Appendix B

PAPER B

Title: Random Feature Approximation for Online Nonlinear Graph Topology Identification

Authors: **R. Money**, J. Krishnan, B. Beferull-Lozano

Conference: IEEE International Workshop on Machine Learning for Signal Processing 2021

Random Feature Approximation for Online Nonlinear Graph Topology Identification

R. Money, J. Krishnan, B. Beferull-Lozano

Abstract: Online topology estimation of graph-connected time series is challenging, especially since the causal dependencies in many real-world networks are nonlinear. In this paper, we propose a kernel-based algorithm for graph topology estimation. The algorithm uses a Fourier-based Random feature approximation to tackle the curse of dimensionality associated with the kernel representations. Exploiting the fact that the real-world networks often exhibit sparse topologies, we propose a group lasso based optimization framework, which is solve using an iterative composite objective mirror descent method, yielding an online algorithm with fixed computational complexity per iteration. The experiments conducted on real and synthetic data show that the proposed method outperforms its competitors.

B.1 Introduction

The amount of data generated from interconnected networks such as sensor networks, financial time-series, brain-networks, etc., are increasing rapidly. Extraction of meaningful information from such interconnected data, represented in the form of a graph can have many practical applications such as, signal denoising [62], change point detection [54], time series prediction [53], etc. Many of the functional relationships in such networks are causal and identification of this causal graph structure is termed topology identification. Many real world causal systems can be well described using vector autoregressive model (VAR) as naturally most of the dependencies are time-lagged in nature. Moreover under causal sufficiency, VAR causality implies well known Granger causality [63].

Topology identification based on the linear VAR model has been well-studied. In [48], an efficient way to estimate linear VAR coefficients from streaming data is proposed. However, such linear VAR models fail to capture the real-world nonlinear dependencies.A novel nonlinear VAR topology identification is proposed in [11] in which, the kernels are used to linearize the nonlinear dependencies by mapping them to a higher-dimensional Hilbert space. However, being a batch-based approach, [11] is computationally expensive and is not suitable for identifying the time-varying topologies.

The above shortcomings are tackled by kernel-based online algorithms [61], [21]. In [21], sparse VAR coefficients are recursively estimated using a composite objective mirror decent (COMID) approach, whereas [61] uses functional stochastic gradient descent (FSGD), followed by soft-thresholding. However, the kernel-based representations have a major drawback of unaffordable growth of computational complexity and memory requirement, which is commonly known as the "curse of dimensionality". Both [61] and [21] propose to circumvent this issue by restricting the numeric calculation to a limited number of time-series samples using a time window, which results in suboptimal performance.

A standard procedure to address the curse of dimensionality is to invoke the kernel dictionaries [64]. Often, the dictionary elements are selected based on a budget maintaining strategy. In large-scale machine learning problems, the dictionary size can go prohibitively high in order to maintain the budget. Recently, the random feature (RF) approximation [16] techniques are gaining popularity in approximating the kernels, which are shown to yield promising results compared to the budget maintaining strategies [16, 42].

In this work, we use RF approximation to avoid the curse of dimensionality in learning nonlinear VAR models. We approximate shift-invariant Gaussian kernels using a fixed number of random Fourier features. The major contributions of this paper are **i)** formulation of a kernel-based optimization framework in the function space, **ii)** reformulation of **i)** to a parametric optimization using RF approximation, and **iii)** an online algorithm to estimate the sparse nonlinear VAR coefficients using COMID updates. We provide numerical results showing the proposed method outperforms the state-of-the-art topology identification algorithms.

B.2 Kernel Representation

Consider a multi-variate time series with N nodes. Let $y_n[t]$ be the value of time series at time $t = 0, 1, \ldots, T-1$ observed at node $1 \leq n \leq N$. A P-th order nonlinear VAR model assuming additive functional dependencies can be formulated as

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

where $f_{n,n'}^{(p)}$ is the function that encodes the nonlinear causal influence of the p-lagged data at node n' on the node n, $a_{n,n'}^{(p)}$ is the corresponding entry of the graph adjacency matrix, and $u_n[t]$ is the observation noise. Considering the model (B.1), topology identification can be defined as the estimation of the functional dependencies $\left\{a_{n,n'}^{(p)} f_{n,n'}^{(p)}(.)\right\}_{p=1}^{P}$ for $n = 1, 2, \ldots, N$ from the observed time series $\{y_n[t]\}_{n=1}^{N}$.

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

$$\mathcal{H}_{n'}^{(p)} := \left\{ f_{n,n'}^{(p)} \mid f_{n,n'}^{(p)}(y) = \sum_{t=0}^{\infty} \beta_{n,n',t}^{(p)} \, \kappa_{n'}^{(p)}(y, y_{n'}[t-p]) \right\}, \tag{B.2}$$

where $\kappa_{n'}^{(p)} : \mathbb{R}\times\mathbb{R} \rightarrow \mathbb{R}$ is the kernel associated with the Hilbert space. The kernel measures the similarity between data points y and $y_{n'}[t-p]$. Referring to (B.2), evaluation of the functional $f_{n,n'}^{(p)}$ at y can be represented as the linear combination of the similarities between y and the data points $\{y_{n'}[t-p]\}_{t=0}^{t=\infty}$, with weights $\beta_{n,n',t}^{(p)}$. The inner product, $\langle \kappa_{n'}^{(p)}(y,x_1), \kappa_{n'}^{(p)}(y,x_2)\rangle := \sum_{t=0}^{\infty} \kappa_{n'}^{(p)}(y[t],x_1)\kappa_{n'}^{(p)}(y[t],x_2)$, is defined in the Hilbert space 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 induces a norm, $\|f_{n,n'}^{(p)}\|^2_{\mathcal{H}_{n'}^{(p)}} = \sum_{t=0}^{\infty}\sum_{t'=0}^{\infty} \beta_{n,n',t}^{(p)}\, \beta_{n,n',t'}^{(p)}\, \kappa_{n'}^{(p)}(y_n[t], y_n[t'])$. We refer to [37] for further reading on RKHS.

For a particular node n, the estimates of $\left\{f_{n,n'}^{(p)} \in \mathcal{H}_{n'}^{(p)}\right\}_{n',p}$ are obtained by solving the functional optimization problem:

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

It is to be noted that in (B.3), the functions $\{f_{n,n'}^{(p)}\}$ belong to the RKHS defined in (B.2), which is an infinite dimensional space. However, by resorting to the Representer Theorem [38], the solution of (B.3) can be written using a finite number of data samples:

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

Notice that the number of coefficients required to express the function increases with the number of data samples. In the recent works [61], [21], this problem is solved by using a time window to fix the number of data points, resulting in suboptimality. However, in this work in align with [42] and [43], we use RF approximation to tackle the dimensionality growth.

B.3 Random Feature Approximation

To invoke RF approximation, we assume the kernel to be shift-invariant, i.e., it satisfies the property $\kappa_{n'}^{(p)}(y_{n'}[\tau-p]), y_{n'}[t-p]) = \kappa_{n'}^{(p)}(y_{n'}[\tau-p]) - y_{n'}[t-p])$. Bochner's theorem [30] states that every shift-invariant kernel can be represented as an inverse Fourier transform of a probability distribution. Hence the kernel evaluation can be expressed as

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

where $\pi_{\kappa_{n'}^{(p)}}(v)$ is the probability density function which depends on type of the kernel, and v is the random variable associated with it. If sufficient amount of iid samples $\{v_i\}_{i=1}^{D}$ are collected from the distribution $\pi_{\kappa_{n'}^{(p)}}(v)$, the real ensemble mean in (B.5) can be expressed as a sample mean:

$$\hat{\kappa}_{n'}^{(p)}(y_{n'}[\tau-p]), y_{n'}[t-p]) = \frac{1}{D}\sum_{i=1}^{D} e^{jv_i(y_{n'}[\tau-p])-y_{n'}[t-p])}, \tag{B.6}$$

irrespective of the distribution $\pi_{\kappa^{(p)}_{n'}}(v)$. Note that the unbiased estimate of kernel evaluation in (B.6) involves a summation of fixed D number of terms. In general, computing the probability distribution corresponding to a kernel is a difficult task. In this work the kernel under consideration is Gaussian; for a Gaussian kernel k_σ with variance σ^2, it is well known that the Fourier transform is a Gaussian with variance σ^{-2}. Considering the real part of (B.6), which is also an unbiased estimator, (B.5) can be approximated as

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

$$\textit{where } \boldsymbol{z_v}(x) = \frac{1}{\sqrt{D}}[\sin v_1 x, \dots, \sin v_D x, \cos v_1 x, \dots, \cos v_D x]^\top. \tag{B.8}$$

Subsisting (B.7) in (B.4), we obtain a fixed dimension ($2D$ terms) approximation of the function $\hat{f}^{(p)}_{n,n'}$:

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

where ${\boldsymbol{\theta}^{(p)}_{n,n'}}^\top = \sum_{t=p}^{p+T-1} \beta^{(p)}_{n,n',(t-p)} \boldsymbol{z_v}\left(y_{n'}[\tau-p]\right)^\top = [\theta^{(p)}_{n,n',1}, \dots, \theta^{(p)}_{n,n',2D}] \in \mathbb{R}^{2D}$. For the sake of clarity, in the succeeding steps, we define the following notation:

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

$$\boldsymbol{z_v}\left(y_{n'}[\tau-p]\right) = [z^{(p)}_{n',1}(\tau), \dots z^{(p)}_{n',2D}(\tau)]^\top \in \mathbb{R}^{2D}, \tag{B.11}$$

where $\alpha^{(p)}_{n,n',d} = \theta^{(p)}_{n,n',d} a^{(p)}_{n,n'}$. The functional optimization (B.3) is reformulated as a parametric optimization problem using (B.9):

$$\left\{\widehat{\alpha}^{(p)}_{n,n',d}\right\}_{n',p,d} = \arg \min_{\left\{\alpha^{(p)}_{n,n',d}\right\}} \mathcal{L}^n\left(\alpha^{(p)}_{n,n',d}\right), \tag{B.12}$$

where

$$\mathcal{L}^n\left(\alpha^{(p)}_{n,n',d}\right) := \sum_{\tau=P}^{T-1} \frac{1}{2}\left[y_n[\tau] - \sum_{n'=1}^{N}\sum_{p=1}^{P}\sum_{d=1}^{2D} \alpha^{(p)}_{n,n',d}\; z^{(p)}_{n',d}(\tau)\right]^2. \tag{B.13}$$

For convenience, optimization parameters $\left\{\alpha^{(p)}_{n,n',d}\right\}$ and $\left\{z^{(p)}_{n',d}(\tau)\right\}$ are stacked in the lexicographic order of the indices p, n', and d to obtain the vectors $\boldsymbol{\alpha}_n \in \mathbb{R}^{2PND}$ and $\boldsymbol{z}_\tau \in \mathbb{R}^{2PND}$, respectively, and (B.12) is rewritten as

$$\widehat{\boldsymbol{\alpha}}_n = \arg\min_{\boldsymbol{\alpha}_n} \mathcal{L}^n\left(\boldsymbol{\alpha}_n\right), \tag{B.14}$$

$$\text{where } \mathcal{L}^n(\boldsymbol{\alpha}_n) = \frac{1}{2}\sum_{\tau=P}^{T-1}\left[y_n[\tau] - \boldsymbol{\alpha}_n^\top \boldsymbol{z}_\tau\right]^2 \tag{B.15}$$

Now, in order to avoid overfitting, we propose a regularized optimization framework:

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

where $\lambda \geq 0$ is the regularization parameter and $\boldsymbol{\alpha}_{n,n'}^{(p)} = (\alpha_{n,n',1}^{(p)}, \alpha_{n,n',2}^{(p)}, \ldots, \alpha_{n,n',2D}^{(p)}) \in \mathbb{R}^{2D}$. The second term in (B.16) is a *group-lasso* regularizer, which promote a *group-sparse structure* in $\boldsymbol{\alpha}_{n,n'}^{(p)}$, supported by the assumption that most of the real world dependencies are sparse in nature.

However, notice that the batch formulation in (B.16) has some significant limitations: **i)** requirement of complete batch of data points before estimation, **ii)** inability to track time varying topologies, and **iii)** explosive computational complexity when T is large even if RF approximation is used. To mitigate these problems, we adopt an online optimization strategy, which is explained in the following section.

B.4 Online Topology Estimation

In this case, we replace the batch loss function $\mathcal{L}^n(\boldsymbol{\alpha}_n)$ in (B.16) with the stochastic (instantaneous) loss function $l_t^n(\boldsymbol{\alpha}_n) = \frac{1}{2}[y_n[t] - \boldsymbol{\alpha}_n^\top \boldsymbol{z}_t]^2$:

$$\widehat{\boldsymbol{\alpha}}_n = \arg\min_{\boldsymbol{\alpha}_n} l_t^n\left(\boldsymbol{\alpha}_n\right) + \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2. \tag{B.17}$$

Notice that the sparsity promoting group lasso regularizer is non-differentiable. The use of online subgradient descent (OSGD) is not advisable in this situation as it linearizes the entire objective function and fails to provide sparse iterates. To avoid this limitation of OSGD, we use the composite objective mirror descent (COMID) [39] algorithm which resembles the nature of proximal methods, hence improving convergence. The online COMID update can be written as

$$\boldsymbol{\alpha}_n[t+1] = \arg\min_{\boldsymbol{\alpha}_n} J_t^{(n)}(\boldsymbol{\alpha}_n), \tag{B.18}$$

$$\text{where } J_t^{(n)}(\boldsymbol{\alpha}_n) \triangleq \nabla \ell_t^n(\boldsymbol{\alpha}_n[t])^\top \left(\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]\right)$$

$$+ \frac{1}{2\gamma_t}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]\|_2^2 + \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2. \tag{B.19}$$

In (B.19) $\boldsymbol{\alpha}_n[t] \in \mathbb{R}^{2PND}$ denotes the estimate of $\boldsymbol{\alpha}_n$ at time t. The first term in equation (B.19) is the gradient of the loss function $l_t^n(\boldsymbol{\alpha}_n)$, the second and third term are Bergman divergence and sparsity promoting regularizer respectively. The Bregman divergence is included to improve the stability of algorithm from adversaries by constraining $\boldsymbol{\alpha}_n[t+1]$ to be close to $\boldsymbol{\alpha}_n[t]$. The Bregman divergence $B(\boldsymbol{\alpha}_n, \boldsymbol{\alpha}_n[t]) = \frac{1}{2}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n[t]\|_2^2$ chosen in such a way that the COMID update has a closed form solution [40] and γ_t is the corresponding step size. The gradient in (B.19) is evaluated as

$$\mathbf{v}_n[t] := \nabla \ell_t^n(\boldsymbol{\alpha}_n[t]) = \boldsymbol{z}_t \left(\boldsymbol{\alpha}_n^\top \boldsymbol{z}_t - y_n[t]\right) \tag{B.20}$$

Expanding the objective function in (B.19) and omitting the constants leads to the following formulation:

$$\begin{aligned}J_t^{(n)}(\boldsymbol{\alpha}_n) &\propto \frac{\boldsymbol{\alpha}_n^\top \boldsymbol{\alpha}_n}{2\gamma_t} + \boldsymbol{\alpha}_n^\top \left(\mathbf{v}_n[t] - \frac{1}{\gamma_t}\boldsymbol{\alpha}_n[t]\right) + \lambda \sum_{n'=1}^{N} \sum_{p=1}^{P} \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2 \\ &= \sum_{n'=1}^{N} \sum_{p=1}^{P} \Bigg[\frac{\boldsymbol{\alpha}_{n,n'}^{(p)^\top} \boldsymbol{\alpha}_{n,n'}^{(p)}}{2\gamma_t} + \boldsymbol{\alpha}_{n,n'}^{(p)^\top} \left(\mathbf{v}_{n,n'}^{(p)}[t] - \frac{1}{\gamma_t}\boldsymbol{\alpha}_{n,n'}^{(p)}[t]\right) \\ &\qquad + \lambda \|\boldsymbol{\alpha}_{n,n'}^{(p)}\|_2 \Bigg]. \end{aligned} \tag{B.21}$$

A closed form solution for (B.18) using (B.21) is obtained via the multidimensional shrinkage-thresholding operator [41]:

$$\begin{aligned}\boldsymbol{\alpha}_{n,n'}^{(p)}[t+1] = &\left(\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - \gamma_t \mathbf{v}_{n,n'}^{(p)}[t]\right) \times \\ &\left[1 - \frac{\gamma_t \lambda}{\|\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - \gamma_t \mathbf{v}_{n,n'}^{(p)}[t]\|_2}\right]_+ ,\end{aligned} \tag{B.22}$$

where $[x]_+ = \max\{0, x\}$. The first term $\boldsymbol{\alpha}_{n,n'}^{(p)}[t] - \gamma_t \mathbf{v}_{n,n'}^{(p)}[t]$ in (B.22) forces the stochastic gradient update of $\boldsymbol{\alpha}_{n,n'}^{(p)}$ in a way to descend instantaneous loss function $l_t^n(\boldsymbol{\alpha}_n)$ and the second term in (B.22) enforces group sparsity of $\boldsymbol{\alpha}_{n,n'}^{(p)}$. Note that the close form solution (B.22) is separable in n' and p.

The proposed algorithm, termed as *Random Feature based Nonlinear Topology Identification via Sparse Online learning* (RF-NLTISO), is summarized in **Algorithm 6**.

Algorithm 6: RF-NLTISO Algorithm

```
Result: $\boldsymbol{\alpha}_{n,n'}^{(p)}, for\ n, n' = 1,..,N$ and $p = 1,..,P$
Store $\{\boldsymbol{y}_n[t]\}_{t=1}^{P}$,
Initialize $\lambda$, $\gamma$, $D$ (heuristically chosen) and kernel parameters depending
 on the type of the kernel.
for $t = P, P+1, \dots$ do
    Get data samples $y_n[t]$, $\forall n$ and compute $\boldsymbol{z}_\tau$
    for $n = 1, \dots, N$ do
        compute $\mathbf{v}_n[t]$ using (B.20)
        for $n' = 1, \dots, N$ do
            compute $\boldsymbol{\alpha}_{n,n'}^{(p)}[t+1]$ using (B.22)
        end
    end
end
```

B.5 Experiments

We compare the performance of the proposed algorithm, RF-NLTISO, with the the state-of-the-art online topology estimation algorithms. Experiments shown in this section are conducted using 1) synthetic datasets with topologies having different transition patterns and 2) real datasets collected from Lundin's offshore oil and Gas platform. For the performance comparison, we choose TIRSO [48] and NL-TISO [21] algorithms, which are the state-of-the-art counterparts of RF-NLTISO, to the best of our knowledge. TIRSO is developed based on a linear VAR model assumption, whereas NL-TISO, a kernel-based topology estimation algorithm, is developed for nonlinear VAR models. Although a kernel-based functional stochastic gradient based algorithm [61] is also available, its performance has been shown to be inferior compared to NL-TISO [21].

B.5.1 Experiments using Synthetic Data

B.5.1.1 Topology with switching edges

We generate a multi-variate time series using nonlinear VAR model (B.1) with $N = 5, P = 2$. An initial random graph with edge probability of 0.1 is generated and the graph adjacency coefficients $a^{(p)}_{n,n'}$ are drawn from a Uniform distribution $\mathcal{U}(0,\, 1)$. After every 1000 samples, one of the active (non-zero) edge disappears and another one appears randomly, which brings an abrupt change in the graph topology. The nonlinearity in (B.1) is introduced using a Gaussian kernel with variance 0.01 and the kernel coefficients are chosen randomly from a zero mean Gaussian distribution with variance 30. Note that the initial P data samples are generated randomly and rest of the data is generated using the model (B.1). The coefficients $\left\{\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\right\}$ are estimated using the proposed RF-NLTISO algorithm with a Gaussian kernel having variance 0.1 and number of random features $D = 50$. The hyper-parameters λ and γ are heuristically chosen as 0.1 and 1000, respectively. To visualize causal relationships, we compute the ℓ_2 norms $b^{(p)}_{n,n'}[t] = \|\boldsymbol{\alpha}^{(p)}_{n,n'}[t]\|_2$ and arrange them in a matrix similar to the graph adjacency matrix. A similar strategy is adopted for the NL-TISO and the TIRSO algorithms. The normalized version of true and the estimated dependencies at various time samples are shown in Fig. B.1, where in each subplot, the 5×5 dependency matrices corresponding to $p = 1$ and 2 are concatenated, resulting in a 10×5 size matrix. We normalized the coefficients by dividing each coefficients with highest value of coefficient in a pseudo adjacency matrix. From the Fig. B.1, it is clear that RF-NLTISO is able to perform equal or better compared to NL-TISO algorithm and clearly outperforms TIRSO.

Next we conduct the same experiments using RF-NLTISO with different numbers of random feature ($D \in \{10, 30, 50\}$). These experiments are repeated 1000 times to find probability of miss detection (P_{MD}) and false alarm (P_{FA}), which we define as

$$\begin{aligned} P_{\text{MD}}[t] &\triangleq \frac{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\nVdash\{\|\widehat{\boldsymbol{b}}^{(p)}_{n,n'}[t]\|_2 < \delta\}\nVdash\{\left\|\boldsymbol{\alpha}_{n,n'}\right\|_2 \geq \delta\}\right]}{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\nVdash\{\left\|\boldsymbol{\alpha}_{n,n'}\right\|_2 \geq \delta\}\right]}, \\ P_{\text{FA}}[t] &\triangleq \frac{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\nVdash\{\|\widehat{\boldsymbol{b}}^{(p)}_{n,n'}[t]\|_2 > \delta\}\nVdash\{\left\|\boldsymbol{\alpha}_{n,n'}\right\|_2 \leq \delta\}\right]}{\sum_{n\neq n'}\sum_{p=1}^{P}\mathbb{E}\left[\nVdash\{\left\|\boldsymbol{\alpha}_{n,n'}\right\|_2 \leq \delta\}\right]}. \end{aligned} \tag{B.23}$$

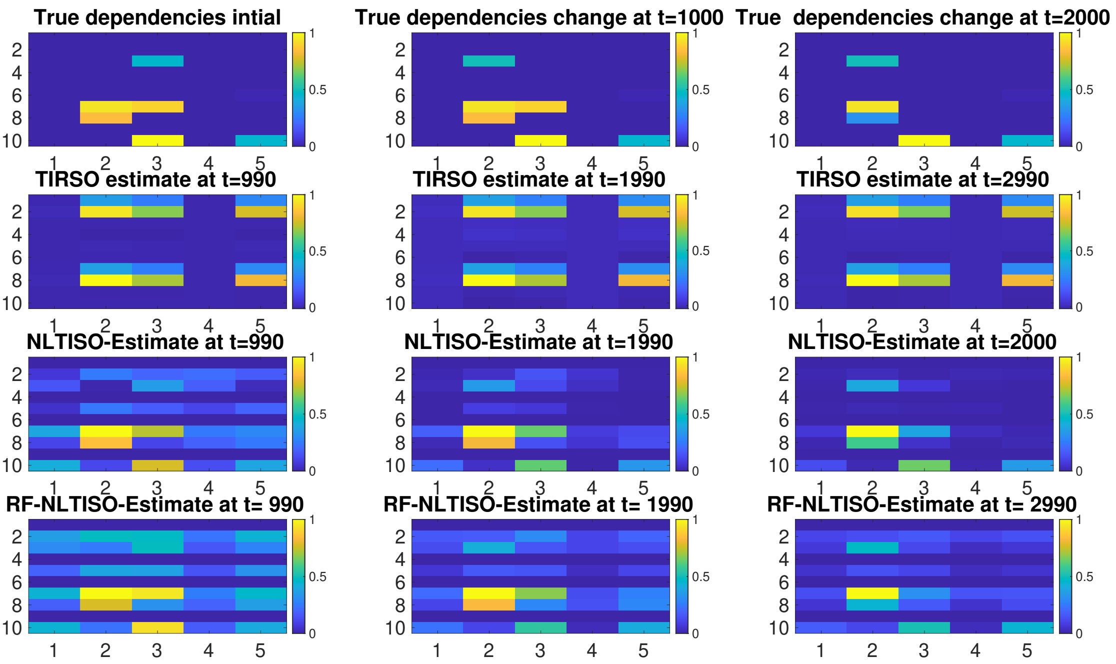


Figure B.1: Causal dependencies estimated using different algorithms compared with the true dependency.

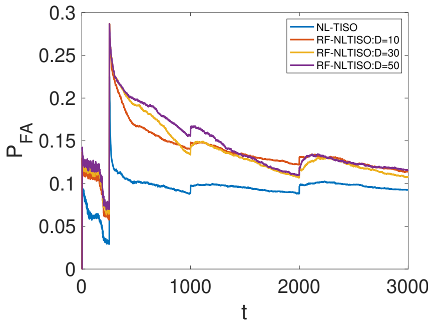


Figure B.2: Probability of False Alarm (P_{FA}).

0.12

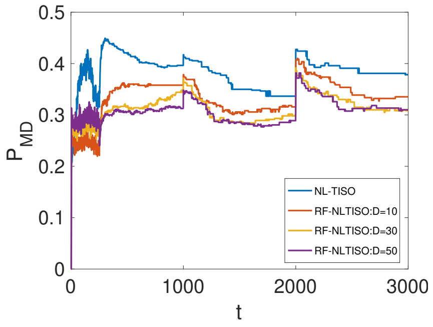
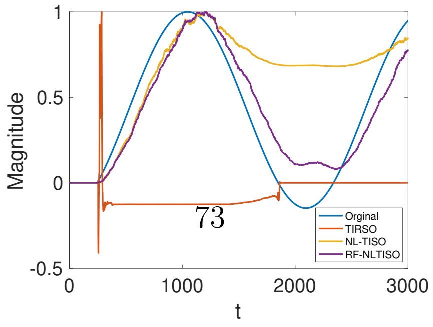


Figure B.3: Probability of Miss Detection (P_{MD}).



From Figs. B.2 and B.3, it is observed that for the given choices of D, P_{MD} is better for RF-NLTISO compared to NL-TISO; however, NL-TISO is performing better in terms of P_{FA}. Both the figures show an overshoot at the topology-switching time instances. It is also observed that for the proposed algorithm, P_{MD} decreases with D, whereas P_{FA} increases with D, which in turn suggests a tuning for D for an effective trade-off between P_{FA} and P_{MD}.

B.5.1.2 Slowly varying topology

We compare the performance of RF-NLTISO with the state-of-the-art algorithms using a slowly varying graph topology. The same experiment setup as discussed in B.5.1.1 is adopted with the following more slowly time varying topology:

$$a_{n,n'}^{(p)}[t+1] = a_{n,n'}^{(p)}[t] + 0.01\sin(0.03 * t) \tag{B.24}$$

The normalized values of one of the active edges is plotted in Fig. B.4. The figure also shows the normalized values of the corresponding estimated coefficients ($\widehat{b}_{n,n'}^{(p)}[t]$ for NL-TISO and RF-NLTISO and $\widehat{a}_{n,n'}^{(p)}[t]$ for TIRSO). From the figure, it can be observed that the RF-NLTISO estimates are closer to the true value compared to the estimates from the other two algorithms. In this example, the quality of TIRSO estimates lags considerably behind the kernel-based algorithms due to the fact that the underlying VAR model is nonlinear.

B.5.2 Experiments using Real Data

This section is dedicated to experiments using real data collected from Lundin's offshore oil and gas (O&G) platform Edvard-Grieg[1]. We have a multi-variate time series with 24 nodes; and the nodes corresponds to various temperature (T), pressure (P), or oil-level (L) sensors. The sensors are placed in the separators of decantation tanks that separate oil, gas, and water. The time series are obtained by uniformly sampling the sensor readings with a sampling rate of $5s$. We assume that hidden logic dependencies are present in the network due to various physical connections and various control actuators. The data obtained from the network is normalized by making it a zero mean unit variance signal, before applying the algorithm. The causal dependencies are learned using RF-NLTISO with $D = 10, 50, 100$ and a Gaussian kernel having a variance of 0.1 and with hyper parameter values $\lambda = 0.1$ and $\gamma = 10$. The signal is reconstructed using the estimated dependencies. Fig. B.6 shows the mean squared error (MSE), defined as $MSE(t) = \mathbb{E}((y_n(t) - \hat{y}_n(t))^2)$ for a particular sensor $n = 8$, of RF-NLTISO estimates in comparison with other algorithms. We observe that the RF-NLTISO estimates with random feature number $D \geq 50$ show better MSE performance compare to NL-TISO. The causality graph estimated by RF-NLTISO is shown in Fig. B.7.

One of the main attractiveness of RF-NLTISO is that even though it is a kernel-based algorithm, it has a fixed computational complexity throughout the online

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

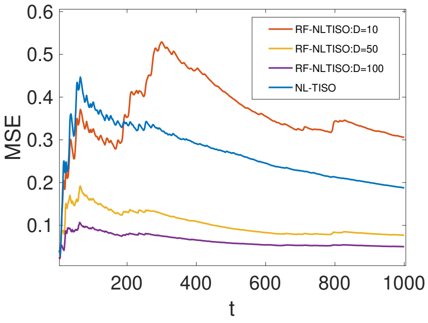


Figure B.6: MSE comparison of NL-TISO with RF-NLTISO.

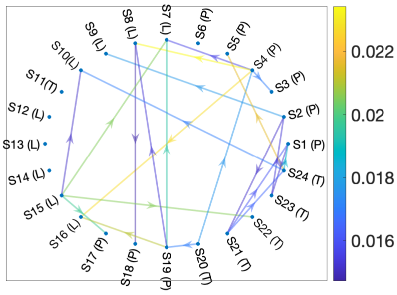


Figure B.7: Causality graph in oil and gas plant estimated by RF-NLTISO. P, T, L represent pressure, temperature, and oil level sensors, respectively.

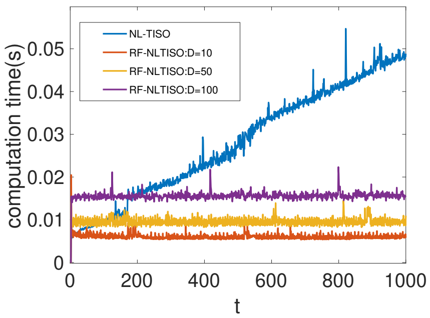


Figure B.8: Comparison of computation time of kernel-based algorithms.

Figure B.9: Real data.

iterations. To demonstrate this, in Fig. B.8, we plot the computation time required to estimate the coefficients at each time instant by NL-TISO and RF-NLTISO with different values of D. The experiment is conducted in a machine with processor 2.4 GHz 8-core Intel Core i9 and 16GB 2667 MHz DDR4 RAM. Fig. B.8 shows that the computation time of NL-TISO increases considerably with time but that of RF-NLTISO remains more or less constant for a particular value of D.

B.6 Conclusion

We propose a kernel-based online topology identification method for interconnected networks of time-series with additive nonlinear dependencies. In this work, the curse of dimensionality associated with kernel representation is tackled using random feature approximation. Assuming that the real-world dependencies are sparse, we use composite objective mirror decent update to estimate the online sparse causality graph. The effectiveness of the proposed algorithm is illustrated through experiments conducted on synthetic and real data, which shows that the algorithm outperforms the state-of-the-art competitors. We devote the convergence and stability analysis of the proposed algorithm to our future work.