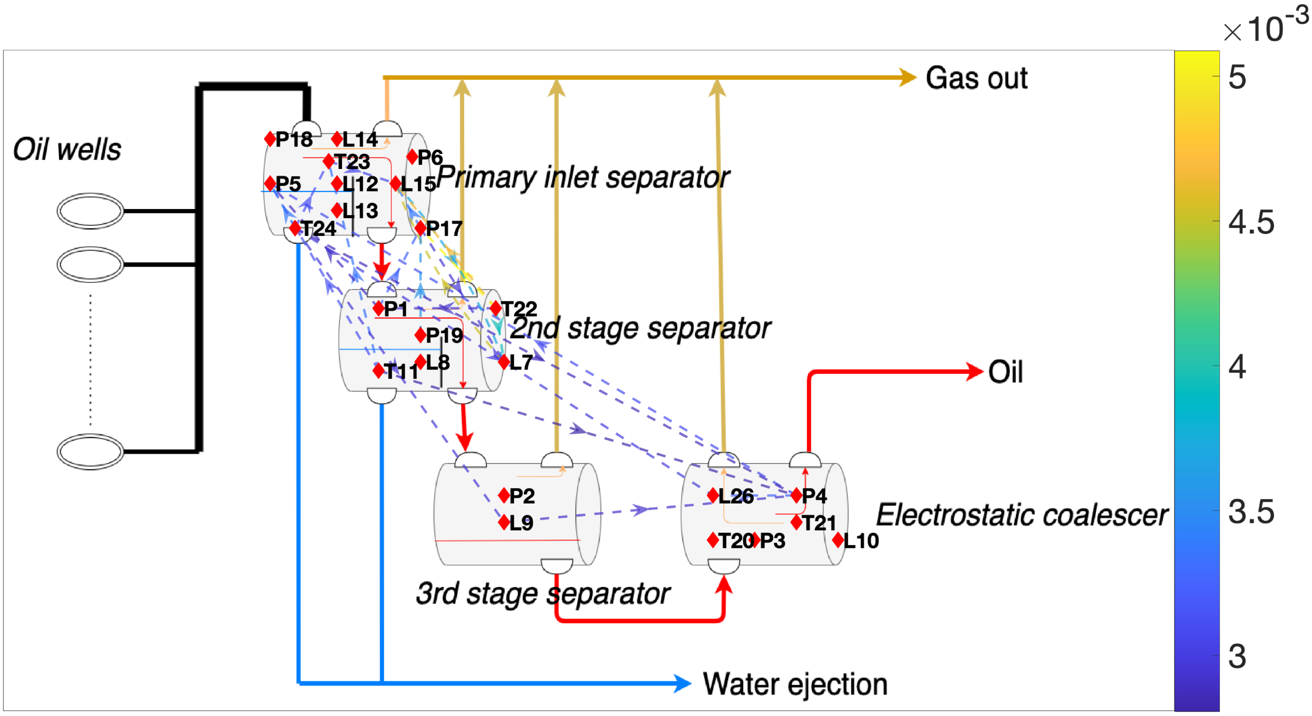


Fig. 8. Topology estimated using RFNL-TIRSO for Oil and Gas platform. Temperature, pressure, and level sensors are denoted by the labels ‘T’,‘P’, and ‘L’ in the node index, respectively.

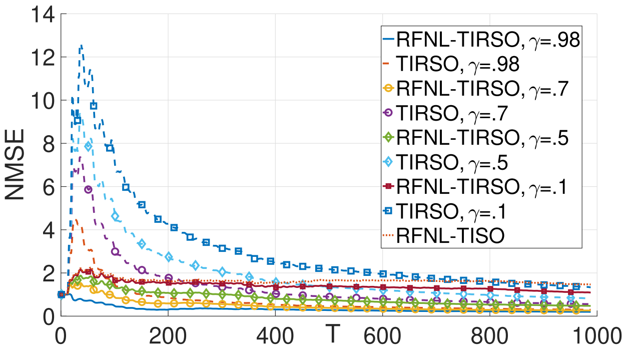


Fig. 9. NMSE comparison: data from the Oil and Gas platform.

container are directly linked to fluid dynamics of the oil and gas in the closed chamber as governed by the underlying differential equations [52]. However, various control mechanisms governing the whole oil and gas platform and the physical connections 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, which 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 spatio-temporal variation of the signals. However, since ground truth dependencies are not available in this experiment, directly evaluating the estimated graph using the underlying differential physics-based equations is a complex and time-consuming task that is beyond the scope

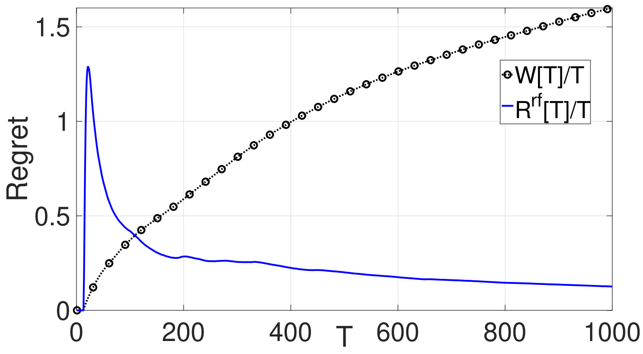


Fig. 10. Time-normalized regret and path length: data from Edvard-Grieg Oil and Gas platform.

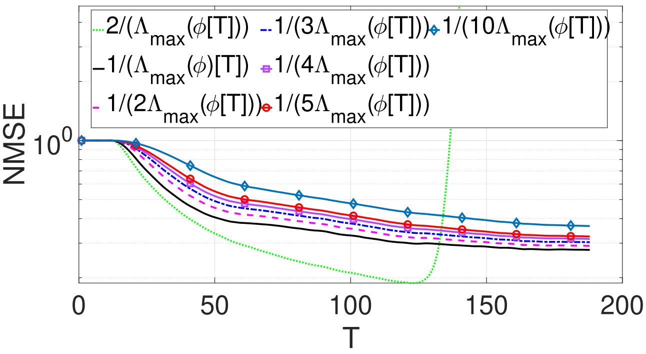


Fig. 11. NMSE with different learning rate a_t: data from Edvard-Grieg Oil and Gas platform.

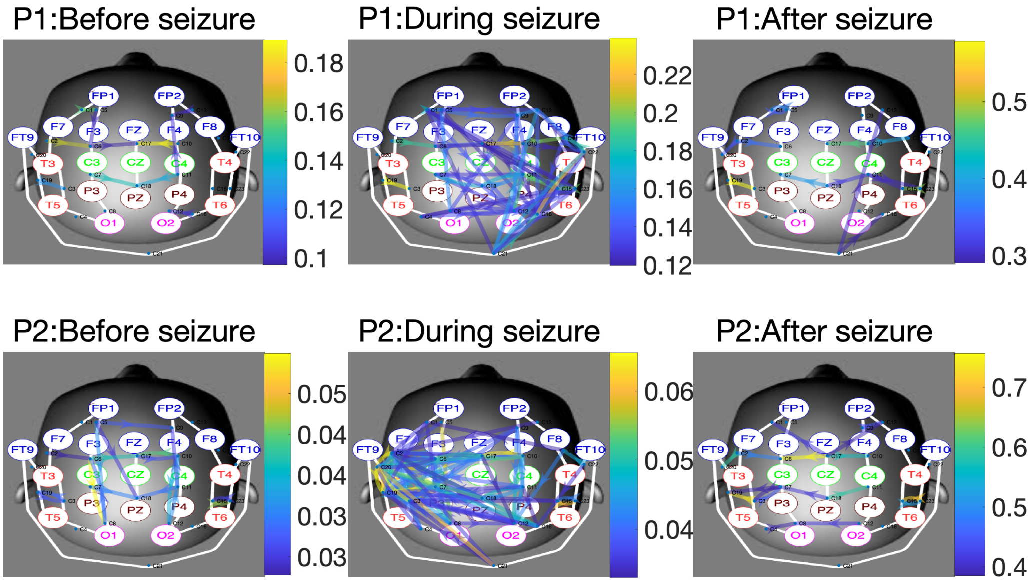


Fig. 12. Estimated brain topology for the subjects P1 and P2 during various stages of seizure.

of this study. Instead, we assess the ability of the algorithms to learn the dependencies based on the accuracy of time series forecasting using the learned VAR model. A high prediction accuracy implies that the estimated dependencies are close to the underlying unknown true dependencies. In our study, we measure the prediction accuracy using normalized mean squared error (NMSE):

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

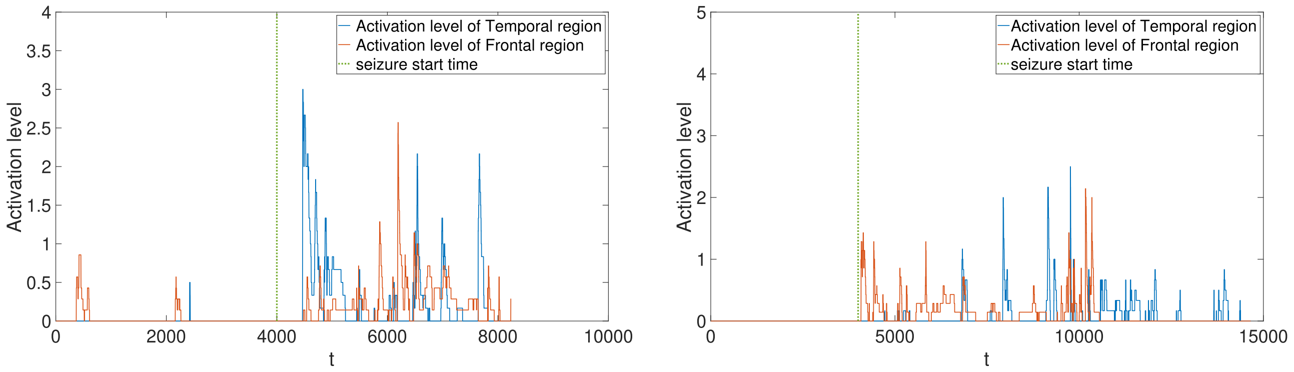
where $\hat{y}_n[t+t_{step}]$ is the estimate of the time series generated by the n^{th} node at time instant $t+t_{step}$ based on the VAR model learned at time t. Fig. 9 shows the NMSE of the estimated signals corresponding to a particular sensor $n=8$ using various algorithms. We exclude the PDIS algorithm in this experiment since it is not specifically designed for signal prediction. The NMSE is calculated according to (36) with a prediction horizon of $t_{step} = 12$, representing one-minute ahead prediction. For the RFNL-TIRSO and TIRSO algorithms, we conduct the experiments by varying the forgetting factor $\gamma \in \{0.1, 0.5, 0.7, 0.98\}$. The best NMSE performance is achieved by the RFNL-TIRSO algorithm when $\gamma = 0.98$, surpassing all the competing algorithms. As γ decreases, the performance of RFNL-TIRSO approaches that of RFNL-TISO, as expected from (17). Furthermore, we plot the dynamic regret and cumulative variation of the optimal parameter estimates in Fig. 10, demonstrating that our algorithm is capable of tracking the topology even when the optimal topology undergoes variations.

In 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. Intuitively as the learning rate increases, RFNL-TIRSO converges faster; and when the learning rate surpasses $1/L$, convergence is not guaranteed, as evidenced in Fig. 11. Note that if the data has a high variance, the value of $\Lambda_{\max}(\boldsymbol{\phi}[T])$ will also be high, necessitating the use of a lower learning rate to ensure 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 after discontinuing anti-seizure medication to characterize their seizures and assess their candidacy for surgical intervention. The EEG recordings followed the electrode positions and nomenclature of the well-known *International 10-20 system* standard. The signals were sampled at a rate of 64 samples per second, and a total of 23 channels were recorded: 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, representing the potential difference between the corresponding electrodes.

The estimated brain topologies using RFNL-TIRSO ($P = 2, D = 20$) at different time instants (before seizure, during seizure, after seizure) are visualized in Fig. 12, based on the ℓ_2 norms $\|\boldsymbol{\alpha}_{n,n'}[t]\|_2$. It can be observed that the estimated topologies before and after the seizure are very similar, with connections concentrated across specific brain regions. However, during the seizure, the topologies become more disrupted, which aligns with the findings in [53]. This disruption can be attributed to increased 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 originate from the temporal region, whereas in FL epilepsy, such connections originate from the frontal region. To illustrate this, we present an experiment using the brain data of P1 and P2, who belong to the TL and FL epilepsy categories [56], respectively. To measure the activity level of different brain regions, we group all the channels connected to the 'temporal' region into group-T and the 'frontal' region into group-F. Note that all the connections between the 'frontal' and the 'temporal' regions are excluded



(a) Subject: P1, Category: Temporal Lobe Epilepsy (b) Subject: P2, Category: Frontal Lobe Epilepsy

Fig. 13. Activation levels in ‘T’ and ‘F’ regions of the brain.

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 in the group, where the degree of a node refers to the total number of edges connected to the node. The activation levels of group-T and group-F for P1 and P2 are shown in Fig. 13(a) and (b), respectively. From the figures, it can be observed that for P1 and P2, the activation levels of group-T and group-F, respectively, exhibit an initial spike, and then the activation spreads across the other brain regions. These observations align with the characteristics of TL and FL epilepsies.

VI. Conclusion

This article presents a novel online nonlinear topology identification algorithm called RFNL-TIRSO. The algorithm is designed to process multivariate time series data in a sequential manner and estimate time-varying nonlinear dependencies. The theoretical analysis demonstrates that RFNL-TIRSO follows a sublinear dynamic regret, guaranteeing its ability to track changes in the topology of the system in dynamic environments. To evaluate the performance of RFNL-TIRSO, both real and synthetic data sets are used, and the algorithm outperforms the state-of-the-art online topology estimation methods.

Appendix A
Proof of Theorem 1

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

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

Note that $\nabla h_t^n(\boldsymbol{\alpha}_n[t], \boldsymbol{z_v}(t))$ is the gradient of $h_t^n(\boldsymbol{\alpha}_n[t], \boldsymbol{z_v}(t))$ with respect to $\boldsymbol{\alpha}_n[t]$. Apply Cauchy-Schwarz inequality on right hand side of (37) to get

$$\begin{aligned}\boldsymbol{R}_n^{\mathrm{rf}}(T) &= \sum_{t=P}^{T-1} \left[h_t^n(\boldsymbol{\alpha}_n[t], \boldsymbol{z_v}(t)) - h_t^n(\boldsymbol{\alpha}_n^*[t], \boldsymbol{z_v}(t)) \right] \\ &\le \sum_{t=P}^{T-1} \|\nabla h_t^n(\boldsymbol{\alpha}_n[t], \boldsymbol{z_v}(t))\|_2 \, \|\boldsymbol{\alpha}_n[t] - \boldsymbol{\alpha}_n^*[t]\|_2.\end{aligned} \tag{38}$$

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 $\|\nabla h_t^n(\boldsymbol{\alpha}_n[t], \boldsymbol{z_v}(t))\|_2$ in (38).

Lemma 1: Under the assumptions A1, A3 and A4,

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

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:

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

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

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

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

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

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

$$\|\boldsymbol{r}_n[t]\|_2 = \left\| \mu \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau] \mathbf{z}_{\boldsymbol{v}}(\tau) \right\|_2$$

$$\le \mu \left\| \sum_{\tau=P}^{t} \gamma^{t-\tau} y_n[\tau] \mathbf{1}_{2PND} \right\|_2 \tag{42}$$

$$\le \mu \sqrt{2PNDB_y} \gamma^t \sum_{\tau=P}^{t} \left(\frac{1}{\gamma}\right)^{\tau} \tag{43}$$

$$= \sqrt{2PNDB_y} \left(1 - \gamma^{t-P+1}\right) \tag{44}$$

$$\le \sqrt{2PNDB_y}. \tag{45}$$

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 $\mu = 1 - \gamma$, and (45) follows from $\gamma \le 1$. 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

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

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

$$\|\boldsymbol{\alpha}_n[t+1]\|_2 \le \delta \|\boldsymbol{\alpha}_n[t]\|_2 + a_t \sqrt{2PNDB_y} \tag{46}$$

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):

$$\|\boldsymbol{\alpha}_n[t+1]\|_2 \le \delta^{t-P+1} \|\boldsymbol{\alpha}_n[P]\|_2 + a\sqrt{2PNDB_y} \sum_{i=0}^{t-P} \delta^i$$

$$= \frac{a\sqrt{2PNDB_y}(1 - \delta^{t-P+1})}{1 - \delta} \tag{47}$$

$$\le \frac{a\sqrt{2PNDB_y}}{1 - (1 - a\rho_l)}) = \frac{1}{\rho_l} \sqrt{2PNDB_y} \tag{48}$$

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:

$$\|\nabla \tilde{\ell}_t^n(\boldsymbol{\alpha}_n[t])\|_2 = \|\boldsymbol{\phi}[t]\boldsymbol{\alpha}_n[t] - \boldsymbol{r}_n[t]\|_2 \text{ (from (22))}$$

$$\le \|\boldsymbol{\phi}[t]\boldsymbol{\alpha}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2$$

$$\le \Lambda_{\max}(\boldsymbol{\phi}[t]) \|\boldsymbol{\alpha}_n[t]\|_2 + \|\boldsymbol{r}_n[t]\|_2 \tag{49}$$

$$= L\frac{\sqrt{2PNDB_y}}{\rho_l} + \sqrt{2PNDB_y} \tag{50}$$

$$\le \left(1 + \frac{L}{\rho_l}\right) \sqrt{2PNDB_y} \tag{51}$$

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

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

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

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

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

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

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

$$|\omega^n(\boldsymbol{a}_n) - \omega^n(\boldsymbol{b}_n)| \tag{54}$$

$$= \lambda \left| \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{b}_{n,n'}^{(p)}\|_2 \right| \tag{55}$$

$$= \lambda \left| \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \|\boldsymbol{b}_{n,n'}^{(p)}\|_2 \right| \tag{56}$$

$$\le \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \left| \|\boldsymbol{a}_{n,n'}^{(p)}\|_2 - \|\boldsymbol{b}_{n,n'}^{(p)}\|_2 \right| \tag{57}$$

$$\le \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{a}_{n,n'}^{(p)} - \boldsymbol{b}_{n,n'}^{(p)}\|_2 \tag{58}$$

$$\le \lambda\sqrt{PN} \|\boldsymbol{a}_n - \boldsymbol{b}_n\|_2. \tag{59}$$

In the above derivation, inequality (57) follows from the triangle inequality, inequality (58) from the reverse triangle inequality and (59) from the basic inequality $\|q\|_1 \le \sqrt{M}\|q\|_2$, $q \in R^M$. 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

$$\boldsymbol{\xi}_n[T] \le \left| \sum_{t=P}^{T-1} \left[h_t^n(\boldsymbol{\alpha}_n^*[t], \boldsymbol{z}_{\boldsymbol{v}}(t)) - h_t^n(\boldsymbol{\beta}_n^*[t], \boldsymbol{\kappa}_t)\right] \right|. \tag{60}$$

Using the triangle inequality,

$$\boldsymbol{\xi}_n[T] \le \sum_{t=P}^{T-1} \left| h_t^n(\boldsymbol{\alpha}_n^*[t], \boldsymbol{z}_{\boldsymbol{v}}(t)) - h_t^n(\boldsymbol{\beta}_n^*[t], \boldsymbol{\kappa}_t) \right|$$

$$\le \sum_{t=P}^{T-1} L_h \left| \sum_{n'=1}^{N} \sum_{p=1}^{P} \sum_{t'=P}^{t+p-1} \left[\beta_{n,n',(t'-p)}^{(p)*} \boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t)^{\top} \boldsymbol{z}_{\boldsymbol{v},n'}^{(p)}(t') \right.\right.$$

$$\left.\left. - \beta_{n,n',(t'-p)}^{(p)*} k_{n'}^{(p)} \left(y_{n'}[t-p], y_{n'}[t'-p]\right) \right] \right| \tag{61}$$

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

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], we can prove that for a given shift-invariant kernel $k^{(p)}_{n'}$ (assumption A2), the approximation error due to the random Fourier approximation is bounded by

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

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

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

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

$$\boldsymbol{\xi}(T) \le \sum_{t=P}^{T-1} L_h\epsilon \sum_{n'=1}^{N}\sum_{p=1}^{P}\sum_{t'=P}^{t+p-1}\left|\beta^{(p)*}_{n,n',(t'-p)}\right| \qquad (65)$$
$$\le \sum_{t=P}^{T-1}\epsilon L_h C \qquad (66)$$
$$\le \epsilon L_h T C, \qquad (67)$$

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.

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].

References

[1] Y. Fan, J. Li, and D. Zhang, "A method for identifying critical elements of a cyber-physical system under data attack," *IEEE Access*, vol. 6, pp. 16972–16984, 2018.

[2] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1189–1203, Oct. 2016.

[3] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognit.*, vol. 121, 2022, Art. no. 108218.

[4] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[5] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proc. IEEE*, vol. 108, no. 11, pp. 2032–2048, Nov. 2020.

[6] B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Trans. Signal Process.*, 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5400–5404.

[8] M. K. Singh and V. Kekatos, "Optimal scheduling of water distribution systems," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 2, pp. 711–723, Jun. 2020.

[9] C. Stam, *Nonlinear Brain Dynamics*. Waltham, MA, USA: Nova Biomedical, 2006.

[10] Y. Shen, G. B. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5325–5339, Oct. 2019.

[11] N. Meike, B. Doina, and S. Christin, "Causal discovery with attention-based convolutional neural networks," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 312–340, 2019.

[12] L. Lopez-Ramos, K. Roy, and B. Beferull-Lozano, "Explainable nonlinear modelling of multiple time series with invertible neural networks," in *Proc. Intell. Technol. Appl.: 4th Int. Conf.*, 2021, pp. 17–30.

[13] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, "Neural Granger causality for nonlinear time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4267–4279, Aug. 2022.

[14] J. Lu, S. C. H. Hoi, J. Wang, P. Zhao, and Z. Liu, "Large scale online kernel learning," *J. Mach. Learn. Res.*, vol. 17, no. 47, pp. 1–43, 2016.

[15] L. Zhang, J. Yi, R. Jin, M. Lin, and X. He, "Online kernel learning with a near optimal sparsity bound," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 621–629.

[16] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online non-linear topology identification from graph-connected time series," in *Proc. IEEE Data Sci. Learn. Workshop*, 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*. Berlin, Germany: Springer, 2005.

[18] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4671–4675.

[19] Y. Shen and G. B. Giannakis, "Online identification of directional graph topologies capturing dynamic and nonlinear dependencies†," in *Proc. IEEE Data Sci. Workshop*, 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 Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 78–91, 2022.

[21] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.

[22] J. Lu, S. Hoi, J. Wang, P. Zhao, and Z. Liu, "Large scale online kernel learning," *J. Mach. Learn. Res.*, vol. 17, no. 47, pp. 1–43, 2016.

[23] T. Nguyen, T. Le, H. Bui, and D. Phung, "Large-scale online kernel learning with random feature reparameterization," in *Proc. 26th Int. Joint Conf. AI*, 2017, pp. 2543–2549.

[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, pp. 773–808, Jan. 2019.

[25] G. Yehudai and O. Shamir, "On the power and limitations of random features for understanding neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 32, 2019.

[26] R. Sato, M. Yamada, and H. Kashima, "Random features strengthen graph neural networks," in *Proc. SIAM Int. Conf. Data Mining*, 2021, pp. 333–341.

[27] B. Can and H. Ozkan, "A neural network approach for online nonlinear Neyman-Pearson classification," *IEEE Access*, vol. 8, pp. 210234–210250, 2020.

[28] F. Porikli and H. Ozkan, "Data driven frequency mapping for computationally scalable object detection," in *Proc. IEEE 8th Int. Conf. Adv. Video Signal Based Surveill.*, 2011, pp. 30–35.

[29] H. Ozkan, N. D. Vanli, and S. S. Kozat, "Online classification via self-organizing space partitioning," *IEEE Trans. Signal Process.*, vol. 64, no. 15, pp. 3895–3908, Aug. 2016.

[30] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, Nov. 2015.

[31] R. Money, J. Krishnan, and B. Beferull-Lozano, "Random feature approximation for online nonlinear graph topology identification," *Proc. IEEE 31st Int. Workshop Mach. Learn. Signal Process.*, 2021, pp. 1–6.

[32] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, "Dynamic regret of strongly adaptive methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5882–5891.

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

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

[35] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, “Online optimization : Competing with dynamic comparators,” in *Proc. Artif. Intell. Statist.*, 2015, pp. 398–406,.

[36] 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.

[37] A. Chakraborty, K. Rajawat, and A. Koppel, “Sparse representations of positive functions via first- and second-order pseudo-mirror descent,” *IEEE Trans. Signal Process.*, 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,” in *Proc. IEEE Amer. Control Conf.*, 2020, pp. 321–326.

[39] D. Calandriello, A. Lazaric, and M. Valko, “Second-order kernel online convex optimization with adaptive sketching,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 645–653.

[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*. Philadelphia, PA, USA: SIAM Press,1990.

[42] B. Olkopf, R. Herbrich, A. Smola, and R. Williamson, “A generalized representer theorem,” *Comput. Learn. Theory*, vol. 42, pp. 416–426, 2000.

[43] S. Bochner, *Lectures on Fourier Integrals*, vol. 42. Princeton, NJ, USA: Princeton Univ. Press, 1959.

[44] 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 Glob. Conf. Signal Inf. Process.*, 2018, pp. 773–777.

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

[46] M. Gutmann and J. Hirayama, “Bregman divergence as general framework to estimate unnormalized statistical models,” in *Proc. UAI*, Arlington, Virginia, USA, 2011, pp. 283–290.

[47] A. T. Puig, A. Wiesel, and A. O. Hero, “A multidimensional shrinkage-thresholding operator,” in *Proc. IEEE 15th Workshop Stat. Signal Process.*, 2009, vol. 18, pp. 113–116.

[48] M. A. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 928–936.

[49] M. Moscu, R. Borsoi, and C. Richard, “Online kernel-based graph topology identification with partial-derivative-imposed sparsity,” in *Proc. IEEE 28th Eur. Signal Process. Conf.*, 2021, pp. 2190–2194.

[50] A. Shoeb, “Application of machine learning to epileptic seizure onset detection and treatment,” PhD Thesis, MIT, Cambridge, MA, USA, 2009.

[51] E. N. Lorenz, “Computational chaos-a prelude to computational instability,” *Physica D: Nonlinear Phenomena*, vol. 35, no. 3, pp. 299–317, 1989.

[52] S. Song et al., “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,” in *Proc. Int. IEEE/EMBS 9th Conf. Neural Eng.*, 2019, pp. 130–133.

[54] F. Pittau, F. Fahoum, R. Zelmann, F. Dubeau, and J. Gotman, “Negative bold response to interictal epileptic discharges in focal epilepsy,” *Brain Topogr.*, vol. 26, pp. 627–640, 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 J. Neurol.*, vol. 119, no. 1, pp. 17–40, 1996.

[56] N. Saadat and P. Hossein, “Epileptic seizure onset detection algorithm using dynamic cascade feed-forward neural networks,” in *Proc. Int. Conf. Intell. Comput. Bio- Med. Instrum.*, 2011, pp. 196–199.

Rohan T. Money (Student Member, IEEE) received the B.Tech. Degree in electrical and electronics engineering from Rajiv Gandhi Institute of Technology, M.G. University, Kottayam, India, in 2015, and the M.Tech. degree in systems and control from the Indian Institute of Technology Hyderabad, Hyderabad, India, in 2018. Since 2019, he has been working toward the Ph.D. degree with the WISENET Research Center, University of Agder, Kristiansand, Norway. His research interests include optimization, time-series analysis, graph signal processing, machine learning, and control theory. He was the recipient of the best student paper runners-up award at the IEEE DSLW 2021 conference.

Joshin P. Krishnan (Member, IEEE) received the B.Tech. Degree in electronics and communication engineering from the College of Engineering, University of Kerala, Thiruvananthapuram, India, in 2010, the M.E. degree in telecommunications from the Indian Institute of Science, Bengaluru, India, in 2014, and the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico, University of Lisbon, Portugal, Lisbon, in 2019. From 2015 to 2019, he was with the Instituto de Telecomunicações, Lisbon, as a Marie Curie Early-Stage Researcher of the Machine Sensing Training Network (MacSeNet). From 2019 to 2021, he was a Postdoctoral Researcher with WISENET Research Center, UiA, Norway. He is currently a Postdoctoral Researcher with SIMULA Research Center, Norway. His research interests include image inverse problems, optimization, time-series analysis, graph signal processing, and machine learning.

Baltasar Beferull-Lozano (Senior Member, IEEE) received the M.Sc. degree in physics (First-in-Class Honours) from Universidad de Valencia, Valencia, Spain, in 1995 and the M.Sc. and P.hD. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1999 and 2002, respectively. In October 2002, he joined the AudioVisual Communications Laboratory, Department of Communication Systems, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland, as a Research Associate, where he spent around three years. In January 2006, he joined the School of Engineering at University of Valencia as Associate Professor. Since August 2014, he has been a Professor with the Department of Information and Communication Technology and (by courtesy) the Department of Engineering, University of Agder, Agder, Norway, where he leads the Center Intelligent Signal Processing and Wireless Networks. Since April 2021, he has been also an Adjunct Chief Research Scientist and Research Professor with the Simula Metropolitan Center for Digital Engineering (SimulaMet), Norway, where he leads the Department of Signal and Information Processing for Intelligent Systems (SIGIPRO). His research interests include the general areas of distributed in-network signal processing and collective intelligence, data science and machine learning, optimization, networked cyber-physical systems and artificial intelligence for next generation wireless networks. He has been the Senior Area Editor of IEEE Transactions on Signal Processing since 2016 and a Member of the Technical Program Committees for several ACM & IEEE International Conferences. At USC, Dr. Beferull-Lozano was the recipient of the several awards including the Best Ph.D. Thesis paper Award in 2002, Outstanding Academic Achievement Award in 1999, and Best Paper Awards at several international conferences, such as IEEE DCOSS and IEEE DSLW and TOPPFORSK Grant Award from the Research Council of Norway, 2015. He is a Member of the Norwegian Academy of Science and Technology.