An approach for the reduction of boundary effects in time-frequency and time-scale representations

1 Introduction

In any digital acquisition system, the study and the interpretation of the measured signals generally require the use an analysis tool, which enable to point out the useful characteristics of the signal. For example, any measured physiological signal, such as photoplethysmogram (PPG), may not be interpreted as it is, from its run chart. An analysis tool would make possible the extraction of some useful characteristics such as heart rate or blood pressure.

In general, the observed quantities are produced by transient phenomena, that can vary rapidly and irregularly. As a consequence, the measured signals exhibit nonstationary behavior. In order to adapt the analysis to nonstationarities, local analysis is generally performed. The short time Fourier transform (STFT) is a typical tool which is build that way, and allows to determine locally the frequency content of a nonstationary signal.

Windowing is a common method for performing local analysis. Among others, STFT, wavelet transform [3], synchrosqueezing transform (SST), and reassignment [1] are representations that fall back on the use of an analysis window. Let $x : I \to \mathbb{R}$ denote the observed signal, where I denotes the finite interval where the signal is measured. Let $g_s : \mathbb{R} \to \mathbb{R}$ denote the analysis window, where s is a shape parameter. The support of g_s is is localized around the origin, and is small with respect to $|I|$. The translation operator is T_τ defined as:

$$T_\tau f = f(t - \tau) , \quad \forall f : \mathbb{R} \to \mathbb{R} .$$

Then, the local analysis of x around the instant $\tau \in I$ rely on the evaluation of the following dot product:

$$V_x(s, \tau) = \langle x, T_\tau g_s \rangle_I . \tag{1}$$

A major shortcoming of this technique occurs when analyzing the signal x near the boundaries of the interval I. Indeed, at these points, half of the information is missing. Consequently the results of the dot product 1 is disturbed by this lack of information. This phenomenon is the so-called *boundary effect*. We display, on the left of Fig. 1 the result of the SST of a PPG (see section 4.2.3 for a comprehensive description). The distortion resulting from the boundary effect is clearly visible on both sides of this representation. Indeed, while in the major central part of the image, clear lines stand out, they become blurred as they approach the left and right boundaries of the image. A zoom on the right boundary of this SST, displayed on the right of Fig. 1, emphasizes the result of boundary effect. More generally, signal characteristics, like instantaneous frequencies or amplitudes appear to be imprecisely determinable in the vicinity of the boundaries.

The purpose of this paper is to provide a fast algorithm allowing to tackle boundary effect and limit the disruptions it may cause on the signals representations based on windowing. In that aim, we proceed in two steps:

1. *Extend the signal by forecasting it.* The aim is to use a dynamic model to predict the values taken by the measured signal outside the measurement interval. Then, once this operation is done, we have access to an extended signal defined on a larger interval I_Δ, where Δ denotes the size of the extension on both boundaries of I.

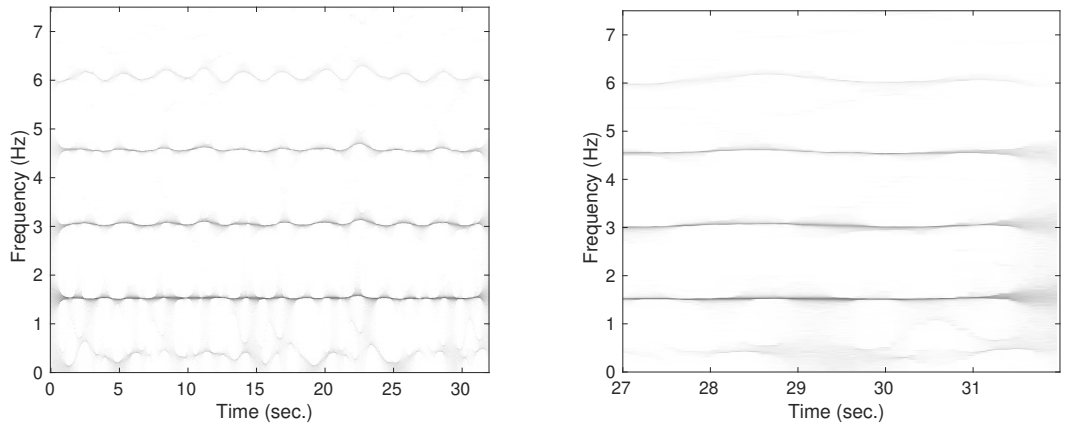

Figure 1: Synchrosqueezing transform of a PPG (left) with a zoom on its right boundary (right).

2. *Run the local analysis tool on the extended signal.* Assuming that the support of the analysis window is smaller that 2Δ, the local analysis near the boundary of I is now possible without lack of information thanks to knowledge brought by the extension on both sides.

Thus, assuming that the quality of the extension step is sufficient, the representation obtained that way will be less sensitive to the boundary effect than the result of the analysis tool applied directly to the non-extended signal. Notice that, in order to obtain a fast algorithm the extension step we propose must have a low computational cost.

2 Algorithm

2.1 Forecasting

Notations. Let $x : \mathbb{R} \to \mathbb{R}$ denote a continuous-time signal. In this work, we consider a finite-length discretization of that one. Thus, the sampled signal $\mathbf{x}$, whose length is denoted by N, is such that

$$\mathbf{x}[n] = x\left(\frac{n}{f_\mathsf{s}}\right) , \quad \forall n \in \{0, \dots, N-1\} ,$$

where f_s denotes the sampling frequency.

Let $M < N$, $K < N$. Then, for all $k \in \{0, \dots, K-1\}$, we extract from $\mathbf{x} \in \mathbb{R}^N$ the sub-signal $\mathbf{x}_k \in \mathbb{R}^M$ given by:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{x}[N-K+(k-1)-(M-1)] \\ \vdots \\ \mathbf{x}[N-K+(k-1)] \end{pmatrix} .$$

These sub-signals are gathered into the matrix $\mathbf{X} \in \mathbb{R}^{M \times K}$ such that:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_0 & \cdots & \mathbf{x}_{K-1} \end{pmatrix} .$$

Notice that these sub-signals are overlapping each other. Indeed, $\mathbf{x}_{k+1}$ is a shifting of $\mathbf{x}_k$ from one sample.

We also consider the matrix $\mathbf{Y} \in \mathbb{R}^{M \times K}$ given by:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_K \end{pmatrix} .$$

Dynamic model and forecasting. Establishing a dynamic model consists in determining the relation linking $\mathbf{Y}$ to $\mathbf{X}$, that is finding a function f so that

$$\mathbf{Y} = f(\mathbf{X}) \, .$$

We consider here a naive dynamic model, assuming that we have the following relation:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \, , \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{M \times M}$. This is a linear dynamic model, which can be written equivalently in function of the sub-signals $\mathbf{x}_k$, as:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \, , \forall k \in \{0, \ldots, K-1\} \, . \tag{3}$$

The forecasting method consists in estimating the unknown matrix $\mathbf{A}$. Indeed, let $\tilde{\mathbf{A}}$ denotes the estimate of $\mathbf{A}$, we then obtain the forecasting of the signal at time $\frac{N-1+\ell}{f_s}$ by:

$$\tilde{\mathbf{x}}[N-1+\ell] = \mathbf{e}_M^T \tilde{\mathbf{A}}^\ell \mathbf{x}_K \, , \tag{4}$$

where $\mathbf{e}_M$ is the vector of length M given by $\mathbf{e}_M = \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}^T$.

Model estimation. To estimate the matrix $\mathbf{A}$, we basically implement the least square estimator. Thus, we solve the following problem:

$$\tilde{\mathbf{A}} = \arg\min_{\boldsymbol{\alpha}} \mathcal{L}(\mathbf{A}) \, , \tag{5}$$

where the loss function $\mathcal{L}$ is given by:

$$\mathcal{L}(\mathbf{A}) = \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|^2 = \sum_{k=0}^{K-1} \|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k\|^2 .$$

Therefore, solving the problem (5), i.e. $\nabla \mathcal{L}(\tilde{\mathbf{A}}) = \mathbf{0}$, gives the following estimate $\tilde{\mathbf{A}}$ of the dynamic model matrix $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \, . \tag{6}$$

Remark 1. *This expression clearly shows that the matrix $\tilde{\mathbf{A}}$ takes the following form:*

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \alpha_1 & \cdots & \cdots & \cdots & \alpha_M \end{pmatrix} .$$

Then, except the row vector $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_M)$, the matrix $\mathbf{A}$ is fully determined by the dynamic model.

Signal extension. We can finally construct the extended signal $\tilde{\mathbf{x}} \in \mathbb{R}^{N+2L}$ concatenating the backward prediction $\tilde{\mathbf{x}}_{\text{bw}}$, the observed signal $\mathbf{x}$, and the forward prediction $\tilde{\mathbf{x}}_{\text{fw}}$. We summarize the extension step in Algorithm 1. Notice that we handle the backward estimation using the same strategy than described above, but applying it to the reverse signal $\mathbf{x}^{\text{r}} = \begin{pmatrix} \mathbf{x}[N-1] & \cdots & \mathbf{x}[0] \end{pmatrix}^T$.

Algorithm 1 Signal extension. $\tilde{\mathbf{x}} = \mathsf{SigExt}(\mathbf{x}, M, K, L)$

Inputs: $\mathbf{x}$, M, K, L

Forward forecasting.
- LS estimation of the forward matrix $\tilde{\mathbf{A}}_{\text{fw}}$ via equation (6).
- Forward forecasting $\tilde{\mathbf{x}}_{\text{bw}}$ obtained applying equation (4) with $\ell \in \{1, \dots, L\}$.

Backward forecasting.
- Reverse signal $\mathbf{x}$ to $\mathbf{x}^{\text{r}}$.
- LS estimation of the backward matrix $\tilde{\mathbf{A}}_{\text{bw}}$ via equation (6) applied to $\mathbf{x}^{\text{r}}$.
- Reversed backward forecasting $\tilde{\mathbf{x}}^{\text{r}}_{\text{bw}}$ obtained applying equation (4) to $\mathbf{x}^{r}$ with $\ell \in \{1, \dots, L\}$.
- Reverse $\tilde{\mathbf{x}}^{\text{r}}_{\text{bw}}$ to obtain the estimate $\tilde{\mathbf{x}}_{\text{bw}}$.

Output: Extended signal $\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{\mathbf{x}}_{\text{bw}} & \mathbf{x} & \tilde{\mathbf{x}}_{\text{fw}} \end{pmatrix}^T$.

2.2 Representation

Transform restriction. Let $\mathscr{F}_N : \mathbb{R}^N \to \mathbb{R}^{F \times N}$ generically denotes the time-frequency or time-scale representation we are interested in. It can be, for instance, such as short-time Fourier transform (STFT), the continuous wavelet transform (CWT), the synchrosqueezing transform (SST), or the reassignment (RS). Here, F typically denotes the size of the representation along the frequency axis. Due to the boundary effects, the representation $\mathscr{F}_N(\mathbf{x})$ shows undesired patterns when approaching its edges.For example, the instantaneous frequencies highlighted by the SST can be blurred near that edges. To limit these phenomena, we apply the representation to the estimated extended signal $\tilde{\mathbf{x}}$. This strategy moves the boundary effects out of the time interval $I = [0, \frac{N-1}{f_{\text{s}}}]$. Finally, the boundary-effects insensitive representation $\mathscr{F}^{\text{ext}}_N : \mathbb{R}^N \to \mathbb{R}^{F \times N}$ of $\mathbf{x}$ is given by:

$$\mathscr{F}^{\text{ext}}_N(\mathbf{x})[f, n] = \mathscr{F}_{N+2L}(\tilde{\mathbf{x}})[f, L + n] \,, \quad \forall f \in \{0, \cdots, F-1\},\ n \in \{0, \cdots, N-1\} \,. \tag{7}$$

This amounts to restricting the representation $\mathscr{F}_{N+2L}(\tilde{\mathbf{x}})$ to the original measurement interval of $\mathbf{x}$. For the sake of simplicity, we denotes the restriction operator by $\mathcal{R}$, where $\mathcal{R} : \mathbb{R}^{F \times (N+2L)} \to \mathbb{R}^{F \times N}$. Consequently, we have:

$$\mathscr{F}^{\text{ext}}_N(\mathbf{x}) = \mathcal{R}\left(\mathscr{F}_{N+2L}(\tilde{\mathbf{x}})\right) \,.$$

Conclusion. Finally, the global procedure we implement to reduce boundary effects on windowing-based representations is summarized by the pseudo-code of Algorithm 2.

Algorithm 2 Tackling boundary effects. $\mathbf{F}_{\mathbf{x}} = \mathsf{BoundEffRed}(\mathbf{x}, M, K, L, \mathscr{F})$

Inputs: $\mathbf{x}$, M, K, L, $\mathscr{F}$

Forecasting step.
- Signal extension: $\tilde{\mathbf{x}} = \text{SigExt}(\mathbf{x})$.

Representation step.
- Representation evaluation: $\mathscr{F}_{N+2L}(\tilde{\mathbf{x}})$.
- Restriction of $\mathscr{F}_{N+2L}(\tilde{\mathbf{x}})$ to the central time interval (see (7)) to obtain $\mathbf{F}_{\mathbf{x}} = \mathscr{F}^{\text{ext}}_N(\mathbf{x})$.

Output: Signal representation $\mathbf{F}_{\mathbf{x}}$

3 Theoretical performance

3.1 General signal model

We model the observed signal as a given waveform corrupted by an additive Gaussian white noise. Therefore, the measured discrete signal $\mathbf{x}$ is written as:

$$\mathbf{x} = \mathbf{z} + \sigma\mathbf{w} \,,$$

where $\mathbf{z}$ is a deterministic signal, $\mathbf{w}$ is a (Gaussian) white noise, whose variance is normalized to one. Thus, σ^2 denotes the variance of the additive noise $\sigma\mathbf{w}$.

3.2 Forecasting error

On the forecasting interval, we decompose the estimated signal $\tilde{\mathbf{x}}$ as follows:

$$\tilde{\mathbf{x}}[N-1+\ell] = \mathbf{z}[N-1+\ell] + \boldsymbol{\epsilon}[\ell] \,,$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^L$ is a the forward forecasting error, which would ideally behave like the measurement noise $\sigma\mathbf{w}$. To evaluate the actual behavior of the forward forecasting error $\boldsymbol{\epsilon}$, we determine its two first moments.

1. The mean, which is also the estimation bias, is such that:

$$\begin{aligned}\boldsymbol{\mu}[\ell] &\triangleq \mathbb{E}\{\boldsymbol{\epsilon}[\ell]\} \\ &= \mathbb{E}\{\mathbf{e}_M^T\tilde{\mathbf{A}}^\ell\}\mathbf{z}_K + \mathbb{E}\{\mathbf{e}_M^T\tilde{\mathbf{A}}^\ell\mathbf{w}_K\} - \mathbf{z}[N-1+\ell] \\ &= \mathbb{E}\{\boldsymbol{\alpha}^{(\ell)}\}\mathbf{z}_K + \sigma\mathbb{E}\{\boldsymbol{\alpha}^{(\ell)}\mathbf{w}_K\} - \mathbf{z}[N-1+\ell] \end{aligned} \tag{8}$$

   where $\boldsymbol{\alpha}^{(\ell)} \in \mathbb{R}^{1\times M}$ denotes the last row of $\tilde{\mathbf{A}}^\ell$.

2. The covariance is given by:

$$\boldsymbol{\gamma}[\ell,\lambda] \triangleq \mathbb{E}\{(\boldsymbol{\epsilon}[\ell] - \boldsymbol{\mu}[\ell])\,(\boldsymbol{\epsilon}[\lambda] - \boldsymbol{\mu}[\lambda])\}$$

   By means of the Cauchy-Schwarz inequality the covariance, we provide an upper bound on the covariance:

$$|\boldsymbol{\gamma}[\ell,\lambda]| \leq \sqrt{\boldsymbol{\gamma}[\ell,\ell]\boldsymbol{\gamma}[\lambda,\lambda]} \,. \tag{9}$$

   We then focus on the determination of $\boldsymbol{\gamma}[\ell,\ell]$. Thus:

$$\begin{aligned}\boldsymbol{\gamma}[\ell,\ell] &= \mathbb{E}\{\boldsymbol{\epsilon}[\ell]^2\} - \boldsymbol{\mu}[\ell]^2 \\ &= \mathbf{z}[N-1+\ell]^2 - 2\mathbf{z}[N-1+\ell]\mathbb{E}\{\tilde{\mathbf{x}}[N-1+\ell]\} + \mathbb{E}\{\tilde{\mathbf{x}}[N-1+\ell]^2\} - \boldsymbol{\mu}[\ell]^2 \,.\end{aligned}$$

   But:

$$\mathbb{E}\{\tilde{\mathbf{x}}[N-1+\ell]^2\} = \mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{z}_K)^2\} + 2\sigma\mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{z}_K)(\boldsymbol{\alpha}^{(\ell)}\mathbf{w}_K)\} + \sigma^2\mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{w}_K)^2\}$$

   The mean square error can therefore be written as follows:

$$\begin{aligned}\boldsymbol{\gamma}[\ell,\ell] = &-\mathbf{z}[N-1+\ell]^2 - 2\mathbf{z}[N-1+\ell]\boldsymbol{\mu}[\ell] + \mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{z}_K)^2\} \\ &+ 2\sigma\mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{z}_K)(\boldsymbol{\alpha}^{(\ell)}\mathbf{w}_K)\} + \sigma^2\mathbb{E}\{(\boldsymbol{\alpha}^{(\ell)}\mathbf{w}_K)^2\} - \boldsymbol{\mu}[\ell]^2 \,.\end{aligned} \tag{10}$$

Decomposition of $\boldsymbol{\alpha}^{(\ell)}$. For all $m, m' \in \{0, \dots, M-1\}$, we have:

$$(\mathbf{X}\mathbf{X}^T)[m,m'] = \sum_{k=0}^{K-1} \mathbf{x}[\underbrace{N-K-M}_{\triangleq N_0} + m + k]\mathbf{x}[N-K-M+m'+k] = K(\mathbf{S}^{(0)} + \mathbf{E}^{(0)}) \tag{11}$$

$$(\mathbf{Y}\mathbf{X}^T)[m,m'] = \sum_{k=0}^{K-1} \mathbf{x}[N_0+m+1+k]\mathbf{x}[N_0+m'+k] = K(\mathbf{S}^{(1)} + \mathbf{E}^{(1)})\,, \tag{12}$$

where

$$\mathbf{S}^{(a)}[m,m'] = \sigma^2\delta_{(m+a)m'} + \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{z}[N_0+m+a+k]\mathbf{z}[N_0+m'+k]\,,$$

and $\mathbf{E}^{(a)} = \sigma\mathbf{E}_1^{(a)} + \sigma^2\mathbf{E}_2^{(a)}$ with:

$$\mathbf{E}_1^{(a)}[m,m'] = \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{z}[N_0+m+a+k]\mathbf{w}[N_0+m'+k] + \mathbf{w}[N_0+m+a+k]\mathbf{z}[N_0+m'+k]\,,$$

and

$$\mathbf{E}_2^{(a)}[m,m'] = \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{w}[N_0+m+a+k]\mathbf{w}[N_0+m'+k] - \delta_{(m+a)m'}\,,$$

with $a \in \{0,1\}$.

Remark 2. *The matrices $\mathbf{E}^{(0)}$ and $\mathbf{E}^{(1)}$ are said to be error matrices because:*

$$\mathbb{E}\{\mathbf{E}^{(0)}\} = \mathbb{E}\{\mathbf{E}_1^{(0)}\} = \mathbb{E}\{\mathbf{E}_2^{(0)}\} = \mathbf{0}$$
$$\mathbb{E}\{\mathbf{E}^{(1)}\} = \mathbb{E}\{\mathbf{E}_1^{(1)}\} = \mathbb{E}\{\mathbf{E}_2^{(1)}\} = \mathbf{0}\,.$$

Thus:

$$\tilde{\mathbf{A}} = (\mathbf{S}^{(1)} + \mathbf{E}^{(1)})(\mathbf{S}^{(0)} + \mathbf{E}^{(0)})^{-1}\,.$$

Let $\tilde{\mathbf{A}}_0 = \mathbf{S}^{(1)}{\mathbf{S}^{(0)}}^{-1}$. Then:

$$\begin{aligned}\boldsymbol{\alpha}^{(\ell)} &= \mathbf{e}_M^T\tilde{\mathbf{A}}^\ell \\ &= \boldsymbol{\alpha}_0^{(\ell)} + \mathbf{h}^{(\ell)}\,,\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{\alpha}_0^{(\ell)} &= \mathbf{e}_M^T\tilde{\mathbf{A}}_0^\ell \\ \mathbf{h}^{(\ell)} &= \mathbf{e}_M^T\left(\left((\mathbf{S}^{(1)} + \mathbf{E}^{(1)})(\mathbf{S}^{(0)} + \mathbf{E}^{(0)})^{-1}\right)^\ell - \tilde{\mathbf{A}}_0^\ell\right)\end{aligned} \tag{13}$$

Furthermore, from equation (8) we have:

$$\boldsymbol{\mu}[\ell] = \boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K + \mathbb{E}\{\mathbf{h}^{(\ell)}\}\mathbf{z}_K + \sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{w}_K\} - \mathbf{z}[N-1+\ell]$$

Besides

$$\begin{aligned}\boldsymbol{\gamma}[\ell,\ell] =& -\mathbf{z}[N-1+\ell]^2 - 2\mathbf{z}[N-1+\ell]\boldsymbol{\mu}[\ell] + (\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K)^2 + 2\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K\mathbb{E}\{\mathbf{h}^{(\ell)}\}\mathbf{z}_K + \mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{z}_K)^2\} \\ &+ 2\sigma\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{w}_K\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\boldsymbol{\alpha}_0^{(\ell)}\mathbf{w}_K\} + \sigma^2\mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{w}_K)^2\} \\ &+ 2\sigma^2\boldsymbol{\alpha}_0^{(\ell)}\mathbb{E}\{\mathbf{w}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + \sigma^2\|\boldsymbol{\alpha}_0^{(\ell)}\|^2 - \boldsymbol{\mu}[\ell]^2 \\ =& \,2\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K\mathbf{z}[N-1+\ell] - (\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K)^2 - \mathbf{z}[N-1+\ell]^2 + 2\left(\boldsymbol{\alpha}_0^{(\ell)}\mathbf{z}_K - \mathbf{z}[N-1+\ell]\right)\boldsymbol{\mu}[\ell] \\ &+ \mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{z}_K)^2\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\boldsymbol{\alpha}_0^{(\ell)}\mathbf{w}_K\} + \sigma^2\mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{w}_K)^2\} \\ &+ 2\sigma^2\boldsymbol{\alpha}_0^{(\ell)}\mathbb{E}\{\mathbf{w}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + \sigma^2\|\boldsymbol{\alpha}_0^{(\ell)}\|^2 - \boldsymbol{\mu}[\ell]^2\,.\end{aligned}$$

3.3 Sum of sine waves

In that section, the deterministic part of the observed signal is assumed to be a multicomponent harmonic signal, that is a sum of sine waves. Then:

$$\mathbf{z}[n] = \sum_{j=0}^{J} \Omega_j \cos\left(2\pi f_j \frac{n}{f_\mathsf{s}}\right) , \tag{14}$$

where J denotes the number of components, Ω_j the amplitude of the j-th component, and f_j its frequency.

Assumption: $f_j = \dfrac{p_j}{M} f_\mathsf{s} = \dfrac{p'_j}{K} f_\mathsf{s}$ for some $p_j, p'_j \in \mathbb{N}^*$.

Evaluation of $\tilde{\mathbf{A}}_0$ and $\boldsymbol{\alpha}_0^{(\ell)} \mathbf{z}_K$.

$$\begin{aligned}
\mathbf{S}^{(a)}[m,m'] &= \sigma^2 \delta_{(m+a)m'} + \sum_{j,j'=1}^{J} \frac{\Omega_j \Omega_{j'}}{K} \sum_{k=0}^{K-1} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(N_0 + m + a + k)\right) \cos\left(2\pi \frac{f_{j'}}{f_\mathsf{s}}(N_0 + m' + k)\right) \\
&= \sigma^2 \delta_{(m+a)m'} + \sum_{j=1}^{J} \frac{\Omega_j^2}{2K} \sum_{k=0}^{K-1} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(m + a - m')\right) + \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(2k + m + a + m' + 2N_0)\right) \\
&= \sigma^2 \delta_{(m+a)m'} + \sum_{j=1}^{J} \left(\frac{\Omega_j^2}{2} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(m + a - m')\right) + \frac{\Omega_j^2}{2K} \underbrace{\sum_{k=0}^{K-1} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(2k + m + a + m' + 2N_0)\right)}_{=0 \text{ because } \frac{f_j}{f_\mathsf{s}} = \frac{p'_j}{K}} \right) \\
&= \sigma^2 \delta_{(m+a)m'} + \sum_{j=1}^{J} \frac{\Omega_j^2}{2} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}}(m + a - m')\right) .
\end{aligned}$$

Thus, $\mathbf{S}^{(0)}$ is a circulant matrix and is therefore diagonalizable in the Fourier basis:

$$\mathbf{S}^{(0)} = \mathbf{U} \boldsymbol{\Lambda}^{(0)} \mathbf{U}^* ,$$

where $\mathbf{U}[m,m'] = \frac{1}{\sqrt{M}} e^{-2\mathrm{i}\pi m m'/M}$ and $\boldsymbol{\Lambda}^{(0)} = \mathrm{diag}(\lambda_0^{(0)}, \ldots, \lambda_{M-1}^{(0)})$ with:

$$\begin{aligned}
\lambda_m^{(0)} &= \sigma^2 + \sum_{j=1}^{J} \frac{\Omega_j^2}{2} \sum_{q=0}^{M-1} \cos\left(2\pi \frac{f_j}{f_\mathsf{s}} q\right) e^{-2\mathrm{i}\pi q m/M} \\
&= \sigma^2 + \frac{M}{4} \sum_{j=1}^{J} \Omega_j^2 (\delta_{m,p_j} + \delta_{m,M-p_j}) .
\end{aligned}$$

Therefore:

$$\mathbf{S}^{(0)^{-1}} = \mathbf{U} \boldsymbol{\Lambda}^{(0)^{-1}} \mathbf{U}^*$$

which leads to:

$$\mathbf{S}^{(0)^{-1}}[m,m'] = \frac{1}{\sigma^2} \delta_{m,m'} - \sum_{j=1}^{J} \frac{\Omega_j^2}{2\sigma^2(\sigma^2 + \Omega_j^2 M/4)} \cos\left(2\pi p_j \frac{m - m'}{M}\right) ,$$

and, consequently:

$$\begin{aligned}\tilde{\mathbf{A}}_0[m,m'] &= \sum_{q=0}^{M-1} \mathbf{S}^{(1)}[m,q]{\mathbf{S}^{(0)}}^{-1}[q,m'] \\ &= \delta_{m+1,m'} + \sum_{j=1}^{J} \frac{2\Omega_j^2}{\Omega_j^2 M + 4\sigma^2} \cos\left(2\pi p_j \frac{m'}{M}\right) \delta_{m+1,M} \end{aligned} \tag{15}$$

Thus:

$$\begin{aligned}\tilde{\boldsymbol{\alpha}}_0^{(1)}[m] &= \sum_{j=1}^{J} \frac{2\Omega_j^2}{\Omega_j^2 M + 4\sigma^2} \cos\left(2\pi p_j \frac{m}{M}\right) \\ &= \frac{2}{M} \sum_{j=1}^{J} \cos\left(2\pi p_j \frac{m}{M}\right) + o(\sigma^2)\,. \end{aligned}$$

Besides, from equation (15), we have

$$\tilde{\mathbf{A}}_0 \mathbf{z}_K = \begin{pmatrix} \mathbf{z}[N-M+1] \\ \vdots \\ \mathbf{z}[N-1] \\ \boldsymbol{\alpha}_0^{(1)} \mathbf{z}_K \end{pmatrix}$$

By induction, we have:

$$\tilde{\mathbf{A}}_0^{\ell} \mathbf{z}_K = \begin{pmatrix} \mathbf{z}[N-M+\ell] \\ \vdots \\ \mathbf{z}[N-1] \\ \boldsymbol{\alpha}_0^{(1)} \mathbf{z}_K \\ \vdots \\ \boldsymbol{\alpha}_0^{(\ell)} \mathbf{z}_K \end{pmatrix}$$

Then:

$$\begin{aligned}\boldsymbol{\alpha}_0^{(\ell)} \mathbf{z}_K &= \tilde{\boldsymbol{\alpha}}_0^{(1)} \tilde{\mathbf{A}}_0^{\ell-1} \mathbf{z}_K \\ &= \sum_{m=0}^{M-\ell} \boldsymbol{\alpha}_0^{(1)}[m] \mathbf{z}[N-M+\ell+m-1] + \sum_{m=M-\ell+1}^{M-1} \boldsymbol{\alpha}_0^{(1)}[m] \boldsymbol{\alpha}_0^{(m-M+\ell)} \mathbf{z}_K \end{aligned} \tag{16}$$

But:

$$\begin{aligned}\boldsymbol{\alpha}_0^{(1)} \mathbf{z}_K &= \sum_{m=0}^{M-1} \boldsymbol{\alpha}_0^{(1)}[m] \mathbf{z}[N-M+m] \\ &= \sum_{j,j'=1}^{J} \Omega_{j'} \frac{2}{M} \underbrace{\sum_{m=0}^{M-1} \cos\left(2\pi p_j \frac{m}{M}\right) \cos\left(2\pi p_{j'} \frac{N+m}{M}\right)}_{=\delta_{j,j'} \frac{2}{M} \cos\left(2\pi p_j \frac{N}{M}\right)} + o(\sigma^2) \\ &= \sum_{j=1}^{J} \Omega_j \cos\left(2\pi p_j \frac{N}{M}\right) + o(\sigma^2) \\ &= \mathbf{z}[N] + o(\sigma^2) \end{aligned}$$

and, by induction from (16):

$$\boldsymbol{\alpha}_0^{(\ell)} \mathbf{z}_K = \mathbf{z}[N-1+\ell] + o(\sigma^2) \tag{17}$$

Then:

$$\begin{aligned}\boldsymbol{\mu}[\ell] &= \mathbb{E}\{\mathbf{h}^{(\ell)}\}\mathbf{z}_K + \sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{w}_K\} + o(\sigma^2)\\ \boldsymbol{\gamma}[\ell,\ell] &= \mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{z}_K)^2\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + 2\sigma\mathbb{E}\{\mathbf{h}^{(\ell)}\mathbf{z}_K\boldsymbol{\alpha}_0^{(\ell)}\mathbf{w}_K\} + \sigma^2\mathbb{E}\{(\mathbf{h}^{(\ell)}\mathbf{w}_K)^2\}\\ &\quad + 2\sigma^2\boldsymbol{\alpha}_0^{(\ell)}\mathbb{E}\{\mathbf{w}_K\mathbf{h}^{(\ell)}\mathbf{w}_K\} + \sigma^2\|\boldsymbol{\alpha}_0^{(\ell)}\|^2 - \boldsymbol{\mu}[\ell]^2 + o(\sigma^2)\ .\end{aligned}$$

3.4 Adaptive Harmonic Model

In this section, the the deterministic part of the observed signal we handle follows the *adaptive harmonic model*, which in its is continuous-time version signal takes the following form:

$$z(t) = \sum_{j=1}^{J} a_j(t)\cos(2\pi\phi_j(t))\ , \tag{18}$$

where a_p and ϕ_p' are smooth function. In other terms, we have:

$$|a_j'(t)| < \varepsilon_a, \quad \forall t \in \mathbb{R}\ , \tag{19}$$

$$|\phi_j''(t)| < \varepsilon_\phi, \quad \forall t \in \mathbb{R}\ , \tag{20}$$

for some positive constants ε_a and ε_ϕ.

4 Numerical results

4.1 Evaluation the forecasting performance

In that section, we first evaluate the quality of the forecasting step. That one depends on at least two parameters:

- The noise variance σ^2.
- The size of the training dataset K.

In subsections 4.1.1 and 4.1.2, we study the influence of these parameters. A comparison with the theoretical results of section 3 is also available.

4.1.1 Sum of sine waves

We proved that the linear dynamic model is sufficient to catch the dynamical behavior of signals taking the form (14). In order to validate this theoretical result, we apply the forecasting Algorithm 1 to a large number of realizations of the random vector $\mathbf{z}$ such that:

$$\mathbf{z} = \mathbf{x} + \sigma\mathbf{w}\ ,$$

where

$$\mathbf{x}[n] = \cos\left(2\pi p_1\frac{n}{M}\right) + R\cos\left(2\pi p_2\frac{n}{M}\right)\ , \quad \forall\, n \in \{1,\dots,N\}\ ,$$

and $N = 10^4$, $M = 750$, $p_1 = 10$, $p_2 = 33$ and $R = 1.4$. Besides, the additive noise is chosen to be Gaussian: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

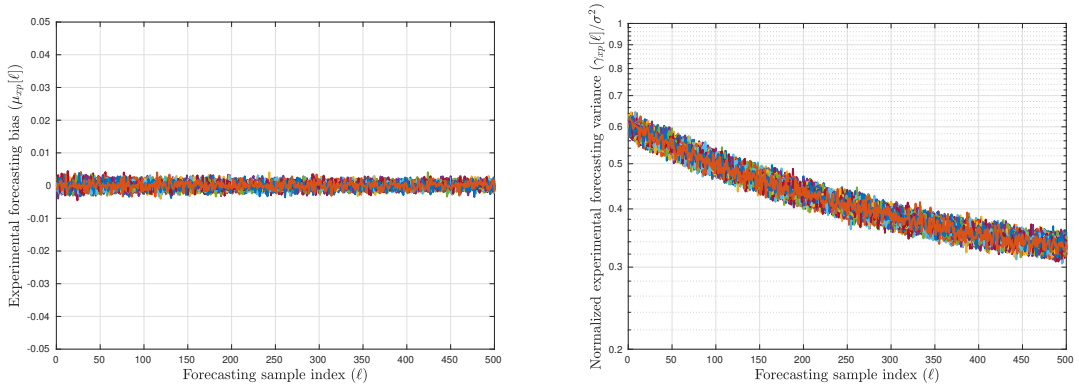

Figure 2: Evolution of the experimental forecasting bias (left) and the normalized forecasting variance (right) in function of the forecasting sample index. Each color characterizes the result for a given value of σ.

Influence of the noise variance σ^2. Here, the size of the training dataset is set to $K = 1875$. Then, the forecasting algorithm is run on 100 realizations on the discrete signal $\mathbf{x}$ for 100 different values of σ^2 logarithmically spaced from 5.10^{-3} to 10^{-1}. For each of these values, we determine the experimental bias $\mu_{xp}[\ell]$ and variance $\gamma_{xp}[\ell, \ell]$ in function of the forecasting sample index ℓ (going from 1 to $L = 500$). These results are displayed on Figure 2. Each color corresponds to these experimental results obtained for a given value of σ. On the left of Figure 2, the result show that the bias is neither depending on the noise variance σ^2 nor the forecasting length ℓ, and always insignificant with respect to the signal values. On the right of Figure 2, we display the normalized experimental variance, that is $\frac{\gamma[\ell,\ell]}{\sigma^2}$. This highligths the fact that the forecasting variance increases linearly with respect to σ^2. More surprisingly, this result shows that the forecasting variance decreases with ℓ, what is counterintuitive.

Influence of the training dataset size K. Here, the noise variance σ is set to $\sigma = 10^{-2}$. Then, the forecasting algorithm is run on 3000 realizations on the discrete signal $\mathbf{x}$ for 25 different values of K, logarithmically spaced from 3.5×10^3 to 6×10^3. For each of these values, we determine the experimental bias $\mu_{xp}[\ell]$ and variance $\gamma_{xp}[\ell, \ell]$ in function of the forecasting sample index ℓ (going from 1 to 500). These results are displayed on Figure 3. Each color corresponds to these experimental results obtained for a given value of K. On the left of Figure 3, as expected, the experimental bias vanishes. On the right of Figure 3, we display the product of K with the normalized experimental variance, that is $\frac{\gamma[\ell,\ell]}{\sigma^2}$. This result shows the asymptotic behavior of the variance. Indeed, as soon as K is sufficiently high, the product $K\gamma[\ell, \ell]$ is approximately independent of K.

Conclusion. Both previous results allows us to describe the influence of the noise variance and the size of the training dataset on the variance of the forecasting noise, which is empirically summarized as follows:

$$\gamma[\ell, \ell] \sim \frac{\sigma^2}{K} g[\ell] \,, \quad \text{when } \sigma^2 \to 0 \text{ and } K \to \infty \,, \tag{21}$$

where g is a bounded positive function.

4.1.2 Adaptive harmonic model

We now consider a signal whose instantaneous frequencies and amplitudes of its components vary over time. The toy signal we use takes the following form:

$$\mathbf{x}[n] = \cos\left(2\pi\phi_1[n]\right) + R[n]\cos\left(2\pi\phi_2[n]\right) \,, \quad \forall\, n \in \{1, \ldots, N\} \,,$$

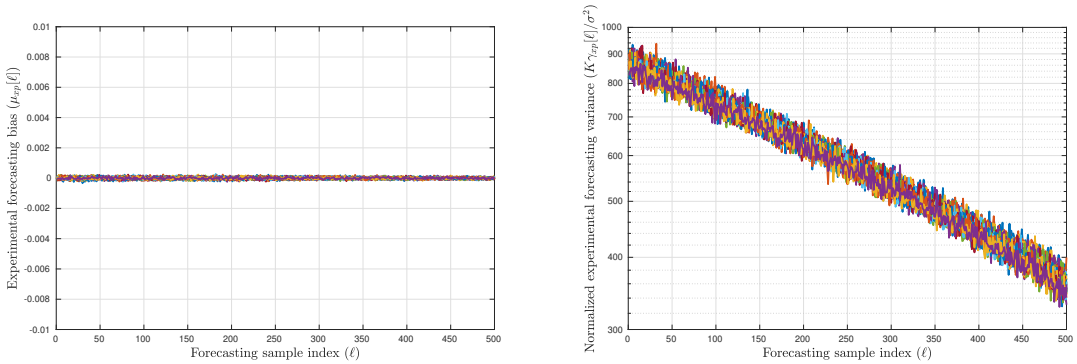

Figure 3: Evolution of the experimental forecasting bias (left) and the normalized forecasting variance (right) in function of the forecasting sample index. Each color characterizes the result for a given value of K.

where the instantaneous amplitude R is given by:

$$R[n] = 1.4 + 0.2\cos\left(4\pi\frac{n}{N}\right) ,$$

and the instantaneous phases are such that:

$$\phi_1[n] = \frac{p_1}{M}\left(n + \frac{0.01}{2\pi}\cos\left(2\pi\frac{n}{N}\right)\right)$$
$$\phi_2[n] = p_2\frac{n}{M} + \frac{20}{2Nf_s}n^2$$

Besides, the additive noise is chosen to be Gaussian: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Numerically, we take: $N = 10^4$, $M = 750$, $p_1 = 10$, $p_2 = 23$ and $R = 1.4$.

To highlight the fact that the linear dynamical model is sufficient to catch most of the dynamical behavior of signals following the AHM, we compare the performance of the Algorithm 2 with reference forecasting algorithm that could be used for extending such signals. These methods are:

- The *Extended Dynamic Mode Decomposition (EDMD)* has been developped by Williams *et al.* [5]. The proposed algorithm is a way to obtain an approximation of the so-called Koopman operator of the observed system, which theoretically allows to catch dynamic of nonlinear systems [2].
- The *Gaussian Process Regression (GPR)* [4] is a method relying on a probabilistic dynamical model. That one is based on the Gaussian process structure, and therefore offer more flexibility in the type of dynamic that could be modeled than the linear model (2).

To quantify the global quality (i.e. not depending on ℓ) of the forecasting approaches, we evaluate the Experimental Mean Square Error $\mathrm{MSE}_{\mathrm{xp}}(\tilde{\mathbf{x}})$ of the forward forecast extended signals, namely:

$$\begin{aligned}\mathrm{MSE}_{\mathrm{xp}}(\tilde{\mathbf{x}}) &= \frac{1}{L}\|\tilde{\mathbf{x}} - \mathbf{x}^{\mathrm{ext}}\|^2 \\ &= \frac{1}{L}\sum_{\ell=1}^{L} \boldsymbol{\mu}_{xp}[\ell]^2 + \boldsymbol{\gamma}_{xp}[\ell,\ell]\,.\end{aligned} \tag{22}$$

where $\mathbf{x}^{\mathrm{ext}}$ is the ground-truth extended signal, that is: $\mathbf{x}^{\mathrm{ext}} = \begin{pmatrix}\mathbf{x}[-L] & \cdots & \mathbf{x}[N-1+L]\end{pmatrix}$. Then, as long as the bias $\boldsymbol{\mu}[\ell]$ and the variance $\boldsymbol{\gamma}[\ell,\ell]$ of the forecasting estimator remain small for all ℓ, the MSE takes small values either. Corresponding results are given in Table 1. They show that the naive

Table 1: Performance of the extension methods.

Algorithm	MSE		Computing time (sec.)
	Mean	Standard deviation	
SigExt	1.433×10^{-3}	4.361×10^{-4}	0.15
EDMD	3.076×10^{-2}	8.095×10^{-2}	2.53
GPR	1.436×10^{-3}	4.346×10^{-4}	146.33

extension we propose gives satisfying results, even though the oher methods, more sophisticated gives MSE values that are somewhat smaller. Netherveless, a major limit of those methods is the computing time they require, which prevent them from being used to exploit real-time data. Thus, SigExt is the method that optimize the trade-off between the forecasting quality and the computing time as extension method. That is why, it is implemented in our algorithm for the reduction of boundary effect.

4.2 Evaluation of the quality of the boundary effect reduction

4.2.1 Metrics

The quality of the boundary effect reduction must be evaluated on the synchrosqueezing representation. To that aim, we compare the obtained synchrosqueezing transform to the optimal synchrosqueezing transform $\mathscr{F}_N^{\text{opt}}(\mathbf{x})$. The optimal synchrosqueezing transform is defined as the restriction of the synchrosqueezing of the ground-truth extended signal $\mathbf{x}^{\text{ext}}$. Therefore, we have:

$$\mathscr{F}_N^{\text{opt}}(\mathbf{x}) = \mathcal{R}\left(\mathscr{F}_{N+2L}(\mathbf{x}^{\text{ext}})\right) .$$

The different representations we obtained must be compared to the optimal SST. To that aim, we establish a criterion that quantify the distance of these time-frequency representations to the optimal one. To do this, we reason in analogy with the optimal transport distance which allows us to quantify the distance between two probability density functions. Let us generically denote a time frequency representation $\mathscr{Q}$. Then, for t fixed, we consider the following pseudo probability density function:

$$p^t_{\mathscr{Q}}(\xi) = \frac{|\mathscr{Q}(\xi, t)|^2}{\int_{\mathbb{R}} |\mathscr{Q}(\nu, t)|^2 \mathrm{d}\nu} . \tag{23}$$

Similarly, let $p^t_{\mathscr{F}_0}$ denotes the associated pseudo-probability density function (defined as in (23)) associated with the reference time-frequency representation $\mathscr{F}_0$. At each instant t, we can then determine the optimal transport distance d_t between the two pseudo densities. It is given by the L^1 norm of the difference between the associated distribution functions. In other words, either $P^t_{\mathscr{Q}}(\xi) = \int_{-\infty}^{\xi} p^t_{\mathscr{Q}}(\nu)\mathrm{d}\nu$ and $\tilde{P}^t_{\mathscr{F}_0}(\xi) = \int_{-\infty}^{\xi} \tilde{p}^t_{\mathscr{F}_0}(\nu)\mathrm{d}\nu$, we have

$$d_t(\mathscr{Q}, \mathscr{F}_0) = \int_{\mathbb{R}} \left| \tilde{P}^t_{\mathscr{Q}}(\xi) - P^t_{\mathscr{F}_0}(\xi) \right| \mathrm{d}\xi .$$

Finally, the distance between the two time-frequency representations is obtained by averaging all the optimal transport distances with respect to time:

$$D(\mathscr{Q}, \mathscr{F}_0) = 100 \times \frac{1}{|I|} \int_I d_t(\mathscr{Q}, \mathscr{F}_0)\mathrm{d}t . \tag{24}$$

The *Optimal Transport Distance* (OTD) quantifies the proximity between the estimated and actual instantaneous frequencies while favouring the sparsity of the estimated time-frequency representation.

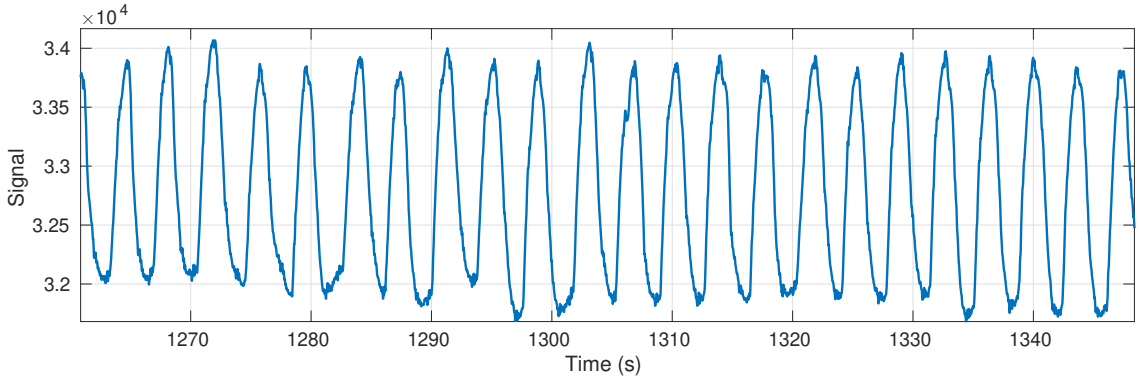

Figure 4: Zoom on the respiratory signal.

Table 2: Averaged MSE.

Algorithm	MSE	
	Mean	Standard deviation
SigExt	0.7935	0.2615
Zero padding	0.9903	0.0579

4.2.2 Respiratory signal

The signal is 50 minutes-long and is sampled at $f_s = 100$ Hz. A zoom on a small portion of the signal is displayed in Figure 4.

From that large signal, we build a dataset of 48 non-overlapping signals of 60 seconds, i.e. $N = 6000$. On each of these pieces of signal, we implement the Algorithm 2 on diverse time-frequency and time-scale representations. The corresponding results are given in Table

The forecasting Algorithm 1 is applied to each of the signals in the dataset we built. These ones are extended of 7 second on each border, corresponding to $L = 700$. Thus, in order to catch slowly varying dynamical behaviors, the size of the training signal M is chosen so that $M = \lfloor 1.5L \rfloor$. As a result of section 4.1.1, we take: $K = \lfloor 2.5M \rfloor$. The average and the standard deviation of the MSE 22 with respect to the simulations are given in Table 2.

The results are compared with the strategy consisting in a zero-padding extension of the signal. These results show a significant improvement of the quality of the forecasting. Indeed, in average, the forecasting MSE is improved of almost 20%. Nevertheless, the variance forecasting MSE shows a huge variability in the quality of the estimation. This is probably due to the outliers and pulse that can be find in the respiratory signal. That ones make the adaptive harmonic model temporary irrelevant, and break the validity of the linear dynamic model.

Even though the performance of the forecasting algorithm is moderate, the boundary effect can be reduced dramatically on the time-frequency and time-scale representations. We evaluate the OTD to the optimal representation for different time-frequency and time-scale representation. Notice that the extension length L is set accordingly to the window length used by the time-frequency analysis tool. For instance, here the window length we use to evaluate the STFT is of 1500 samples. To prevent the STFT from being sensitive to the boundary effect, we set $L = 750$. In this way, when evaluating the spectral content of the signal near its boundaries, the analysis is not limited by a lack of information all along the window support. From now on, all results are given for L at equal to the half of the width of the window used in the time-frequency transform.

4.2.3 Photoplethysmogram

We perform a study similar to the previous one on a 640 second-long photoplethysmogram (PPG) signal, sampled at $f_s = 125$ Hz. A 32 second-long piece of this signal is displayed on the top of

Table 3: Performance of the Boundary effect reduction on TFR.

Extension method	Time-Frequency Representation		
	STFT	SST	RS
Without extension	2.16×10^{-2}	5.26×10^{-3}	3.07×10^{-2}
SigExt	1.72×10^{-2}	4.00×10^{-3}	2.43×10^{-2}
EDMD	1.75×10^{-2}	4.23×10^{-3}	2.45×10^{-2}
GPR	1.84×10^{-2}	4.10×10^{-3}	2.48×10^{-2}

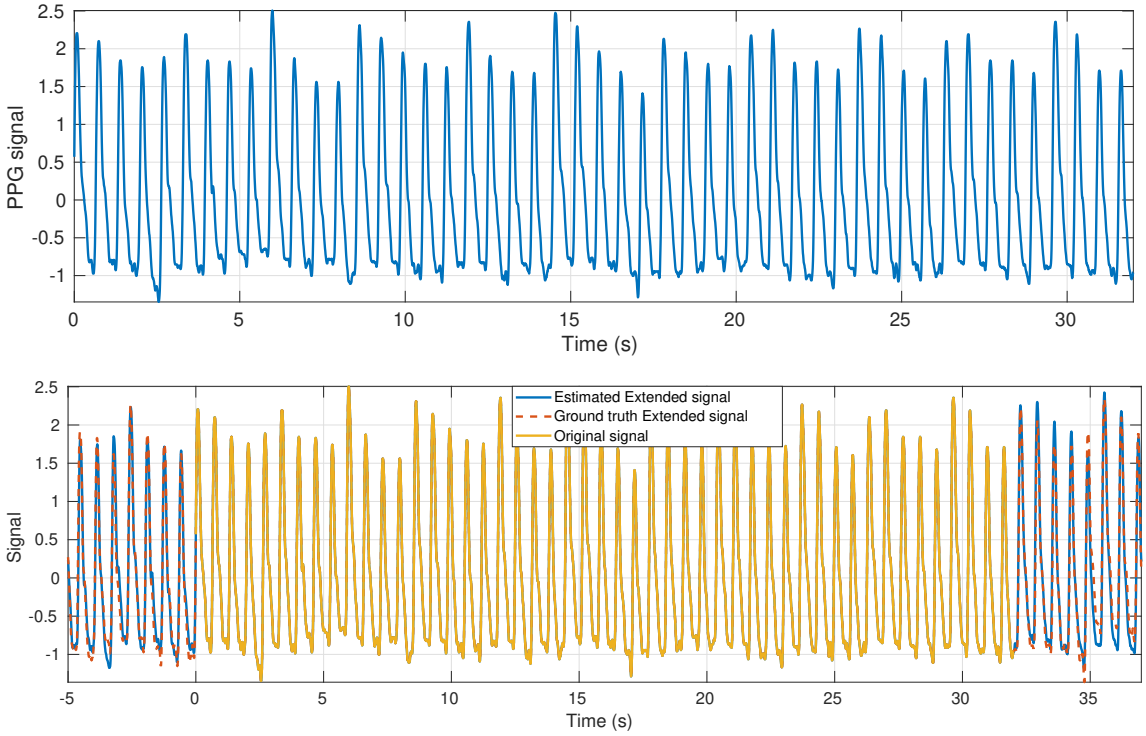

Figure 5: PPG signal. Top: original measured signal. Extended signal obtained by forecasting superimposed with the ground truth signal.

Figure 5. The estimated 2-second extension on both sides of this signal is superimposed to the ground-truth signal is provided on the bottom of Figure 5.

We divide the signal into 32-second long pieces, and apply Algorithm 2 on each piece. We provide in Table 4 the OTD to the optimal time-frequency representation averaged over the signals. For all the considered time-frequency representations, the results clearly shows that our algorithm reduce the influence of the boundary effect. This highlights the ability of our approach to limit the distortion due the boundary effect and provide a more accurate representations.

On Fig. 6, we display the SST resulting from BoundEffRed strategy, applied to the same portion of PPG than what is used to display Fig. 5. We clearly observe an improvement of the quality of the STT near boundaries. Indeed, the blurring visible on the zoom on the right boundary of the SST has almost vanished (see right of Fig. 6). The real-time tracking of the instantaneous frequencies contained the measured signal is therefore largely facilitated.

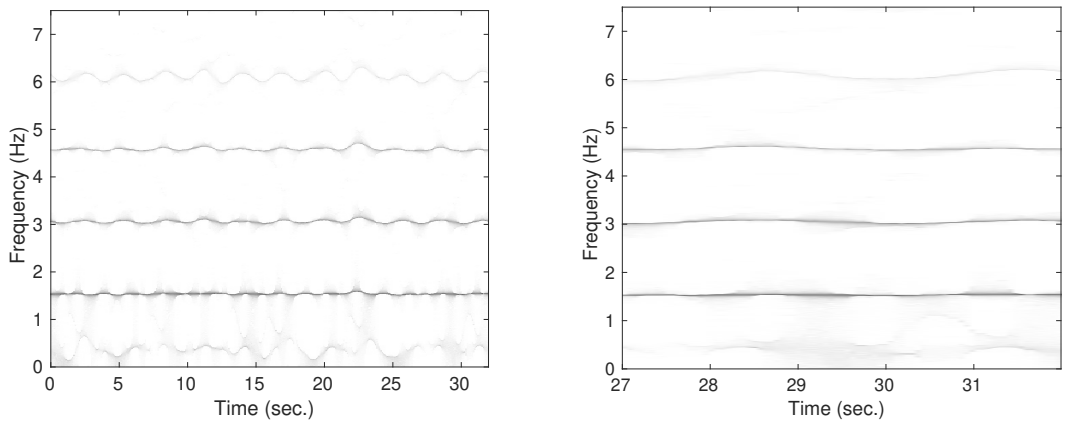

Figure 6: Result of BoundEffRed on the synchrosqueezing transform of a PPG (left) with a zoom on its right boundary (right).

Table 4: Performance of the Boundary effect reduction on TFR.

Extension method	Time-Frequency Representation		
	STFT	SST	RS
Without extension	2.52×10^{-2}	9.41×10^{-2}	1.03×10^{-1}
SigExt	1.22×10^{-3}	7.31×10^{-2}	1.11×10^{-1}
EDMD	9.83×10^{-4}	5.59×10^{-2}	9.80×10^{-2}
GPR	1.14×10^{-3}	7.02×10^{-2}	1.07×10^{-1}

References

[1] François Auger, Patrick Flandrin, Yu-Ting Lin, Steve McLaughlin, Sylvain Meignen, Thomas Oberlin, and Hau-Tieng Wu. Time-frequency reassignment and synchrosqueezing: An overview. *IEEE Signal Processing Magazine*, 30(6):32–41, 2013.

[2] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149 – 160, 2018.

[3] Stéphane Mallat. *A Wavelet Tour of Signal Processing (Second Edition)*. Academic Press, San Diego, 1999.

[4] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. The MIT Press, 2006.

[5] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.