

From Fourier to Koopman: Spectral Methods for Long-term Time Series Prediction

Henning Lange

*Department of Applied Mathematics
University of Washington
Seattle, WA 98195-4322, USA*

HELANGE@UW.EDU

Steven L. Brunton

*Department of Mechanical Engineering
University of Washington
Seattle, WA 98195-4322, USA*

SBRUNTON@UW.EDU

J. Nathan Kutz

*Department of Applied Mathematics
University of Washington
Seattle, WA 98195-4322, USA*

KUTZ@UW.EDU

Editor: Ryan Adams

Abstract

We propose spectral methods for long-term forecasting of temporal signals stemming from linear and nonlinear quasi-periodic dynamical systems. For linear signals, we introduce an algorithm with similarities to the Fourier transform but which does not rely on periodicity assumptions, allowing for forecasting given potentially arbitrary sampling intervals. We then extend this algorithm to handle nonlinearities by leveraging Koopman theory. The resulting algorithm performs a spectral decomposition in a nonlinear, data-dependent basis. The optimization objective for both algorithms is highly non-convex. However, expressing the objective in the frequency domain allows us to compute global optima of the error surface in a scalable and efficient manner, partially by exploiting the computational properties of the Fast Fourier Transform. Because of their close relation to Bayesian Spectral Analysis, uncertainty quantification metrics are a natural byproduct of the spectral forecasting methods. We extensively benchmark these algorithms against other leading forecasting methods on a range of synthetic experiments as well as in the context of real-world power systems and fluid flows.

1. Introduction

Forecasting of time series data is fundamental to almost every discipline in science and engineering, with applications in oceanography, epidemiology, meteorology, economics, etc (Armstrong, 1985). Indeed, *time-series analysis* has been historically a core discipline in statistical sciences where prediction accuracy and uncertainty quantification are critical for characterizing the underlying complex dynamical system measured. Such future state predictions also play a fundamental role in control theory where improved forecasting can in turn improve the latency time of control laws (Box et al., 2015). While there are a host of effective

short-time forecasting methods (Hamilton, 1994), generic mathematical architectures that allow for stable long-term forecasts, while mitigating compounding errors, are lacking.

The fundamental problem of forecasting has generated a diverse set of approaches for producing future state predictions and uncertainty metrics. Most of the pre-existing approaches are of the following type: a parameterized function $f_{\Theta} : \mathbb{R}^{T' \times N} \rightarrow \mathbb{R}^N$ is introduced and tasked with predicting the next temporal snapshot given a collection of previous ones by minimizing some loss function with respect to the parameters Θ . Prominent members of this family of approaches include Box-Jenkins models (Box et al., 2015; Pankratz, 2009), for example ARIMA or SARIMAX, and several neural network approaches, including Recurrent Neural Networks, such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Units (Chung et al., 2014) and most Echo State Networks (Jaeger, 2007), as well as Reservoir Computing approaches (Paquot et al., 2012) and hybrid approaches that combine wavelets and Neural Networks (Jiang and Adeli, 2005). Since f_{Θ} usually only allows one to obtain an estimate of the value the signal will take in the next time step, in order to allow for long-term forecasting, estimates must be recursively fed back into f_{Θ} . The main drawback of approaches that recursively perform one-step-ahead predictions, i.e. an approach that iteratively performs a succession of local decisions, is that errors compound over time.

In this work, ideas from operator theory, signal processing, and machine learning are combined to (i) derive generic and, under some conditions, optimal algorithms for long-term forecasting that do not require knowledge of the dynamics and that mitigate compounding errors, and (ii) derive practical error bounds and uncertainty metrics of the resulting algorithms as functions of noise and amounts of data provided. Because the contributions in this work touch multiple disciplines, they can be understood from different perspectives.

From the viewpoint of signal processing, we leverage *Bayesian Spectral Analysis* (Bretthorst, 2013) and derive an algorithm that performs a Fourier decomposition in an optimal nonlinear oscillatory basis. This nonlinear basis is learned from data and is optimal in the least-squares sense. Because of a symmetry relationship, we can leverage the *Fast Fourier Transform* to obtain model parameters in a fast and scalable way.

When adopting a Machine Learning perspective, because the theoretical and practical contributions of this work are primarily based on Koopman theory (Mezić, 2015), they can be understood by analogy to Cover’s theorem, which enabled the breakthroughs that have led to the development of Kernel Methods (Scholkopf and Smola, 2001) and Deep Learning (LeCun et al., 2015). Succinctly, Cover’s theorem states that for any data set of finite length there always exists a nonlinear transformation such that in the transformed space, the data can be described by linear methods, i.e. it is *linearly separable*. Analogously, Koopman theory postulates that any nonlinear dynamical system can be lifted by a time-invariant but nonlinear function into a space in which its time evolution can be described by linear methods, i.e. by a linear dynamical system (Kutz et al., 2016). This analogy is depicted graphically in Fig. 1.

For finite amounts of data, Cover’s and Koopman’s theorem can be made to hold in a trivial way by projecting the data into an N dimensional space, where N denotes the number of data points. For Cover’s theorem, a nonlinear function can be devised that maps the n -th data point to the indicator vector that is 0 everywhere except for the n -th position, where it is 1. By doing so, every finite data set can be made linearly separable by

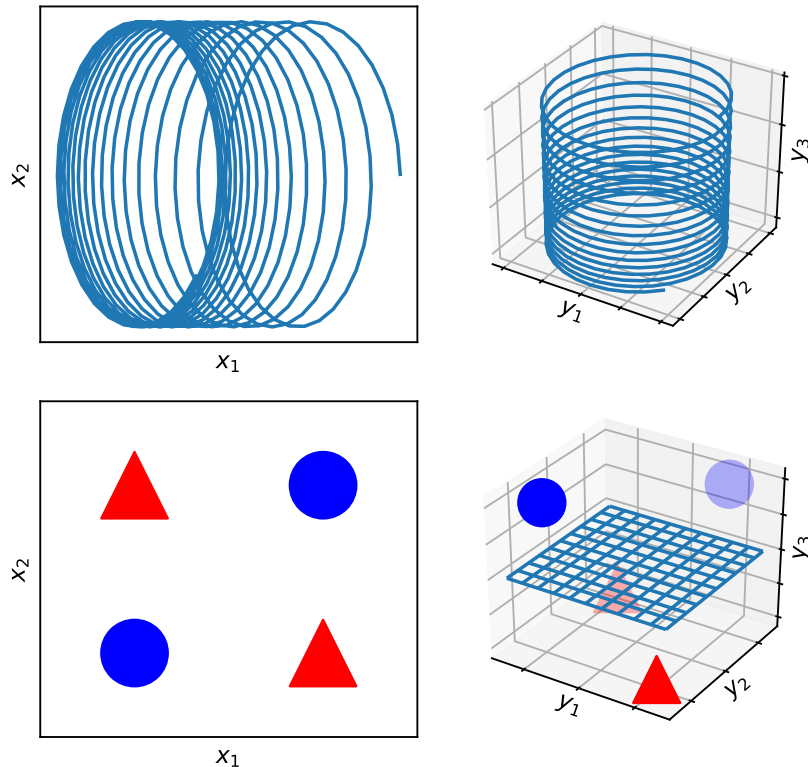


Figure 1: Top: Left shows the phase portrait of a nonlinear system that can be described as a linear system in three dimensions, as shown in the right panel. Bottom: The *exclusive-or* problem does not have a linear solution in 2D, but does in 3D.

constructing a weight vector $\mathbf{w} \in \mathcal{R}^N$ that is 1 for all data points belonging to one class and 0 for the other, i.e. by simply enumerating the training set. Similarly, Koopman’s theorem can be made true in an N dimensional space by the means of the discrete Fourier transform. Since sine and cosine pairs are linear dynamical systems and every sequence of N equidistantly sampled data points can be explained by N equidistant Fourier modes. However, both trivial solutions are not useful in practical scenarios since they do not allow for generalization. Note that the inability of the trivial solution of Koopman’s theorem to generalize stems from implicit periodicity assumptions of the discrete Fourier transform. We will show later how these implicit periodicity assumptions can be broken in order to allow for generalization.

We leverage Koopman theory for forecasting by constructing the linear *Koopman operator* that characterizes the time evolution of a dynamical system, even chaotic systems. However in practice, such chaotic systems result in a Koopman operator that is infinite dimensional. In this work, we focus our investigation on systems for which a finite dimensional Koopman operator exists (Brunton et al., 2016b), and furthermore restrict the scope to systems that preserve energy, or persist over long times, i.e. *quasi-periodic* systems. Mathematically speaking, we require the following property of the time series data x_t :

$$\lim_{T \rightarrow \infty} \mathcal{F}[f^{-1}(x_1), \dots, f^{-1}(x_T)] = \sum_{n=1}^N \delta(\omega - k_n)$$

where \mathcal{F} is the Fourier transform, f is a function, and f^{-1} its pre-image, and finite N . When systems fulfill these properties, we show empirically that stable long-term forecasting can be accomplished.

2. Fourier and Forecasting

When Fourier studied heat transfer in the early 19th century, he discovered that solutions to the heat equation can be simply represented as a superposition of sines and cosines (Fourier, 1822). This finding was generalized and ultimately resulted in what is known today as the Fourier Transform. In the 1960s, Cooley and Tukey (re)discovered the Fast Fourier Transform (FFT), which allows for the computation of coefficients of the discrete Fourier transform in $\mathcal{O}(T \log T)$ given a signal sampled at T equidistant points in time (Cooley and Tukey, 1965). The computational speed up relies on evaluating the Fourier transform at integer multiples of $\frac{1}{T}$, which in turn implicitly assumes that the signal is periodic with period T , since

$$x_{t+T} = \sum_{n=1}^N a_n e^{-j2\pi(t+T)\frac{n}{T}} = x_t$$

This assumption, unfortunately, makes the FFT unsuitable for forecasting anything but strictly T -periodic signals. However, in the following, we will treat the FFT as a computationally efficient optimization tool that, in conjunction with gradient descent, allows us to fit a linear dynamical system y_t to an observed signal x_t . The strategy is as follows: Given an optimization problem, we will derive an expression for a loss as a function of model parameters in the frequency domain and then use the FFT to evaluate the loss and pick optimal parameters. This strategy is similar to the initial idea of Fourier, i.e. to express an otherwise difficult problem in the frequency domain in which finding a solution is significantly easier. As we will show later, the difficulty of finding a solution in time-domain stems from the fact that the optimization objective is highly non-convex. However, note that in order to leverage the computational efficiency of the FFT, implicit periodicity assumption are being made. While these periodicity assumptions bring about computational efficiency, they prohibit forecasting but they can easily be broken by subsequently applying gradient descent to improve an initial guess provided by the FFT.

Let $\mathbf{y}_t \in \mathbb{R}^m$ be measurements of a linear dynamical system, i.e. $\mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1}$. The goal is to infer an optimal linear mapping between a measured signal $\mathbf{x}_t \in \mathbb{R}^n$ and \mathbf{y}_t . Specifically, the goal is to solve:

$$\begin{aligned} & \text{minimize } E(\mathbf{A}, \mathbf{B}) = \sum_{t=1}^T (\mathbf{x}_t - \mathbf{A}\mathbf{y}_t)^2 \\ & \text{subject to } \mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1} \end{aligned}$$

Since we assume \mathbf{x}_t to be quasi-periodic, we in turn assume \mathbf{B} to have strictly imaginary eigenvalues. For the continuous time solution of \mathbf{y}_t , the following holds:

$$\mathbf{y}_t = \sum_{i=1}^n (\mathbf{l}_i \cdot \mathbf{y}_0) \mathbf{r}_i e^{\lambda_i t} \tag{1}$$

where \mathbf{l}_i , \mathbf{r}_i and λ_i denote the i th left and right eigenvector and the i th eigenvalue of \mathbf{B} , respectively. Note, because eigenvalues and eigenvectors of real-valued square matrices come in conjugate pairs, we can rewrite equation (1) as:

$$\begin{aligned} \mathbf{y}_t &= \sum_{i=1}^{n/2} (\mathbf{l}_i \cdot \mathbf{y}_0) \mathbf{r}_i e^{\lambda_i t} + ((\mathbf{l}_i \cdot \mathbf{y}_0) \mathbf{r}_i)^* e^{-\lambda_i t} \\ &= \sum_{i=1}^{n/2} \mathbf{b}_i e^{\lambda_i t} + \mathbf{b}_i^* e^{-\lambda_i t} \end{aligned}$$

Since we assume λ_i to be strictly imaginary, let $\lambda_i = j\omega_i$. We can further simplify:

$$\begin{aligned} \mathbf{y}_t &= \sum_{i=1}^{n/2} 2\operatorname{Re}(\mathbf{b}_i) \cos(\omega_i t) + 2\operatorname{Im}(\mathbf{b}_i) \sin(\omega_i t) \\ &= 2 \begin{bmatrix} \operatorname{Re}(\mathbf{b}_1) & \cdots & \operatorname{Re}(\mathbf{b}_{n/2}) & \operatorname{Im}(\mathbf{b}_1) & \cdots & \operatorname{Im}(\mathbf{b}_{n/2}) \end{bmatrix} \begin{bmatrix} \cos(\omega_1 t) \\ \vdots \\ \cos(\omega_{n/2} t) \\ \sin(\omega_1 t) \\ \vdots \\ \sin(\omega_{n/2} t) \end{bmatrix} \\ &= 2\bar{\mathbf{B}} \begin{bmatrix} \cos(\vec{\omega} t) \\ \sin(\vec{\omega} t) \end{bmatrix} \end{aligned}$$

Without loss of generality, because \mathbf{A} is a free-parameter of the optimization problem, we can omit $2\bar{\mathbf{B}}$ and absorb this into \mathbf{A} :

$$\mathbf{y}_t = \begin{bmatrix} \cos(\vec{\omega} t) \\ \sin(\vec{\omega} t) \end{bmatrix} := \Omega(\vec{\omega} t) \quad (2)$$

Thus, under the assumption that \mathbf{B} only has imaginary eigenvalues, we can rewrite the minimization problem in the following way:

$$\begin{aligned} L_i(\mathbf{A}, \vec{\omega}, t) &:= [(\mathbf{x}_t)_i - (\mathbf{A}\Omega(\vec{\omega} t))_i]^2 \\ L(\mathbf{A}, \vec{\omega}, t) &= \|\mathbf{x}_t - \mathbf{A}\Omega(\vec{\omega} t)\|_2^2 && \text{(temporally local loss)} \\ &= \sum_{i=1}^n L_i(\mathbf{A}, \vec{\omega}, t) \\ E(\mathbf{A}, \vec{\omega}) &= \sum_{t=1}^T L(\mathbf{A}, \vec{\omega}, t) = \sum_{t=1}^T \sum_{i=1}^n L_i(\mathbf{A}, \vec{\omega}, t) && \text{(global loss)} \quad (3) \end{aligned}$$

The question arises which optimization procedure to employ to solve (3) for $\vec{\omega}$. Note that the assumption of suitable convexity required for gradient descent to perform well does

not hold. This can be understood by realizing that the sensitivity of the loss with respect to $\vec{\omega}$ grows linearly in time which in turn can be understood easily by investigating the gradients of the loss with respect to $\vec{\omega}$.

$$\frac{\partial L(\mathbf{A}, \vec{\omega}, t)}{\partial \vec{\omega}} = 2t \begin{bmatrix} -\sin(\vec{\omega}t) \\ \cos(\vec{\omega}t) \end{bmatrix} [\mathbf{x}_t - \mathbf{A}\Omega(\vec{\omega}t)]$$

This implies that the magnitude of gradients grows linearly with time which in turn entails that gradient descent disproportionately considers the loss function of temporal snapshots close to T . This effect can also be understood by realizing that $L(A, \vec{\omega}, t)$ is periodic in $\frac{2\pi}{t}$ with respect to $\vec{\omega}$ which follows directly from the properties of the sine and cosine functions, since:

$$\begin{aligned} L(\mathbf{A}, \vec{\omega} + \frac{2\pi}{t}, t) &= \|\mathbf{x}_t - \mathbf{A}\Omega((\vec{\omega} + \frac{2\pi}{t})t)\|_2^2 \\ &= \|\mathbf{x}_t - \mathbf{A}\Omega(\vec{\omega}t)\|_2^2 \\ &= L(\mathbf{A}, \vec{\omega}, t) \end{aligned}$$

Note that this entails that the temporally local loss functions close to T will oscillate rapidly and therefore create local minima in which gradient descent will get stuck.

In order to overcome these issues, we will derive an algorithm that under some conditions finds global minimizers by performing coordinate descent on $E(A, \vec{\omega})$. Coordinate descent updates model parameters in a sequential fashion starting with an initial guess of $\vec{\omega}^{(0)}$.

$$\omega_i^{(k)} = \arg \min_{\omega_i} E(\omega_1^{(k)}, \dots, \omega_{i-1}^{(k)}, \omega_i, \omega_{i+1}^{(k-1)}, \dots, \omega_m^{(k-1)})$$

Under some conditions (Tseng, 2001), for $k = 1, 2, 3, \dots$, $\vec{\omega}^{(k)}$ converges to a minimizer of E . We will now show that the update-step in the direction of ω_i can be performed in a computationally efficient manner using the FFT. Let $\mathcal{M}_i = \{A, \omega_1, \dots, \omega_{i-1}, \omega_{i+1}, \dots, \omega_m\}$, i.e. the set of all model parameters except ω_i . We will derive an analytic expression for $E(\omega_i; \mathcal{M}_i)$ in the frequency domain which allows us, by applying the FFT, to compute the entire error surface and pick the optimal value for ω_i .

Let $L_l(\omega_i; \mathcal{M}_i, t)$ be the temporally local loss as a function of only ω_i , i.e. while keeping all other model parameters fixed and only considering the l th dimension of the loss and let $I(i) = \{1, \dots, i-1, i+1, \dots, i+n/2-1, i+n/2+1, \dots, n\}$ be the set of all indices sans i and $i+n/2 := i_2$, then:

$$\begin{aligned} L_l(\omega_i; \mathcal{M}_i, t) &= [(\mathbf{x}_t)_l - (\mathbf{A}\Omega(\vec{\omega}t))_l]^2 \\ &= [(\mathbf{x}_t)_l - \underbrace{\sum_{k \in I(i)} \mathbf{A}_{l,k} \Omega(\vec{\omega}t)_k - \mathbf{A}_{l,i} \cos(\omega_i t) - \mathbf{A}_{l,i_2} \sin(\omega_i t)}_{=R(i)_{t,l}}]^2 \end{aligned}$$

Note that $R(i)$ defines the residual that captures the difference between Ay_t and x_t but ignores all terms that involve ω_i .

$$\begin{aligned}
 E_l(\omega_i; \mathcal{M}_i) &:= \sum_t L_l(\omega_i; \mathcal{M}_i, t) \\
 &= \sum_t (\mathbf{R}(i)_{l,t} - \mathbf{A}_{i,l} \cos(\omega_i t) - \mathbf{A}_{i_2,l} \sin(\omega_i t))^2 \\
 &= \sum_t [\mathbf{R}(i)_{l,t}^2 - \mathbf{A}_{i,l}^2 \cos^2(\omega_i t) - \mathbf{A}_{i_2,l}^2 \sin^2(\omega_i t) \\
 &\quad - 2\mathbf{R}(i)_{l,t} \mathbf{A}_{i,l} \cos(\omega_i t) - 2\mathbf{R}(i)_{l,t} \mathbf{A}_{i_2,l} \sin(\omega_i t) \\
 &\quad + 2\mathbf{A}_{i,l} \mathbf{A}_{i_2,l} \cos(\omega_i t) \sin(\omega_i t)] \tag{4}
 \end{aligned}$$

We proceed by investigating this loss function at extrema of \mathbf{A} .

$$\begin{aligned}
 \frac{\partial E_l(\omega_i; \mathcal{M}_i)}{\partial \mathbf{A}_{i,l}} &= \sum_t -2\mathbf{A}_{i,l} \cos^2(\omega_i t) - 2\mathbf{R}(i)_{l,t} \cos(\omega_i t) + 2\mathbf{A}_{i_2,l} \cos(\omega_i t) \sin(\omega_i t) \\
 \frac{\partial E_l(\omega_i; \mathcal{M}_i)}{\partial \mathbf{A}_{i_2,l}} &= \sum_t -2\mathbf{A}_{i_2,l} \sin^2(\omega_i t) - 2\mathbf{R}(i)_{l,t} \sin(\omega_i t) + 2\mathbf{A}_{i,l} \cos(\omega_i t) \sin(\omega_i t)
 \end{aligned}$$

From $\frac{\partial E_l(\omega_i; \mathcal{M}_i)}{\partial \mathbf{A}_{i,l}} = 0$ and $\frac{\partial E_l(\omega_i; \mathcal{M}_i)}{\partial \mathbf{A}_{i_2,l}} = 0$, we can derive the following respectively:

$$\mathbf{A}_{i_2,l} \sum_t \cos(\omega_i t) \sin(\omega_i t) = \sum_t \mathbf{A}_{i,l} \cos^2(\omega_i t) + \mathbf{R}(i)_{l,t} \cos(\omega_i t) \tag{5}$$

$$\mathbf{A}_{i,l} \sum_t \cos(\omega_i t) \sin(\omega_i t) = \sum_t \mathbf{A}_{i_2,l} \sin^2(\omega_i t) + \mathbf{R}(i)_{l,t} \sin(\omega_i t) \tag{6}$$

Plugging both of these findings into (4) yields:

$$E_l(\omega_i; \mathcal{M}_i) = \sum_t [\mathbf{R}(i)_{l,t}^2 - \mathbf{R}(i)_{l,t} \mathbf{A}_{i,l} \cos(\omega_i t) - \mathbf{R}(i)_{l,t} \mathbf{A}_{i_2,l} \sin(\omega_i t)]$$

Since our goal is to employ the FFT to evaluate $E(\omega_i; \mathcal{M}_i)$, we assume that all $\omega_i = \frac{k}{T}$ with $k \in \mathbb{N}$. This entails that $\sum_t^T \cos^2(\omega_i t) = \sum_t^T \sin^2(\omega_i t) = T/2$, $\sum_t^T \cos(\omega_i t) \sin(\omega_i t) = 0$. We can then deduce from (5) and (6) respectively, that:

$$\begin{aligned}
 \mathbf{A}_{i,l} &= \frac{2\mathbf{R}(i)_{l,t} \cos(\omega_i t)}{T} \\
 \mathbf{A}_{i_2,l} &= \frac{2\mathbf{R}(i)_{l,t} \sin(\omega_i t)}{T}
 \end{aligned}$$

Therefore:

$$E_l(\omega_i; \mathcal{M}_i) = \sum_t^T \mathbf{R}(i)_{l,t}^2 - \frac{2}{T} \left(\sum_t^T \mathbf{R}(i)_{l,t} \cos(\omega_i t) \right)^2 - \frac{2}{T} \left(\sum_t^T \mathbf{R}(i)_{l,t} \sin(\omega_i t) \right)^2$$

Let $\hat{\mathbf{R}}(i)_{l,\omega_i}$ be the Fourier transform of $\mathbf{R}(i)_{l,t}$, then:

$$\begin{aligned} E(\omega_i; \mathcal{M}_i) &= \sum_l^n \left(\sum_t^T \mathbf{R}(i)_{l,t}^2 - \frac{2}{T} \text{Re}(\hat{\mathbf{R}}(i)_{l,\omega_i})^2 - \frac{2}{T} \text{Im}(\hat{\mathbf{R}}(i)_{l,\omega_i})^2 \right) \\ &= \|\mathbf{R}(i)\|_F - \sum_l^n \frac{2}{T} |\hat{\mathbf{R}}(i)_{l,\omega_i}|^2 \end{aligned}$$

Thus, there is a symmetry relationship between the error surface and the Fourier coefficients of the residual. This symmetry is sketched in Figure 2. In the following, we will exploit this symmetry by using the FFT to locate the valley of global optimum of $E(\omega_i; \mathcal{M}_i)$ with precision $1/T$ in a computationally efficient way. Note that precision of $1/T$ usually suffices to locate the global optimum for reasonably sized data set. If this precision should not suffice, it is always possible to oversample the FFT by appending zeros in time domain.

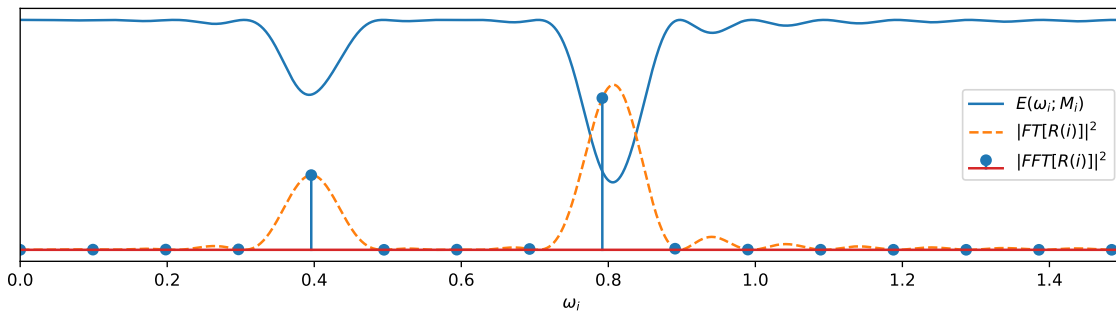


Figure 2: Symmetry between the squared error, i.e. $E(\omega_i; \mathcal{M}_i)$, and the Fourier transform of the residual. Colloquially speaking, the absolute value of the Fourier coefficients moved up and flipped on the x-axis constitutes the error surface.

2.1 Breaking periodicity

As laid out earlier, when computing the FFT of the residual, i.e. when evaluating $E(\omega_i; \mathcal{M}_i)$ at integer multiples of $\frac{1}{T}$, implicit periodicity assumptions that are harmful for the purpose of forecasting are being made. However, we can exploit the FFT to locate the valley in which the global minimum resides and then apply gradient descent to improve this initial guess and thereby breaking periodicity assumptions. The procedure is depicted graphically in Figure 3 and algorithmically in Algorithm 1. Because gradients with respect to frequencies grow linearly with time ($\sin(\omega t)' = \underline{t} \cos(\omega t)$), the learning rate with respect to the frequency vector $\vec{\omega}$ need to be scaled by T in order to ensure stability of the gradient descent procedure.

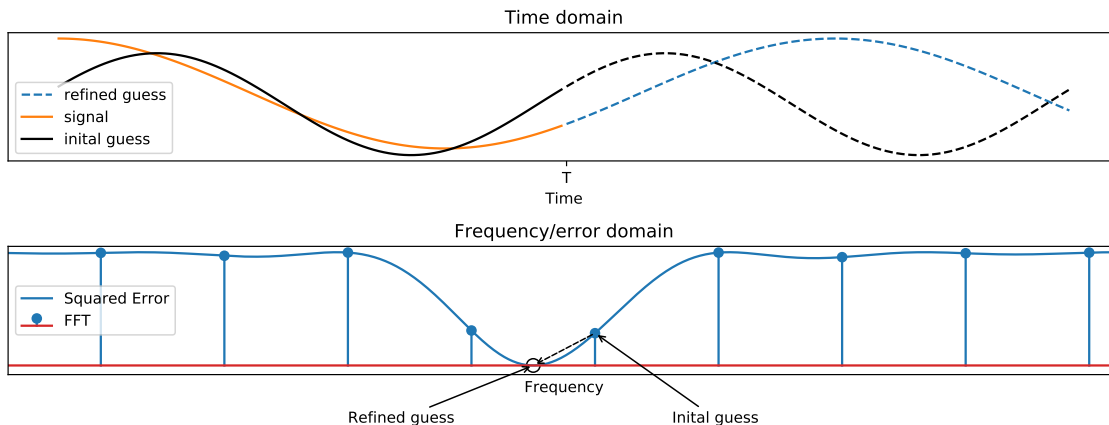


Figure 3: Graphical depiction on how the FFT and gradient descent are combined to extract global minima of the error surface. The FFT is employed to generate an initial guess of the valley in which the global optimum resides, then subsequently this initial guess is refined by gradient descent.

Algorithm 1 Learning a linear oscillator from data

Randomly initialize $\vec{\omega}$ and A

while not converged **do**

for $i \in \{1, \dots, n\}$ **do**

$\omega_i \leftarrow \arg \min_{\omega_i} E(\omega_i; \mathcal{M}_i)$ (using the FFT)

while not converged **do**

$\omega_i \leftarrow \omega_i - \frac{\alpha}{T} \frac{\partial E(\omega_i; \mathcal{M}_i)}{\partial \omega_i}$ ▷ Refine initial guess via GD

end while

$A \leftarrow (\Omega^T \Omega)^{-1} \Omega^T \mathbf{X}$ ▷ Optimize A using pseudo-inverse

end for

end while

The technique described in Algorithm 1 overcomes the weaknesses of the respective optimization techniques. Note that, as described earlier, while gradient descent from a random initial starting point will overemphasize temporally local losses close to T , the FFT will do the opposite. In order to fulfill the periodicity assumption, temporally local losses at the edges, i.e. close to 0 and T , will usually be underemphasized. While gradient descent can ‘get stuck’ in local minima and generally struggles with the fact that the sensitivity with respect to the precision of the model parameters grows linearly with time, the FFT relies on periodicity assumptions in order to be computationally efficient. In a sense, gradient descent and the Fast Fourier Transform are local and global optimization techniques respectively and in this work, we combine global and local techniques to arrive at a global minimizer in a computationally efficient manner.

2.2 Connection to Spectral Leakage and Bayesian Spectral Analysis

Consider the case when x_t and y_t are one- and two-dimensional respectively. In that case, $R(1) = x_t$ which entails that

$$E(\omega) = \sum_{\omega'} \hat{x}(\omega')^2 - \frac{2}{T} |\hat{x}(\omega)|^2$$

Thus, the error of the model is closely linked to the Fourier coefficients. In a sense, this augments the traditional view of the semantics of the Fourier coefficients. In the traditional understanding, the signal is assumed to exhibit a continuous spectrum and the real and imaginary parts of Fourier coefficients denote the amplitude of the corresponding sines and cosines.

Consider the signal $x_t = \cos(\omega t)$. The Fourier transform of x_t is a Dirac delta function centered at $\pm\omega$. However, the discrete Fourier transform of a finite amount of equidistant samples from x_t will exhibit *sinc*-shapes. This effect, usually referred to as spectral leakage (Lyon, 2009), has been studied extensively in the signal processing community. Spectral leakage occurs when computing the FFT with finite amounts of data and can be explained in the following way: A signal of finite length can be understood as the product of an infinitely long signal and the *rect* function stretched to T . Because *rect* and *sinc* functions are Fourier pairs and multiplication in time domain is equivalent to a convolution in frequency domain, for quasi-periodic systems, the FFT will locally exhibit the shape of the *sinc* function. Thus, given finite amounts of data, the FFT will appear to be continuous even though the data generating process, in the limit, exhibits a discontinuous frequency spectrum. This reasoning also explains why combining the FFT and gradient descent is a promising strategy: The error surface is globally non-convex but because *sinc* functions are suitably convex, there is local convexity. The FFT overcomes global non-convexity and gradient descent exploits local convexity.

The amount of data provided influences the width and height of the *sinc* function. As T increases, the height increases and width decreases. This entails that the more data is provided, the higher the precision with which the frequencies can be extracted from data. This view coincides and is to some degree analogous to the work by E.T. Jaynes (Jaynes, 1987) and Bayesian Spectral Analysis (Bretthorst, 2013). In Bayesian Spectral Analysis, a probabilistic model of the signals spectrum is assumed and posterior inference is performed in order to obtain the most likely model parameters given the data. Interestingly, if one assumes the signal to be a simple harmonic oscillation corrupted by Gaussian noise with unknown frequency ω , the DFT provides sufficient statistics of the probability that a specific frequency is present in the data. Assume $\mathbf{X}_{1:T}$ is zero-mean unit-variance and sampled at T equidistant points in time and $\hat{x}(\omega)$ its Fourier transform, then it can be shown that:

$$p(\omega|\mathbf{X}_{1:T}, \sigma) \propto \exp \left[\frac{|\hat{x}(\omega)|^2}{\sigma^2} \right] \tag{7}$$

$$p(\omega|\mathbf{X}_{1:T}) \propto \left[1 - \frac{2|\hat{x}(\omega)|^2}{T} \right]^{1-\frac{T}{2}} \tag{8}$$

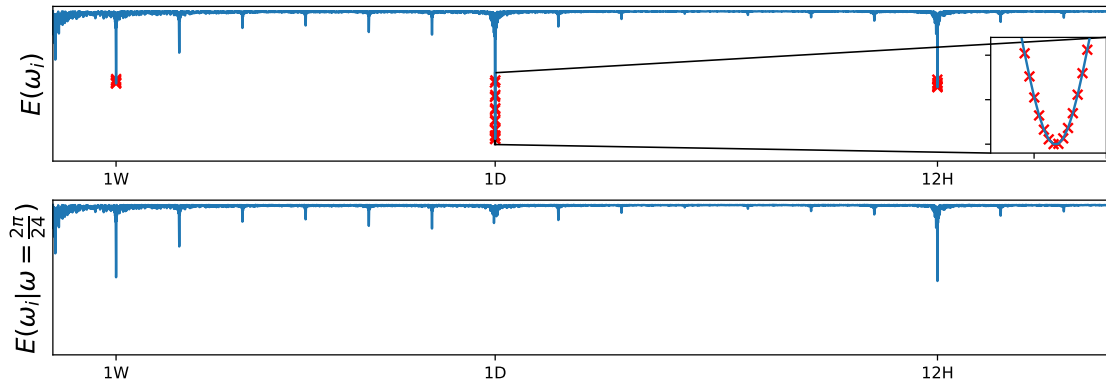


Figure 4: Top: The error surface when no previous frequencies have been selected. Red ‘x’ mark the 50 frequencies with the lowest error. Bottom: The same error surface but when daily patterns have been explained away.

with, if known, σ denoting the standard deviation of the Gaussian noise corrupting the signal. Because the squared error is closely linked to the probability of the Gaussian distribution, this result may come as no surprise to some.

Furthermore, note that the application of coordinate descent in the way described in Algorithm 1 has the property of ‘explaining away’. Because of spectral leakage, even if the true latent process exhibits a frequency spectrum that is a superposition of Dirac delta functions, the Fourier transform will exhibit *sinc*-like shapes. Thus, a naïve approach that computes the error surface and extracts the k frequencies associated with the smallest error might select frequencies that can be explained away by a single frequency. However, by sequentially selecting frequencies, entire *sinc*-shapes are deleted. This is shown graphically in Figure 4.

2.3 Related Work

The class of algorithms that bears the highest resemblance to the introduced algorithm is Dynamic Mode Decomposition (Schmid, 2010) (DMD). DMD was first introduced to extract the dynamic behavior in fluid flows but has since been applied in various fields, such as epidemiology (Proctor and Eckhoff, 2015), acoustics (Song et al., 2013), neuroscience (Brunton et al., 2016a), combustion modeling (Moeck et al., 2013) and video processing (Erichson et al., 2019a). A strong connection between DMD and Koopman theory has been established in the past. Specifically, DMD approximates the Koopman operator when the observables are restricted to direct measurements of the state variable (Rowley et al., 2009; Mezic, 2013; Tu et al., 2014; Brunton et al., 2016b; Kutz et al., 2016).

The goal of DMD is to discover a low-dimensional linear system that captures most of the dynamic behavior of a potentially high dimensional phenomenon. Given temporal snapshots x_t , DMD first applies a dimensionality reduction technique and subsequently learns parameters of a linear system. Let \mathbf{Z}_1 and \mathbf{Z}_2 denote the collection of low-dimensional temporal snapshots starting at 1 to $T - 1$ and 2 to T respectively. The parameters of the

linear system are usually obtained by means of pseudo-inverse, i.e. by solving $\mathbf{Z}_2 = \mathbf{A}\mathbf{Z}_1$. In a sense, the algorithm introduced earlier can be understood as a scalable and stable way to perform the second step of DMD, i.e. to learn parameters of the low-dimensional linear system, however by the means of FFT instead of pseudo-inverse. This is also reflected in the parameterization of the linear system. While DMD extracts a matrix A , our algorithm extracts a vector of frequencies $\vec{\omega}$.

One of the main challenges for DMD-like algorithms is the fact that, under the assumption of merely measurement noise, the pseudo-inverse estimator of \mathbf{A} is biased (Hemati et al., 2017; Dawson et al., 2016). Note that this bias can be explained by the fact that the pseudo-inverse estimator performs one-step-ahead predictions. This challenge has been acknowledged in the research community and approaches have been proposed to mitigate this problem, most notably with the introduction of optimized DMD (Chen et al., 2012) and de-biasing techniques (Hemati et al., 2017). However, solving the optimized DMD objective proves difficult (Askham and Kutz, 2018). Note that the optimization objective introduced in equation (3) is equivalent to the optimized DMD objective with the additional constraint that eigenvalues λ_i come in conjugate pairs and are strictly imaginary which ultimately allows to obtain a globally optimal and unbiased estimator in the direction of ω_i . The fact that this frequency estimator is unbiased under the assumption of merely measurement noise is easy to understand and in the following we will give the intuition why this is true but the reader is referred to pre-existing work in Bayesian Spectral Analysis (Bretthorst, 2013) for a more thorough treatment of the issue. For a signal corrupted by measurement noise, the following holds: $x_e(t) = x(t) + \epsilon(t)$. As long as $Cov(\epsilon(t_1), \epsilon(t_2)) = 0$ and $\mathbb{E}[\epsilon(t)] = 0$, the Fourier transform of $\epsilon(t)$ is $\epsilon(t)$ which entails, because the Fourier transform is linear, that adding noise in time domain is equivalent to adding noise in frequency domain. Thus, measurement noise does not add bias but uncertainty.

Note that previously a strong connection between the Fourier Transform and DMD (Chen et al., 2012) has been shown. In (Chen et al., 2012), the authors advocate not to subtract the mean of the data to avoid reducing DMD to the DFT to ultimately avoid the pitfalls of the DFT, i.e. making harmful and unnecessary periodicity and stationarity assumptions. In this work, enforcing stationarity is a design goal and, as pointed out earlier, periodicity assumptions are broken by the subsequent application of gradient descent in order to obtain frequencies that are not periodic in T . Note that the assumption of stationarity is an inductive bias and from data alone, it is usually impossible to determine whether a slow increase in the signal is caused by an upward trend or a slow frequency. Furthermore, the introduced algorithm allows to model n -dimensional signals composed of m frequencies with $m > n$. Note that the original DMD formulation does not allow this because the dimensionality of the signal is directly tied to the number of frequencies that can potentially be extracted. However, approaches to mitigate this problem have been introduced based on mostly delay embeddings (Tu et al., 2014; Brunton et al., 2016a, 2017; Kamb et al., 2018; Pan and Duraisamy, 2019b). Our algorithm naturally allows one to model low-dimensional signals without these additional steps.

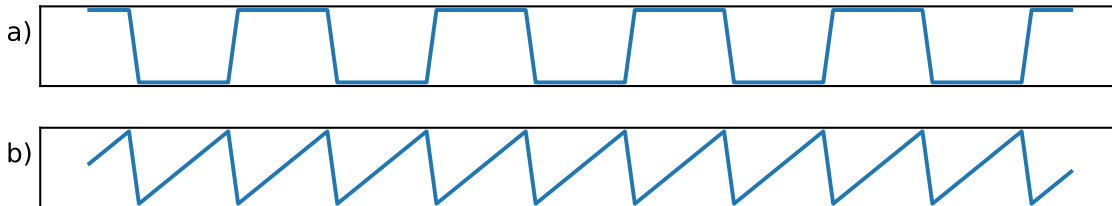


Figure 5: Two signals that both exhibit linearly decaying but infinite frequency spectra.

3. Koopman and Forecasting

Consider the two signals depicted in Figure 5. It is easy to show analytically that both signals exhibit linearly decaying but infinite frequency spectra. However, one could argue that both signals consist of only a single oscillatory pattern that is however not easily expressed by sines and cosines. This raises the question of whether expanding the signal in the basis of sines and cosines is the correct choice and what a better choice of oscillatory basis functions is. Note that optimal basis functions for the two signals are most likely different which implies that the choice of basis functions is dependent on the signal.

In the following, we will derive an algorithm based on Koopman theory that tries to answer these questions. Specifically, we will derive an algorithm that performs a frequency decomposition in a nonlinear basis. The nonlinear oscillatory basis as well as a least-squares fit is learned jointly from data. The algorithm will retain the notions of frequency and periodicity and will therefore provide some feedback to practitioners of the duration of the extracted patterns. Because the resulting algorithm is closely linked to the Fourier transform, we will show how the Fast Fourier Transform can be leveraged to obtain model parameters in a computationally efficient and scalable manner similar to approach introduced earlier, in spite of non-convexities and nonlinearities.

Koopman theory postulates that any nonlinear dynamical system can be lifted into the space of observable functions ψ , in which the system evolves linearly. Specifically, it can be shown that there always exists a \mathcal{K} such that:

$$\psi(\mathbf{x}_t) = \mathbf{k}_t = \mathcal{K}\mathbf{k}_{t-1} = \mathcal{K}\psi(\mathbf{x}_{t-1}) \quad (9)$$

with \mathcal{K} commonly referred to as the Koopman operator. Note that the Koopman operator and therefore \mathbf{k}_t might be infinite dimensional and that ψ and \mathcal{K} are usually unknown.

Even if \mathcal{K} and ψ were known, forecasting \mathbf{x}_t would still not be trivial because even though one could evolve \mathbf{k}_t into the future, transforming knowledge of future values of \mathbf{k}_t into knowledge of \mathbf{x}_t is complicated. This is why in the past, ψ was often assumed to be invertible. Additionally, it was assumed that ψ can be approximated by a function (as opposed to a functional). In the following, we will show that these assumptions severely limit the applicability of Koopman theory. Specifically, we will show that these assumption imply that if \mathbf{x}_t takes the same value for two points in time t_1 and t_2 , future values \mathbf{x}_{t_1+1}

and \mathbf{x}_{t_2+1} must be equal. Or mathematically speaking, $\mathbf{x}_{t_1} = \mathbf{x}_{t_2} \iff \mathbf{x}_{t_1+1} = \mathbf{x}_{t_2+1}$.

$$\mathbf{x}_{t_1} = \mathbf{x}_{t_2} \iff f(\mathbf{x}_{t_1}) = f(\mathbf{x}_{t_2}) \quad (f \text{ is function and invertible})$$

Evolving $f(\mathbf{x}_t)$ into the future using the Koopman operator yields:

$$f(\mathbf{x}_{t_1+1}) = \mathcal{K}f(\mathbf{x}_{t_1}) = \mathcal{K}f(\mathbf{x}_{t_2}) = f(\mathbf{x}_{t_2+1}) \iff \mathbf{x}_{t_1+1} = \mathbf{x}_{t_2+1}$$

By making the assumption of invertibility and that ψ can be approximated by a function, even very simple systems such as $x_t = \cos(t)$ cannot be lifted into a space where it evolves linearly. Let $t_1 = \frac{\pi}{2}$ and $t_2 = \frac{3\pi}{2}$, which implies that $x_{t_1} = 0 = x_{t_2}$. However, $x_{t_1+1} < 0$ and $x_{t_2+1} > 0$. Note that lifting x_t into a space where it evolves linearly is trivial using a functional (as opposed to function): Let $\psi(f) = \begin{bmatrix} f \\ f' \end{bmatrix}$, then $\psi(x_t) = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$ which evolves linearly.

In order to overcome these limitations but still allow for forecasting of x_t , we make slightly different assumptions. In order to allow for forecasting, we assume ψ to be invertible but we only assume that ψ^{-1} can be approximated by a function f . Therefore:

$$\mathbf{x}_t = f(\mathbf{k}_t) \text{ with } \mathbf{k}_t = \mathcal{K}\mathbf{k}_{t-1} \tag{10}$$

Additionally, similar to the reasoning laid out earlier, we assume \mathcal{K} to have strictly imaginary eigenvalues, which in turn implies that, if k_t is being projected onto the eigenvectors of \mathcal{K} , the following holds:

$$\mathbf{k}_t = \begin{bmatrix} \cos(\vec{\omega}t) \\ \sin(\vec{\omega}t) \end{bmatrix} := \Omega(\vec{\omega}t) \tag{11}$$

3.1 Nonlinear Frequency Decomposition

By making the above mentioned assumptions, we arrive at an interpretation of Koopman theory that bears resemblance to the Fourier transform. Specifically, while the Fourier transform expands a signal x_t into a possibly infinite sum of sines and cosines, the transformation that results from our interpretation of Koopman theory explains the signal as a nonlinear function of sines and cosines. Thus, it explains the signal in a nonlinear oscillatory basis but because we assume the nonlinear function f to be time-invariant, notions of frequency and periodicity persist, i.e., if $\Omega(\vec{\omega}t)$ is periodic with period P , then $f(\Omega(\vec{\omega}t))$ will also be periodic with period P .

$$\mathbf{x}_t = \mathbf{A}\Omega(\vec{\omega}t) \tag{Fourier}$$

$$\mathbf{x}_t = f(\Omega(\vec{\omega}t)) \tag{Koopman}$$

Let f be parameterized by Θ . The minimization problem we wish to solve is therefore:

$$\begin{aligned} \text{minimize } E(\vec{\omega}, \Theta) &= \sum_{t=1}^T (\mathbf{x}_t - f_{\Theta}(\Omega(\vec{\omega}t)))^2 && \text{(global loss)} && (12) \\ &= \sum_{t=1}^T L(\vec{\omega}, \Theta, t) && \text{(sum of temporally local losses)} \end{aligned}$$

In the following, we will derive computationally efficient strategies to solve this optimization problem. Again, similar to the algorithm introduced earlier, coordinate descent will be performed on (12). We will show how a global minimizer in the direction of ω_i can be obtained in a computationally efficient manner using the FFT. Note that optimizing E in the direction of Θ can be achieved efficiently using Dynamic Programming, specifically using the Backpropagation algorithm, whose implementation is trivial in programming packages that allow for automatic differentiation. However, the question of how to optimize $\vec{\omega}$ remains.

Let $\mathcal{M}_i = \{\Theta, \omega_1, \dots, \omega_{i-1}, \omega_{i+1}, \dots, \omega_m\}$, i.e. the set of all model parameters except ω_i . Let $L(\omega_i; \mathcal{M}_i, t)$ be the loss as a function of solely ω_i , i.e. while keeping all other model parameters fixed. Recall that $L(\omega_i; \mathcal{M}_i, t)$ is periodic in $\frac{2\pi}{t}$. The periodicity property follows directly from characteristics of the sine and cosine function:

$$\sin((\omega_i + \frac{2\pi}{t})t) = \sin(\omega_i t + 2\pi) = \sin(\omega_i t)$$

Thus, in a span of 2π , a temporally local loss function at t repeats itself t -times as shown graphically in Figure 6. It is therefore sufficient to compute the loss in the first period and then extrapolate this knowledge by simple repetition. Equipped with this knowledge an efficient strategy can be devised: Each temporally local loss function is sampled at N equidistant points between 0 and $\frac{2\pi}{t}$:

$$\begin{aligned} \text{for } n = 1, \dots, N \quad s_n &= \frac{n}{N} \frac{2\pi}{t} \\ S_{i,t}[n] &= L(s_n; \mathcal{M}_i, t) \end{aligned} \tag{13}$$

Note that this step can be carried out in parallel and is therefore surprisingly fast and that for every i and t , $S_{i,t}[n]$ is periodic with period N . In order to obtain the global loss $E(\vec{\omega}, \Theta)$, in a subsequent step, the temporally local loss functions are summed up by repeating the loss at time point t , t times and then aligning the sampling intervals. This seemingly easy step proves challenging because of the different scales of the samples drawn from the local loss functions. However, because summation requires that temporally local loss functions to be repeated and interpolated, this step can be performed efficiently in frequency domain ($\mathcal{O}(T \log T)$) by *zero-stuffing* and *zero-padding* respectively, avoiding the computational burden of a naïve time-domain implementation ($\mathcal{O}(T^2)$).

Repeating the local loss functions can be achieved efficiently by inserting t -many 0's in between values of the FFT of the samples drawn (*zero-stuffing*). Then, in order to match scales, temporally local loss functions are interpolated by appending 0's in frequency domain (*zero-padding*).

Therefore:

$$\text{for } k = 0, \dots, K \quad \mathbf{S}_{i,t}[k] = \sum_{n=1}^N S_{i,t}[n] e^{-j\pi \frac{nk}{K}} \tag{14}$$

For the frequency spectrum of the global loss function then holds:

$$\text{for } l = 0, \dots, KT \quad \mathbf{E}_i(l) = \sum_{t=1}^T \sum_{k=0}^K \mathbf{S}_{i,t}[k] \delta(tk - l) \tag{15}$$

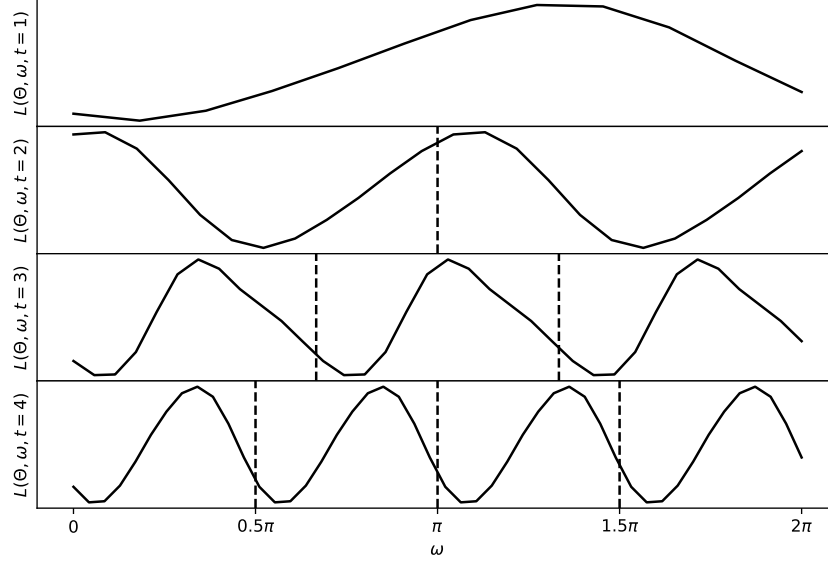


Figure 6: The temporally local loss function at time t oscillates with period $\frac{2\pi}{t}$. In order to reconstruct the global loss, it is sufficient to evaluate each local loss function within the first period.

Algorithm 2 Learning a nonlinear oscillator from data

```

Randomly initialize  $\vec{\omega}$  and  $\Theta$ 
while not converged do
  for  $i \in \{1, \dots, n\}$  do
    Compute  $\mathbf{S}_{i,t}[k]$  ▷ Based on (13) and (14)
     $E \leftarrow [0]^{TK}$ 
    for  $t \in \{1, \dots, T\}$  do
      for  $k \in \{0, \dots, 2N\}$  do
         $E[tk] \leftarrow E[tk] + S_{i,t}[k]$  ▷ Implements (15)
      end for
    end for
     $\omega_i \leftarrow \arg \min_{\omega} \mathcal{F}[E](\omega)$ 
    while not converged do
       $\omega_i \leftarrow \omega_i - \frac{\alpha}{T} \frac{\partial E(\omega_i | \vec{\omega})}{\partial \omega_i}$  ▷ Refine initial guess of  $\omega_i$ 
    end while
    for a couple of iterations do
       $\Theta \leftarrow \Theta - \alpha \frac{\partial E(\vec{\omega}, \Theta)}{\partial \Theta}$  ▷ Gradient descent on  $\Theta$ 
    end for
  end for
end while
    
```

Thus, similar to the previously introduced algorithm, we derived a computationally efficient expression for the frequency spectrum of the error function which allows us to obtain the approximate location of the optimal parameter ω_i . This initial guess is then refined using gradient descent. The resulting algorithm is described in Algorithm 2.

3.2 Related Work

The fact that any nonlinear dynamical system can be lifted by the means of observables into a space where its time evolution can be described by linear methods was first introduced by Koopman for Hamiltonian systems (Koopman, 1931) and later generalized to continuous-spectrum systems (Koopman and Neumann, 1932). Even then it was of considerable importance as a building block for advances in ergodic theory (Birkhoff, 1931; Neumann, 1932b; Birkhoff and Koopman, 1932; Neumann, 1932a; Moore, 2015). Interest in Koopman theory was renewed by work by Mezić (2005); Budišić et al. (2012); Mezić (2013). More recent theoretical advances in Koopman theory e.g. include generalization of the Hartman-Grobman theorem to the entire basin of attraction of equilibrium points or periodic orbits (Lan and Mezić, 2013).

Recent applied research has focused on algorithms to estimate the Koopman operator from data. Most of these approaches rely on auto-encoder structures (Otto and Rowley, 2019; Lusch et al., 2018; Yeung et al., 2019; Wehmeyer and Noé, 2018; Takeishi et al., 2017). The optimization objective is usually constituted of terms that encourage linearity in the latent space as well as reconstruction performance and is usually solved by means of gradient descent. These approaches have been extended in various ways. For example, Bayesian Neural Networks as encoders were utilized to extend Koopman theory to the probabilistic setting (Pan and Duraisamy, 2019a) whereas Champion et al. (2019) relaxed the linearity requirement and allowed for sparse dynamics in the latent space. These approaches usually do not consider linearity in Koopman-space a constraint but rather the optimization objective. This usually results in latent spaces that are approximately and only locally linear. On top of that, unstable eigenvalues often cause difficulties. In Erichson et al. (2019b), stable eigenvalues are encouraged through additional regularization by means of Lyapunov stability. In this work, on the other hand, global linearity and stability are constraints of the optimization objective, i.e. all solutions are globally linear and stable by construction without the need for any additional regularization. Furthermore, in this work, we show that the global optimization objective is highly non-convex limiting the practicality of gradient descent and show how solutions can be obtained in frequency domain.

The Fourier and Koopman Forecast algorithms are also tangentially related to phase retrieval (PR) algorithms (Fienup, 1982; Saxton, 2013; Metzler et al., 2018). These algorithms usually assume partial knowledge in time and/or Fourier domain and the goal is to reconstruct missing phase information. PR algorithms find application in wavefront sensing, beam shaping, optical encryption, optical refraction tomography, x-ray coherent diffractive imaging, etc. Whereas PR assumes partial knowledge in Fourier and/or time domain and the goal is inference of unknown phase quantities, the Koopman/Fourier algorithms assume full knowledge in time domain and the objective is prediction. Similarly to the Koopman/Fourier algorithm, PR methods oftentimes solve their objective function in an alternating fashion, i.e. by iteratively updating unknown quantities in time and then

Fourier domain. However, unlike the methods described in this paper, PR algorithm do not seem to be confronted with the problem of obtaining optimal frequencies for time-series prediction from data.

4. Algorithmic properties

In this section, basic properties of the algorithms are derived.

4.1 Universal approximation capabilities

4.1.1 FINITE DATA

We will show that even the linear Fourier-based algorithm has universal approximation capabilities for finite data sets. More formally, we show by construction that there always exists finite model parameters that can explain any finite data set. Note that this property is not very useful in practice because the solution does not allow to generalize as it implicitly assumes that the signal is periodic in T . It is included here because it is analogous to Cover's theorem. Similarly to Cover's theorem, colloquially speaking, we will show that even the linear Fourier-based algorithm can always overfit on the training set. Analogous to Cover's theorem, the proof requires projecting the data into an N -dimensional space. In summary, Cover's theorem and the Fourier-based algorithm both have trivial solutions in N dimensions that however do not generalize.

Lemma 1 (Finite data: Universal approximation capabilities) *Assume T equidistant samples x_1, \dots, x_T . There always exist finite-dimensional $\vec{\omega}$ and A such that, $A\Omega(\omega t) = x_t$. Proof by construction: Choose $M = T/2$ and for $i = 1, \dots, M$ let $\omega_i = \frac{2\pi i}{T}$. Let $O \in \mathbb{R}^{T \times T}$ the matrix whose i th and $(i + T/2)$ th rows constitute $\cos(\omega_i t)$ and $\sin(\omega_i t)$ evaluated at $t = 1, \dots, T$, respectively. What is left to show is that O forms a basis of \mathbb{R}^T which follows directly from*

$$\begin{aligned}
 i \neq j : \sum_{t=1}^T \sin\left(\frac{2\pi i t}{T}\right) \sin\left(\frac{2\pi j t}{T}\right) &= 0 \\
 i \neq j : \sum_{t=1}^T \cos\left(\frac{2\pi i t}{T}\right) \cos\left(\frac{2\pi j t}{T}\right) &= 0 \\
 i \neq j : \sum_{t=1}^T \sin\left(\frac{2\pi i t}{T}\right) \cos\left(\frac{2\pi j t}{T}\right) &= 0
 \end{aligned}$$

4.1.2 INFINITE DATA

However, when the signal is assumed to be infinite, we will show that the Koopman-based algorithm is more expressive compared to its Fourier counterpart. Specifically, we will show that there are signals for which the Fourier algorithm would require infinite model size, while the Koopman algorithm allows for a finite parameterization. The signal for which this is shown is a simple square wave.

Lemma 2 (Infinite data) *Let $\%$ denote the modulo operator and consider the time-discrete system that evolves according to the following equations:*

$$x_t = \begin{cases} 1, & \text{for } t\%4 = 0 \\ 1, & \text{for } t\%4 = 1 \\ 0, & \text{otherwise} \end{cases}$$

It is well known that the discrete-time Fourier transform of x_t is linearly decaying but infinite. The system can however be modeled with a simple nonlinear function f and a single frequency:

$$f(x) = \begin{cases} 1, & \text{for } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$x_t = f(\cos(\frac{\pi}{2}t))$$

4.2 Error bounds

In the following, practical error bounds for the Fourier algorithm will be derived as a function of prediction horizon, noise and amount of data provided.

Lemma 3 (Fourier: Linear-in-time error bounds) *Let $\bar{x}_t(\vec{\omega}) = A\Omega(\vec{\omega}t) = \sum_{i=1}^{N/2} A_i \cos(\omega_i t) + \sum_{i=N/2+1}^N A_i \sin(\omega_i t) \in \mathbb{R}$. Given number of data points T , number of frequencies $N/2$ and noise variance σ^2 , for the prediction error at horizon t then holds:*

$$|\hat{x}_t(\vec{\omega}) - \hat{x}_t(\vec{\omega}^*)| \in \mathcal{O}\left(\frac{t}{\sqrt{T^3}} N \sigma^2\right)$$

Proof.

$$\begin{aligned} |\bar{x}_t(\vec{\omega}) - \bar{x}_t(\vec{\omega}^*)| &= \left| \sum_{i=1}^{N/2} A_i \cos(\omega_i t) + \sum_{i=N/2+1}^N A_i \sin(\omega_i t) - \sum_{i=1}^{N/2} A_i \cos(\omega_i^* t) + \sum_{i=N/2+1}^N A_i \sin(\omega_i^* t) \right| \\ &\leq \sum_{i=1}^{N/2} |A_i| |\cos(\omega_i t) - \cos(\omega_i^* t)| + \sum_{i=N/2+1}^N |A_i| |\sin(\omega_i t) - \sin(\omega_i^* t)| \end{aligned}$$

Note that $\sin(\omega t)$ and $\cos(\omega t)$ are Lipschitz continuous with constant t , therefore:

$$\begin{aligned} |\sin(\omega_i t) - \sin(\omega_i^* t)| &\leq t |\omega_i - \omega_i^*| \\ |\cos(\omega_i t) - \cos(\omega_i^* t)| &\leq t |\omega_i - \omega_i^*| \end{aligned}$$

Because of that:

$$|\bar{x}_t(\vec{\omega}) - \bar{x}_t(\vec{\omega}^*)| \leq t \sum_{i=1}^N |A_i| |\omega_i - \omega_i^*|$$

From *Bayesian Spectral Analysis*, we know the standard deviation of the uncertainty with which the frequencies can be resolved. As was derived in Jaynes (1987):

$$\omega = \omega^* \pm \frac{\sigma^2}{|A|} \sqrt{\frac{48}{T^3}}$$

Assume ω_i was resolved within ρ standard deviations of ω_i^* , then:

$$\begin{aligned} \omega_i^* - \rho \frac{\sigma^2}{|A_i|} \sqrt{\frac{48}{T^3}} &\leq \omega_i \leq \omega_i^* + \rho \frac{\sigma^2}{|A_i|} \sqrt{\frac{48}{T^3}} \\ |\omega_i - \omega_i^*| &\leq \rho \frac{\sigma^2}{|A_i|} \sqrt{\frac{48}{T^3}} \end{aligned}$$

Combining results yields:

$$\begin{aligned} |\hat{x}_t(\vec{\omega}) - \hat{x}_t(\vec{\omega}^*)| &\leq t \sum_{i=1}^N |A_i| |\omega_i - \omega_i^*| \leq \rho \frac{t}{\sqrt{T^3}} \sum_{i=1}^N \sqrt{48} \sigma^2 \\ |\hat{x}_t(\vec{\omega}) - \hat{x}_t(\vec{\omega}^*)| &\in \mathcal{O} \left(\frac{t}{\sqrt{T^3}} N \sigma^2 \right) \end{aligned}$$

Thus, the Fourier algorithm scales super-linearly with the amount of data provided and degrades linearly with noise variance and prediction horizon t .

4.3 Convergence and stability: Unknown phase problem

As described earlier, because the Fourier and Koopman methods rely on coordinate descent, they inherit the convergence properties of the optimization algorithm. Coordinate ascent has known requirements for convergence that are however hard to verify in reality (Beck and Tetruashvili, 2013; Spall, 2012). Note that both algorithms obtain optimal frequencies ω_i in sequence and optimize either A (Fourier) or Θ (Koopman) jointly. The question arises to which degree guesses of frequencies and either A or Θ affect the stability and convergence of the respective algorithms.

4.3.1 FOURIER

In the case of the Fourier algorithm, a ‘bad guess’ of A does not affect the optimality of the subsequent guess of ω_i . This follows directly from the fact that when deriving the symmetry relationship between the error surface and the Fourier coefficients, an analytic solution for optimal A , i.e. a solution to $\frac{\partial E}{\partial A} = 0$, was obtained which led to the elimination of A from the objective function. That is why, the initial guess of A does not affect subsequent guesses of ω_i . The question to which degree guesses of $\{\omega_j\}_{i \neq j}$ affect the guess ω_i remains. The answer to the question is linked to the number of data points provided. The more data is provided, the smaller the ‘radius of influence’ of the respective frequencies. This is due to the fact that larger amounts of data lead to slimmer and peakier *sinc*-functions in Fourier domain, and the fact that two frequencies are independent if their *sinc*-function do not overlap. This also implies that in the limit of $T \rightarrow \infty$, guesses of $\{\omega_j\}_{i \neq j}$ do not influence ω_i because the

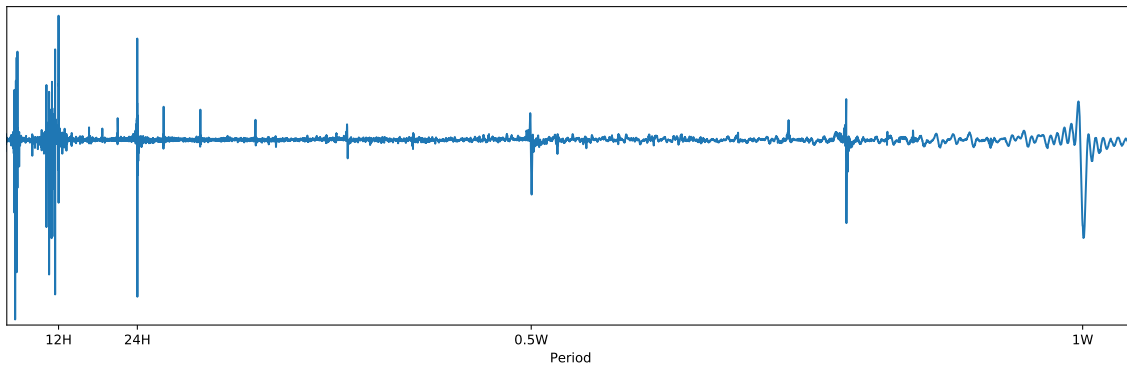


Figure 7: The error surface $E(\omega, \Theta)$ of a signal exhibiting daily, weekly and seasonal patterns. In this case f_{Θ} is an *untrained* Neural Network.

corresponding *sinc*-functions have become Dirac deltas. These properties make the Fourier algorithm stable and easy to use in practice. After the number of frequencies N are selected by either domain knowledge, cross-validation or any hyper-parameters optimization scheme, the algorithm almost always converges to the same solution given appropriate scaling of the data and a small enough learning rate.

4.3.2 KOOPMAN

Unfortunately, this story is different for the Koopman algorithm. Because of the non-linearity f , the ‘radius of influence’ of frequencies is much harder to reason about. That is why we conjecture that whether or not guesses of $\{\omega_j\}_{i \neq j}$ affect guesses of ω_i is data dependent. From experience, frequency guesses seem to not influence each other much. However, because analytical solution to optimal Θ , i.e. $\frac{\partial E}{\partial \Theta} = 0$, are unavailable for most non-linear functions f_{Θ} and numerically computing the optimal parameters Θ for every value that ω_i can take is computationally extremely expensive, ‘bad initial guesses’ of Θ influence subsequent guesses of ω_i . However, the way in which this pathology manifests itself is fortunately structured. Figure 7 shows the error surface $E(\omega)$ with respect to a signal that exhibits daily, weekly and seasonal patterns given a neural network with random weights, i.e. Θ was initialized randomly. It is easy to see that correct frequencies can be identified. However, there is a price to pay for the non-linearity of f , namely that the correct frequency merely cause the error function to deviate from its plateau. Thus, even an upward spike in the error function can indicate a correct frequency. We call this issue the *unknown phase problem*. Training the Neural Network for some iterations or rerunning the experiments from different initial conditions often alleviates this problem. A heuristic that significantly improves numerical stability is to flip spikes into a common direction followed by the application of a Gaussian filter G , i.e. $E^{\#} = G(|E - \text{median}(E)|)$. These properties make the Koopman method more difficult to use in practice but, as we will show in the next section, successful application of the Koopman algorithm can significantly improve performance on noisy and high dimensional data.

5. Experiments

The code for the Fourier- and Koopman forecasting algorithms is available at https://github.com/helange23/from_fourier_to_koopman.

5.1 Synthetic experiments

In the following, the efficacy of the algorithms is probed on synthetic tasks and compared to the Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). LSTM is a trainable nonlinear filter and one of the de facto standard tools to model time series. LSTMs were initially developed to mitigate the problem of exploding or vanishing gradients by choosing a specific neural architecture that ensures constant error flow. Note that exploding gradients is also a problem we faced but instead mitigated by employing the FFT.

5.1.1 FORECASTING NONLINEAR OSCILLATORS UNDER MEASUREMENT NOISE

In order to probe the performance of the resulting algorithms in long-term forecasting of nonlinear oscillators under measurement noise, a synthetic time series is created. Specifically, we choose a strongly nonlinear signal resembling data at an hourly rate with daily patterns that is corrupted by Gaussian noise, i.e.:

$$x_t = \sin\left(\frac{2\pi}{24}t\right)^{17} + \epsilon_t$$

In a first experiment, we vary the amount of training data provided to the models and keep the noise variance fixed at 0.2. Each model is provided with 2000, 4000, 6000 and 8000 samples of x_t and the performance at predicting the next 10^4 snapshots is evaluated. For the LSTMs, a 5-layer 20-unit network comprising 15301 parameters was chosen. For long-term predictions, previously predicted snapshots are recursively fed back into the network. The LSTM is compared to the Koopman algorithm with a 3-layer decoder with roughly the same number of parameters (16384) and a single frequency. The Fourier algorithm was instantiated with 8 frequencies and therefore 24 parameters.

The left panel of Figure 8 shows the Relative Cumulative Error as a function of the prediction horizon and the amount of provided training data. We define the Relative Cumulative Error as:

$$RCE(h) = \frac{\sum_{t=1}^{T+h} (x_t - \hat{x}_t)^2}{\sum_{t=1}^{T+h} x_t^2} \quad (16)$$

For the sake of readability, the right panel shows the performance of only the Fourier-based and Koopman-based algorithms on a subset of the benchmarks when the prediction horizon equals 10^4 . Note that this experiment confirms the theoretical findings provided earlier, namely that the algorithms scale super-linearly with the amount of data. Furthermore, note that the Koopman algorithm outperforms the Fourier algorithm. Figure 9 gives the intuition why this is true.

Figure 9 shows the time-domain long-term forecasts of the respective algorithms. Note that the LSTM seems to have correctly extracted the nonlinear shape of the waveform but

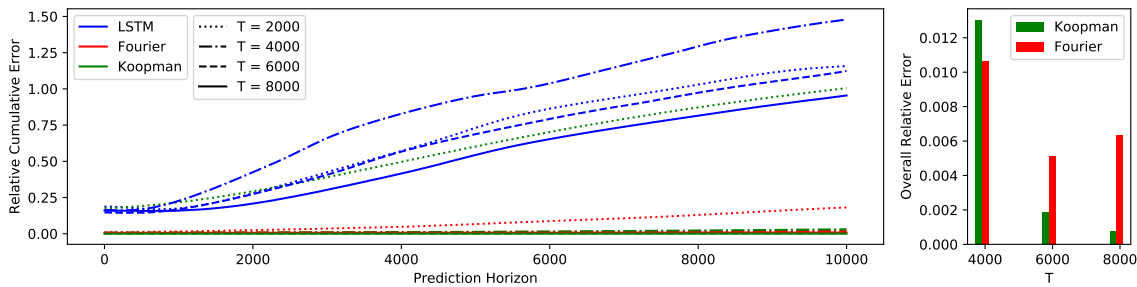


Figure 8: Left panel shows the Relative Cumulative Error as a function of the amount of training data provided. The right panel displays the Relative Cumulative Error when the prediction horizon equals 10^4 but, for better visibility, leaves out the results for the LSTM and $T=2000$.

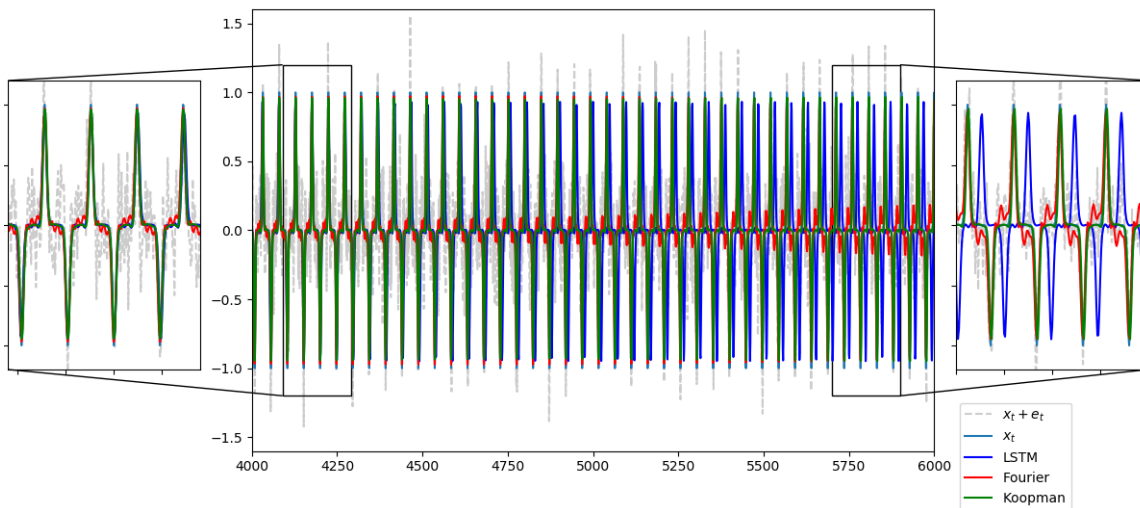


Figure 9: Long-term forecasting results for $T = 4000$. Note that LSTMs correctly extract the shape of the nonlinearity but fail to extract the correct frequency. Furthermore note that for the Fourier algorithm small Fourier modes seem to be drowned by noise.

failed at extracting the correct frequency. The impression that LSTMs seem to be biased frequency estimators will be reinforced in later experiments. Furthermore note that the Koopman algorithm significantly outperforms the Fourier algorithm. While the Fourier algorithm correctly extracted large Fourier modes from the data, it struggles extracting smaller ones as they appear to be wiggling slowly out of phase. This behavior can be explained by the theoretical findings earlier: smaller Fourier modes have a smaller signal-

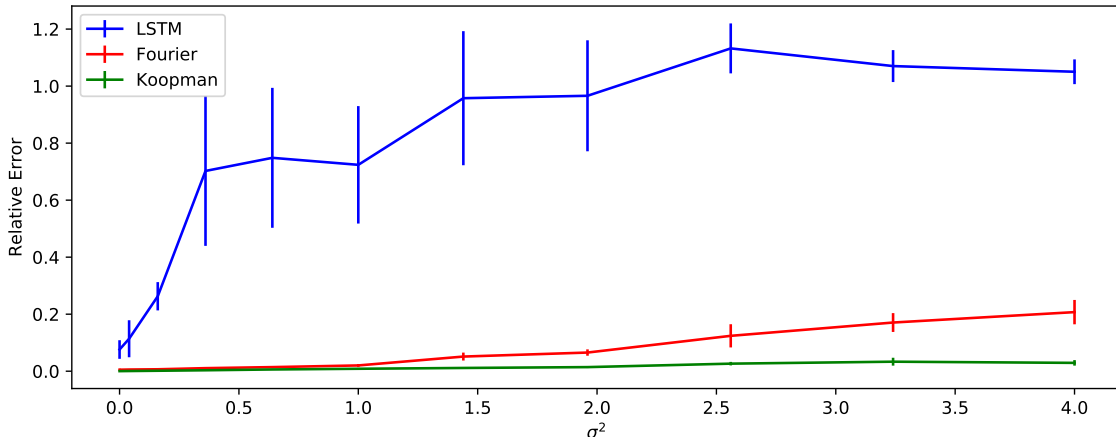


Figure 10: Robustness of the respective algorithms to noise. The amount of training data is kept fixed at $T = 4000$ while noise variance is varied. The Koopman-based algorithm significantly outperforms its Fourier counterpart as well as LSTMs.

to-noise-ratio compared to bigger ones and are therefore harder to detect correctly.

In a subsequent experiment, we probe the robustness of the respective algorithms to noise by keeping the amount of training data fixed ($T = 4000$) and varying the variance of the Gaussian noise corrupting the signal. Figure 10 shows the results. First, note that the theoretical results are again confirmed, i.e. the performance of the algorithms seems to degrade linearly with noise variance. Furthermore note that, again, the Koopman-based algorithm outperforms its Fourier counterpart. This can, again, be explained by the fact that the Fourier-based algorithm requires more Fourier modes to model the signal appropriately. However, these additional Fourier modes often have small amplitudes and therefore small signal-to-noise ratios and get therefore drowned out by noise quickly. The Koopman-based algorithm avoids this pitfall as it requires fewer frequencies because it can explain away harmonics using the non-linearity.

5.1.2 TRAVELING WAVES

For many data-driven modeling approaches to high dimensional systems such as e.g. fluid flows, the dimensionality of the measurements are often first reduced by applying dimensionality reduction techniques. The Koopman-based algorithm allows for jointly learning the inverse of the transformation that reduces the dimensionality and therefore extends the applicability of the techniques to high dimensional systems with which traditional dimensionality reduction techniques such as Principal Component Analysis (PCA) struggle. A standard example for systems with which PCA struggles are traveling waves. Because PCA exploits redundancies in the co-variance structure of the measurements and measurements within one cycle of a traveling wave are approximately uncorrelated, achieving good reconstruction with PCA requires many modes which in turn reduces robustness to noise as highlighted in previous experiments. In the following, we will show how the Koopman-based

algorithm can overcome these difficulties. Specifically, by showing how a 2D traveling wave can be modeled with a transpose-convolutional decoder.

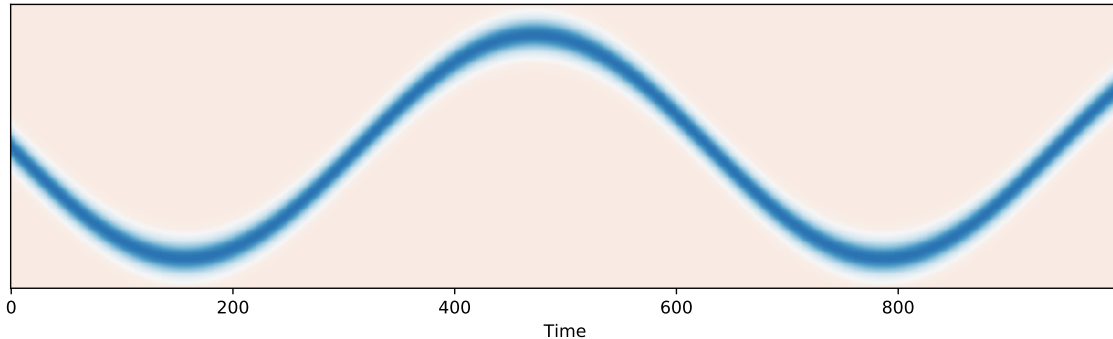


Figure 11: A traveling wave for which reducing the dimension with PCA proves difficult.

Figure 11 shows the first 1000 snapshots of a traveling wave with spatial dimensionality of 256. Let each spatial dimension be indexed by $1 \leq u \leq 256$. For the spatio-temporal evolution of the wave holds:

$$x(u, t) = \mathcal{N}(u | \mu = (\sin(0.01t) + 1) * 100 + 28, \sigma^2 = 10)$$

The performance of the Fourier-based algorithm in conjunction with PCA is compared to an instance of the Koopman-based algorithm that employs a transpose-convolutional Neural Network as a decoder. For the Fourier-based algorithm in conjunction with PCA, a parameter sweep over the combination of retained PCA modes and the number of frequencies is performed. In total, 100 combinations are tested and the results are shown in Figure 12. The best performing model incurs an error of approximately 6% with 55 retained PCA modes and frequencies, respectively. On the other hand, the Koopman-based algorithm in conjunction with a transpose-convolutional decoder achieves an error smaller than 1% with just a single frequency.

5.2 Natural data experiments

5.2.1 ENERGY DEMAND: RE-EUROPE DATA SET

In the following, we will show that physical systems can be modeled effectively as nonlinear oscillators. The example at hand is taken from the realm of power systems. Specifically, the goal of the experiment is to predict the total energy consumption at the distribution level of a node in the German electrical grid. Predicting future consumption is paramount for numerous applications such as power quality, grid stability and demand response (Chan et al., 2012). For this, a one-dimensional time series of past demand is extracted from the RE-Europe data set (Jensen and Pinson, 2017). Energy demand usually exhibits multiple scales, i.e. because energy consumption is usually higher during weekdays compared to weekends, energy consumption often exhibits weekly alongside daily and seasonal patterns. On top of that, because energy consumption is dependent on hard-to-predict outside factors such as weather, predicting energy demand into the future requires high robustness to

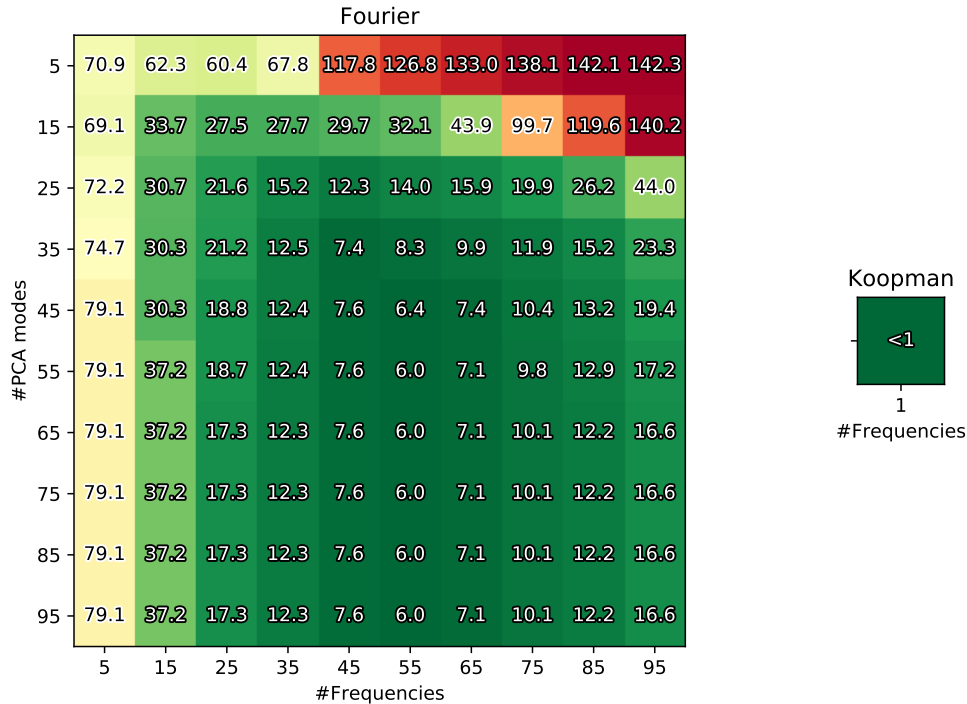


Figure 12: The results of a parameter sweep of the Fourier-based algorithm in conjunction with PCA on modeling a traveling wave in comparison to the Koopman-based algorithm with a deconvolutional decoder.

structured noise. The data set at hand contains 3 years of data at an hourly rate, therefore 26.280 data points. The first 19.000 data points were used as training, the next 1000 for testing, and the last 6.280 for evaluation. The evaluation set was in turn divided into four equal in size consecutive subsets. The performance of the Fourier- as well as Koopman-based algorithm is compared to various temporal models such LSTM, GRU, Box-Jenkins models, Clockwork-RNNs, Echo State Networks, Temporal Convolutional Networks and Fourier Neural Networks.

For LSTMs and GRUs, the standard pytorch implementation is utilized and a parameter sweep over the number of layers as well as the number of hidden units per layer is performed totalling 100 different configurations. In order to be able to learn yearly patterns, the Temporal Convolutional Network was instantiated with 13 layers ($2^{13} \approx 365 * 24$), a kernel size of 3 and a hidden dimensionality of 25 Bai et al. (2018). For the Box-Jenkins models the Hyndman-Khandakar (Hyndman et al., 2007) AutoARIMA algorithm implemented in the *R* forecasting package is employed which uses a combination of unit root tests, minimization of the Akaike Information Criterion (Sakamoto et al., 1986) and Maximum Likelihood Estimation to obtain an ARIMA model. We compare to the predicted mean. Note that the hyperparameter space for Box-Jenkins models is very large and computationally too expensive to sweep. But in addition to the AutoARIMA approach, handcrafted ARIMA models are created. Because of the long-range dependencies in the data, the best practices of detrending and differencing followed by an investigation of (partial) auto-correlation func-

Algorithm	Forecast Horizon				Patterns		
	25%	50%	75%	100%	D	W	Y
Koopman Forecast	0.19	0.21	0.19	0.19	✓	✓	✓
Fourier Forecast	0.31	0.39	0.33	0.3	✓	✓	✓
LSTM	0.37	0.4	0.42	0.45	✓	×	×
GRU	0.53	0.55	0.52	0.5	✓	×	×
Echo State Network	0.67	0.73	0.76	0.73	✓	×	×
AR(1,12,24,168,4380,8760)	0.75	0.95	1.07	1.13	✓	✓	✓
CW-RNN (data clocks)	1.1	1.14	1.14	1.15	(✓)	×	×
CW-RNN	1.05	1.08	1.08	1.09	(✓)	×	×
AutoARIMA	0.83	1.11	1.18	1.26	×	×	×
Temporal Convolutional Nets	0.96	1.69	1.87	2.33	✓	(✓)	×
Fourier Neural Networks	1.1	1.15	1.21	1.21	✓	×	×

Table 1: Performance in long-term forecasting of distribution level energy consumption as measured by the Relative Cumulative Error for various algorithms. Note that long-term predictions are obtained by recursively feeding predictions back into algorithms where applicable. Furthermore, column Patterns indicates whether algorithms at hand have successfully extracted **D**aily, **W**eekly or **Y**early patterns.

tions failed. That is why numerous hand-crafted models imbued with expert knowledge of the underlying data are probed. The best performing hand-crafted model is a simple auto-regressive process with lags corresponding to periods and harmonics found in the underlying data. Additionally, the performance of Clockwork-RNNs (Koutnik et al., 2014) (CW-RNN) is evaluated. CW-RNNs have an internal representation of periodicity and frequency because subsets of the internal state are updated based on fixed (not learned) periods of internal clocks. We compare to a CW-RNN whose internal clock is instantiated according to the power law as described by the authors but also compare to a CW-RNN whose internal clock ticks according to the periods found in the data. For both models, a single layer CW-RNN was utilized and a parameter sweep over the number of clocks and states per clock was performed. Adjusting the internal clock to the underlying data did not improve the model. Furthermore, we compare to Fourier Neural Networks (Silvescu, 1999; Gashler and Ashmore, 2016) and Echo State Networks (Jaeger, 2007). Fourier NNs are by design not temporal models but merely employ cos-activation functions. Temporal dependencies were attempted to be modeled by learning a function that produces the value of the next time step as a function of the previous 10. For Echo State Networks, the 'easyns' package was employed. Table 5.2.1 shows the results.

Figure 13 exemplifies the pitfalls of time-stepping methods for long-term time series prediction. It shows the frequency content of the prediction of the Koopman- and Fourier-based algorithm alongside the prediction of the best performing LSTM. In all experiments, time stepping methods like LSTMs, GRUs or ESNs have two important limitations. First, because their objective is one-step-ahead prediction and slow frequencies incur little error in the short-term, they fail to extract slow frequencies from data. Second, the resulting

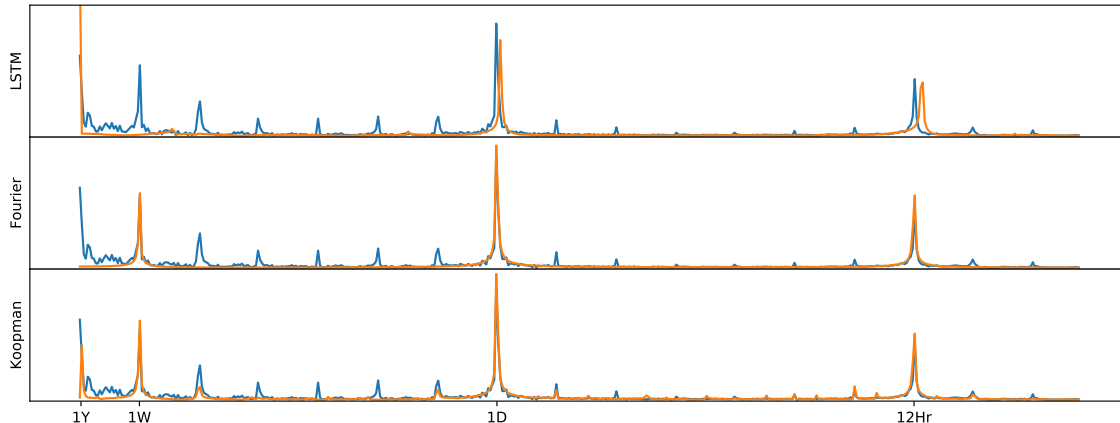


Figure 13: Comparison of the best performing model of each contender in frequency domain. The x-axis shows frequencies with periods of one year, one week, one day and 12 hours. The LSTM fails to extract slow frequencies and seems to be a biased frequency estimator.

predictions are biased in terms of frequency. When integrating the LSTM (recursively feeding predictions back into the network), the resulting time series does oscillate but not with the correct frequency. This behavior is detrimental, especially when the Mean Squared Error is employed to evaluate the performance. Predicting a constant often incurs a smaller error than predicting an out-of-phase oscillation.

5.2.2 FLUID FLOWS AND VIDEO FRAME PREDICTION

There seems to be a dichotomy in fluid flows. On one hand, there are noise amplifying flows and on the other hand there are oscillator flows. Because of the inductive bias of the algorithms introduced in this work, we will focus on evaluating the performance of the proposed algorithms in modeling the behavior of oscillator flows. The performance is compared to Dynamic Mode Decomposition (DMD) and a variant of DMD designed to be more noise robust, namely the forward-backward DMD algorithm (Hemati et al., 2017). Modeling fluid flows is a long-standing standard benchmark for DMD-like algorithms. Indeed, DMD was first introduced to extract the dynamic behavior in fluids (Kutz et al., 2016). Note that in the following experiments, the dimensionality of the high dimensional flows was reduced by Principal Component Analysis and the temporal forecasting was performed using the Fourier-based algorithm. For each experiment, the first 75% of the data was used for training whereas the remaining 25% are used for testing (temporal split).

Table 2 shows the performance of the Fourier based algorithm graphically, specifically it shows the last predicted frame when predicting a Kolmogorov 2D flow, flow around a cylinder, flow over a cavity and a video of a fan. The data from the Kolmogorov 2D flow was taken from the experiments conducted in Tithof et al. (2017). Note that the Kolmogorov 2D flow and the data for the video frame prediction task constitute real measurements and therefore exhibit a considerable amount of noise, whereas the cylinder and cavity flow data

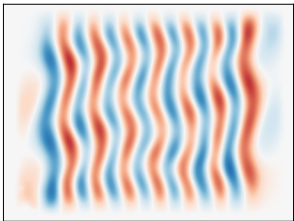
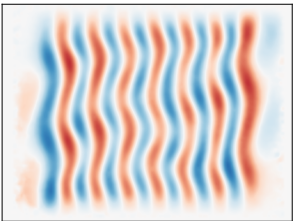
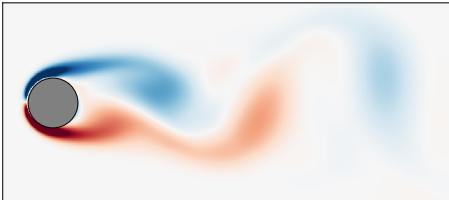
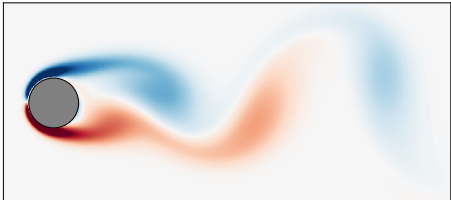
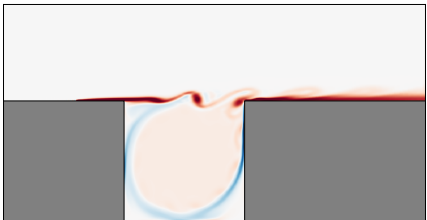
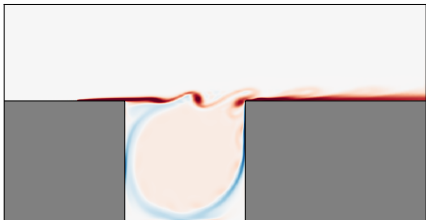


	Prediction	Ground Truth
Kolmogorov 2D Rey. no.: ≈ 10		
Cylinder flow Rey. no.: ≈ 100		
Flow over cavity Rey. no.: ≈ 7500		
Video frame -		

Table 2: The last frame as predicted by PCA in conjunction with the Fourier-based algorithm of fluid flows and video frame prediction. For a video that shows the performance, visit <https://www.youtube.com/watch?v=trbXYMqi2Tw>.

stems from simulation.

The test set was split into four equal length subsets and Table 3 shows the performance as measured by the relative cumulative error on each split. For each task, a parameter sweep over the number of frequencies was performed and the best performing model was selected. Note that the performance was measured in PCA-space, i.e. in predicting PCA modes. The results suggest that the Fourier-based algorithm significantly outperforms DMD and FB-DMD on three out of four tasks. This is mainly due to the fact that under the assumption of measurement noise, DMD is a biased frequency estimator Hemati et al. (2017). On top

	Algorithm	Performance			
Kolmogorov 2D Reynolds no. ≈ 10	Fourier	1.1%	1.4%	1.57%	1.69%
	DMD	175%	171%	169%	165%
	FB DMD	<i>inf</i>	<i>inf</i>	<i>inf</i>	<i>inf</i>
Cylinder flow Reynolds no. 100	Fourier	0%	0%	0%	0.1%
	DMD	4.44%	4.91%	4.71%	4.69%
	FB DMD	4.22%	4.75%	4.68%	4.56%
Flow over cavity Reynolds no. 7500	Fourier	0.9%	1%	1%	1%
	DMD	69%	66.7%	64.9%	63.8%
	FB DMD	63%	70.6%	77.2%	84%
Video frame	Fourier	12.3%	17.2%	22.8%	27.3%
	DMD	7%	11%	16%	22%
	FB DMD	194%	196%	196%	195%
Prediction Horizon		25%	50%	75%	100%

Table 3: The test set was split into four consecutive subsets of equal length. For every quantile, the relative cumulative error is reported for the Fourier-based algorithm, DMD and forward-backward DMD.

of that, DMD based approaches oftentimes lack stability as eigenvalues can lie outside the unit-circle causing predictions to either blow up to infinity (as was the case of FB-DMD applied to the Kolmogorov flow) or crash into the origin. The reason why DMD outperforms the Fourier-based algorithm on the video frame prediction task is simple: one of the PCA modes does not fit the inductive bias of the algorithm, i.e. it does not oscillate but it increases over time, therefore impeding the performance of the algorithm.

5.3 Process noise and phase drifts

The question arises whether process noise as opposed to measurement noise can be modeled with the algorithms introduced here. Human gait is known to be a oscillator that however exhibits a considerable amount of process noise (Zielińska, 1996). We model process noise as a phase drift and slightly modify the optimization objective of the Fourier algorithm:

$$L(A, \vec{\omega}, \phi_t, t) = \|x_t - A\Omega(\vec{\omega}t + \hat{\phi}_t)\|_2^2 + |\hat{\phi}_t - \hat{\phi}_{t-1}|$$

We test the modified algorithm on a data set containing accelerometer readings of a mobile phone located in test subjects pockets (subject 28, right pocket) (Vajdi et al., 2019) and compare to the Fourier algorithm. Figure 14 shows the first portion of the predictions of the Fourier algorithm alongside the phase-corrected modification (Fourier PC). Thus, small modifications to the algorithms introduced in this work allow to model process noise.

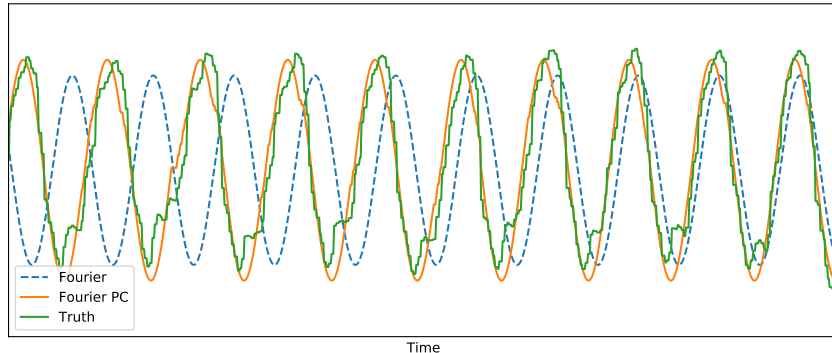


Figure 14: LocationZ recordings from subject 28. We visually compare the Fourier algorithm and its phase-corrected sibling (Fourier PC) to the true readings.

6. Conclusion

In this work, techniques to fit linear and nonlinear oscillators to data are introduced. When modeling quasi-periodic systems, the optimization objectives for both problems are highly non-convex and are therefore not amenable to be solved by gradient descent. The key insight to solving both optimization problems is to express the objectives in frequency domain where solutions can be found in a computationally efficient and scalable manner thanks to the computational properties of the Fast Fourier Transform. Because the optimization objectives are global, the algorithms introduced in this work allow for long-term forecasting while mitigating compounding errors. In the following, advantages and disadvantages of the algorithms are discussed.

6.1 Disadvantages

Strong inductive bias: The algorithms introduced here exhibit a strong inductive bias. The algorithms will not work well for data that stems from systems which are not *quasi-periodic*. However, we have shown empirically that many real life phenomena such as fluid flows, space weather and energy consumption seem to fulfill this assumption.

Unknown phase problem: This problem was alluded to earlier. For the Koopman algorithm, because analytical solutions for parameters Θ when f is non-linear are usually unobtainable, swings in either direction from the ‘error plateau’ can be indicative of a correct choice of frequency.

Conditional Independence of predictions: Given model parameters, pairs of predicted values are conditionally independent, i.e. $p(x_{t_1}, x_{t_2} | \Theta, \vec{\omega}) = p(x_{t_1} | \Theta, \vec{\omega}) p(x_{t_2} | \Theta, \vec{\omega})$. While this property is beneficial for *compounding errors* and *time stepping* (see below) this property can also have adverse effects. Consider the scenario where the goal is to perform a one-step ahead prediction for $t + 1$ and all previous data points up to t are known. If the model error for the prediction of x_t is big, the prediction error for

x_{t+1} will most likely also be high. Note that in this scenario, we assume we know x_t but even though, we know the true value of x_t , incorporating this knowledge, i.e. the conditional dependence on x_{t+1} , into the model is not trivial.

6.2 Advantages

Strong inductive bias: On the flip side, if the data is known to stem from a *quasi-periodic* system, the algorithms introduced here offer a tight fit to the data. The resulting predictions are guaranteed to be quasi-periodic by construction therefore removing potential possibilities of failure like e.g. predictions exploding to plus/minus infinity.

Time stepping: Past values are not required to make future state predictions when model parameters are known. This entails that time consuming time stepping methods are not required. This is especially useful if the data is high dimensional. This is closely related to the fact that predictions are conditionally independent.

Compounding errors: Again, because a global optimization criterion (as opposed to one step-ahead predictions) is solved and because predictions are conditionally independent, errors do not compound. However, because of uncertainty caused by noise, errors will still accumulate over time but in a provably linear manner.

Noise resilience: Because of the strong inductive bias, the algorithms introduced in this work exhibit very strong resiliency towards noise. As an anecdote, when Jaynes first published his results on *Bayesian Spectral Analysis* (Jaynes, 1987), because he was able to resolve frequencies with much higher accuracy than Nyquist’s theorem would suggest, his work was compared to ‘squaring a circle’ and initially dismissed.

7. Future Work

In the opinion of the authors, the algorithms presented in this work create new and interesting potential future research paths. The following outlines some ideas for future work.

Input/output systems: In its current inception, the algorithms introduced in this work assume the underlying systems to be autonomous, i.e. systems that are not driven by inputs. This severely limits the applicability of the algorithms in more general contexts. Koopman theory and DMD have recently been extended to include externally forced systems (Proctor et al., 2016, 2018). This idea could, in principle, be used to create new recurrent Neural Network (RNN) architectures that abandon nonlinearities in between time steps for linear systems in conjunction with a time-invariant nonlinearity. It may also lead to improved performance and could potentially overcome the issues that arise when training current RNNs, like e.g. exploding or vanishing gradients through time.

Interpretable flows: When modeling the fluid flows little regard to the interpretability of the resulting model was given. Even though the flow over the cavity could, in principle, be modeled using a single frequency given an appropriate nonlinearity that is amenable to rotations, so far, the dimensionality was reduced using Principal Component Analysis and forecasting was performed using the Fourier-based algorithm.

Employing the Koopman algorithm in conjunction with a nonlinear decoder that can handle rotations could reduce the number of required frequencies and result in higher noise robustness and interpretability.

Non-parametric Fourier/Koopman: In this work, we showed a direct link between time series prediction and Bayesian Spectral Analysis (BSA). However, the insights from BSA are, so far, merely used to derive error bounds. These error bounds can, in principle, be used to influence predictions. One could conceive a Bayesian probabilistic model that marginalizes the frequencies out, i.e. by performing predictions by computing the following quantity: $\int p(x_{T+h}|\vec{\omega})p(\vec{\omega}|x_{1:T})d\vec{\omega}$. Note that Bayesian Spectral Analysis provides ways to calculate $p(\vec{\omega}|x_{1:T})$.

Local and global modeling: Consider the following scenario: The goal is to predict a signal with seasonal patterns given multiple years of training data. Imagine also that the current year is anomalous which entails that short-term predictions will be impeded. In its current inception, because global optima are sought, the information that the system is currently in an anomalous state is hard to incorporate. Ideally, in addition to one of the global models introduced in this work, there is also a local model and with increasing prediction horizon, more weight should be given to the global over the local model. Optimally weighing local and global models seem an interesting and important future research path.

Acknowledgements

The authors acknowledge support from the Defense Advanced Research Projects Agency (DARPA PA-18-01-FP-125) and the Army Research Office (ARO W911NF-17-1-0306). We would like to thank Charles Delahunt for feedback on the manuscript, and Jeffrey Tithof, Jared Callahan, Kadierdan Kaheman, Benjamin Herrmann and Isabel Scherl for providing data sets.

References

- J. S. Armstrong. *Long-range forecasting*. Wiley New York ETC., 1985.
- T. Askham and J. N. Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.
- S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.
- A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013.
- G. D. Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 17(12):656–660, 1931.
- G. D. Birkhoff and B. Koopman. Recent contributions to the ergodic theory. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):279, 1932.

- G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- G. L. Bretthorst. *Bayesian spectrum analysis and parameter estimation*, volume 48. Springer Science & Business Media, 2013.
- B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, 258:1–15, 2016a.
- S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11(2):e0150171, 2016b.
- S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1):1–9, 2017.
- M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.
- K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.
- S.-C. Chan, K. M. Tsui, H. Wu, Y. Hou, Y.-C. Wu, and F. F. Wu. Load/price forecasting and managing demand response for smart grids: Methodologies and challenges. *IEEE signal processing magazine*, 29(5):68–85, 2012.
- K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.
- J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- S. T. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.
- N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. *Journal of Real-Time Image Processing*, 16(5):1479–1492, 2019a.
- N. B. Erichson, M. Muehlebach, and M. W. Mahoney. Physics-informed autoencoders for Lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019b.
- J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

- J. Fourier. *Theorie analytique de la chaleur, par M. Fourier*. Chez Firmin Didot, père et fils, 1822.
- M. S. Gashler and S. C. Ashmore. Modeling time series data with deep Fourier neural networks. *Neurocomputing*, 188:3–11, 2016.
- J. D. Hamilton. *Time series analysis*, volume 2. Princeton New Jersey, 1994.
- M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- R. J. Hyndman, Y. Khandakar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics . . . , 2007.
- H. Jaeger. Echo state network. *scholarpedia*, 2(9):2330, 2007.
- E. Jaynes. Bayesian spectrum and chirp analysis. In *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, pages 1–37. Springer, 1987.
- T. V. Jensen and P. Pinson. Re-europe, a large-scale dataset for modeling a highly renewable european electricity system. *Scientific data*, 4:170175, 2017.
- X. Jiang and H. Adeli. Dynamic wavelet neural network model for traffic flow forecasting. *Journal of transportation engineering*, 131(10):771–779, 2005.
- M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz. Time-delay observables for Koopman: Theory and applications. *arXiv preprint arXiv:1810.01479*, 2018.
- B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315, 1931.
- B. O. Koopman and J. v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255, 1932.
- J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.
- J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- Y. Lan and I. Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013.
- Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

- D. A. Lyon. The discrete Fourier transform, part 4: spectral leakage. *Journal of object technology*, 8(7), 2009.
- C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk. prdeep: Robust phase retrieval with a flexible deep network. *arXiv preprint arXiv:1803.00212*, 2018.
- I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- I. Mezic. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.
- I. Mezić. On applications of the spectral theory of the Koopman operator in dynamical systems and control theory. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 7034–7041. IEEE, 2015.
- J. P. Moeck, J.-F. Bourgoign, D. Durox, T. Schuller, and S. Candel. Tomographic reconstruction of heat release rate perturbations induced by helical modes in turbulent swirl flames. *Experiments in fluids*, 54(4):1498, 2013.
- C. C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015.
- J. v. Neumann. Physical applications of the ergodic hypothesis. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):263, 1932a.
- J. v. Neumann. Proof of the quasi-ergodic hypothesis. *Proceedings of the National Academy of Sciences*, 18(1):70–82, 1932b.
- S. E. Otto and C. W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.
- S. Pan and K. Duraisamy. Physics-informed probabilistic learning of linear embeddings of non-linear dynamics with guaranteed stability. *arXiv preprint arXiv:1906.03663*, 2019a.
- S. Pan and K. Duraisamy. On the structure of time-delay embedding in linear models of non-linear dynamical systems. *arXiv preprint arXiv:1902.05198*, 2019b.
- A. Pankratz. *Forecasting with univariate Box-Jenkins models: Concepts and cases*, volume 224. John Wiley & Sons, 2009.
- Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. Optoelectronic reservoir computing. *Scientific reports*, 2:287, 2012.
- J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.
- J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

- J. L. Proctor, S. L. Brunton, and J. N. Kutz. Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.
- C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.
- Y. Sakamoto, M. Ishiguro, and G. Kitagawa. Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel*, 81, 1986.
- W. Saxton. *Computer techniques for image processing in electron microscopy*, volume 10. Academic Press, 2013.
- P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- A. Silvescu. Fourier neural networks. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings*, volume 1, pages 488–491. IEEE, 1999.
- G. Song, F. Alizard, J.-C. Robinet, and X. Gloerfelt. Global and Koopman modes analysis of sound generation in mixing layers. *Physics of Fluids*, 25(12):124101, 2013.
- J. C. Spall. Cyclic seesaw process for optimization and identification. *Journal of optimization theory and applications*, 154(1):187–208, 2012.
- N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.
- J. Tithof, B. Suri, R. K. Pallantla, R. O. Grigoriev, and M. F. Schatz. Bifurcations in a quasi-two-dimensional kolmogorov-like flow. *Journal of Fluid Mechanics*, 828:837–866, 2017.
- P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.
- J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.
- A. Vajdi, M. R. Zaghian, S. Farahmand, E. Rastegar, K. Maroofi, S. Jia, M. Pomplun, N. Haspel, and A. Bayat. Human gait database for normal walk collected by smart phone accelerometer. *arXiv preprint arXiv:1905.03109*, 2019.
- C. Wehmeyer and F. Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of chemical physics*, 148(24):241703, 2018.

- E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839. IEEE, 2019.
- T. Zielińska. Coupled oscillators utilised as gait rhythm generators of a two-legged walking machine. *Biological Cybernetics*, 74(3):263–273, 1996.