Stabilization and Variations to the Adaptive Local Iterative Filtering Algorithm: the Fast Resampled Iterative Filtering Method

Giovanni Barbarino[1*] and Antonio Cicone[2,3,4†]

[1*]Faculté Polytechnique de Mons , Université de Mons, Rue de Houdain 9, Mons, 7000, Wallonia, Belgium.
[2]Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Via Vetoio 1, Coppito, 67100, L'Aquila, Italy.
[3]INAF - Istituto di Astrofisica e Planetologia Spaziali, via del Fosso del Cavaliere 100, Rome, 00133, Rome, Italy.
[4]Istituto Nazionale di Geofisica e Vulcanologia, Via di Vigna Murata 605, Rome, 00143, Rome, Italy.

*Corresponding author(s). E-mail(s): giovanni.barbarino@gmail.com;
Contributing authors: antonio.cicone@univaq.it;
†These authors contributed equally to this work.

Abstract

Non-stationary signals are ubiquitous in real life. Many techniques have been proposed in the last decades which allow decomposing multi-component signals into simple oscillatory mono-components, like the groundbreaking Empirical Mode Decomposition technique and the Iterative Filtering method. When a signal contains mono-components that have rapid varying instantaneous frequencies like chirps or whistles, it becomes particularly hard for most techniques to properly factor out these components. The Adaptive Local Iterative Filtering technique has recently gained interest in many applied fields of research for being able to deal with non-stationary signals presenting amplitude and frequency modulation. In this work, we address the open question of how to guarantee a priori convergence of this technique, and propose two new algorithms. The first method, called Stable Adaptive Local Iterative Filtering, is a stabilized version of the Adaptive Local Iterative Filtering that we prove to be always convergent. The stability, however, comes at the cost of higher complexity in the calculations. The second technique, called Resampled Iterative Filtering, is a new generalization of the Iterative Filtering method. We prove that Resampled Iterative Filtering is guaranteed to converge a priori for any kind of signal. Furthermore, we show that in the discrete setting its calculations can be drastically accelerated by leveraging on the mathematical properties of the matrices involved. Finally, we present some artificial and real-life examples to show the power and performance of the proposed methods.

Keywords: iterative filtering, adaptive local iterative filtering, empirical mode decomposition, convergence analysis, nonstationary signals, signal decomposition.

MSC Classification: 94A12 , 68W40 , 15A18 , 47B06 , 15B05.

Acknowledgments and Declarations

The authors are members of the Italian “Gruppo Nazionale di Calcolo Scientifico” (GNCS) of the Istituto Nazionale di Alta Matematica “Francesco Severi” (INdAM).

AC thanks the Italian Ministry of the University and Research for the financial support under the PRIN PNRR 2022 grant number E53D23018040001 ERC field PE1 project P2022XME5P titled “Circular Economy from the Mathematics for Signal Processing prospective”, and the Ministry of Foreign Affairs and the International Cooperation for the financial support under the “Grande Rilevanza” Italy – China Science and Technology Cooperation Joint Project titled “sCHans – Solar loading infrared thermography and deep learning teCHniques for the noninvAsive iNSpection of precious artifacts”.

GB is supported by the Alfred Kordelinin säätiö grant no. 210122 and partly by an Academy of Finland grant (Suomen Akatemian päätös 331240). GB is also supported by the European Union (ERC consolidator grant, eLinoR, no 101085607).

All authors contributed to the work in equal parts. All authors read and approved the final manuscript.

1 Introduction

Real life signals are, in general, non-stationary, meaning that their features, including frequencies and amplitudes of their components, are varying over time. We mention, for instance, in Engineering, the analysis and mitigation of multipath-corrupted measurements for the differential Global Navigation Satellite System (GNSS) vehicle navigation which represents a challenge for the usage of GNSS in big cities [1]; in Medicine, the automatic assessment of patients' general physical health from physiological time series like the arterial blood pressure without the assistance of practitioners [2]; in Physics, the study of plasma instabilities [3], the forecast of the ionospheric "Space Weather" to reduce the impact on satellites and telecommunications of solar storms [4], the analysis of earthquakes, and possibly their prediction, via the satellite measurements of the Earth's magnetic field [5], gravitational-wave signals of astrophysical origin, as measured in the Advanced LIGO-Virgo collaboration, which are strongly affected by rapid noise excesses, a.k.a. glitches [6]. The analysis and decomposition of non-stationary signals is an active research direction in both Mathematics and Signal Processing.

In general, the problem can be expressed in mathematical terms as the blind decomposition of a signal $g(x)$ into a combination of non-stationary purely oscillating components, called Intrinsic Mode Functions (IMFs), as in

$$g(x) := \epsilon(x) + \sum_{j=1}^{M} a_j(x) g_j(x), \qquad g_j(x) = \cos(\alpha_j(x))$$

where $\alpha_i'(x) > \delta > 0$ are the instantaneous frequency functions, $\epsilon(x)$ is a low-intensity noise and $a_j(x)$ are bounded and slowly oscillating amplitude functions. Several methods have been developed over the years dealing with different specifications of the above formulation, e.g. for components that are stationary ($\alpha_i'(x)$ constant) or not amplitude-modulated ($a_j(x)$ constant), and so on.

Classical methods based on Fourier and wavelets transform proved to be limited in handling non-stationary signals, especially the ones whose features change quickly over time [7]. In the last decades, many new techniques have been proposed to overcome such limitations. Among them, the Iterative Filtering (IF) algorithm [8] was proposed a decade ago as an alternative technique to the celebrated Empirical Mode Decomposition (EMD) technique [9] and its variants [10–13]. The EMD and its variants, in fact, were missing a rigorous mathematical analysis, due to the usage of a number of heuristic and ad hoc elements. Some results have been presented in the literature [14–16], but a complete rigorous mathematical analysis is still missing nowadays.

The EMD-like methods are based on the iterative computation of the signal moving average via envelopes connecting its extrema. The computation of the signal moving average allows to split the signal itself into a small number of IMFs, which are separated in frequencies and almost uncorrelated [17].

The IF method has been developed following the same structure as EMD but with a key difference: the moving average is now obtained through an iterated convolutional filtering operation on the signal, with the aim to single out all its non-stationary components, starting from the highest frequency one.

The IF algorithm structure allowed the development of a complete mathematical analysis [18–20]. On the other hand, this method is "rigid" in the sense that it allows to extract only IMFs which are amplitude modulated, but almost stationary in frequencies. This is a clear limitation if the signal contains chirps or whistles, which are components with quickly changing instantaneous frequencies. For this reason in [19] the authors proposed a generalization of IF called Adaptive Local IF (ALIF). ALIF does not suffer any more of the rigidity of IF in extracting IMFs containing rapidly varying instantaneous frequencies. However, this new technique looses most of the mathematical background of IF. Even if the algorithm gained visibility since its introduction five years ago, see for instance [21–31], an initial mathematical analysis has been only recently developed [32, 33], and additional research on extensions, variations, and stabilization methods is currently ongoing, as in [34].

Due to the missing theoretical background of the ALIF method, we introduce in this paper two new algorithms for which such analysis is possible. The first, called the Stable ALIF (SALIF) method, is always convergent, even in the presence of noise, but it presents an increased computational cost with respect to ALIF. The second, called the Resampled IF method (RIF), is actually a modification of the IF algorithm preserving its convergence property and at the same time sporting the same flexibility as ALIF. Furthermore, the RIF method can be made, in the discrete case, highly computationally efficient via the FFT computation of the convolutions, in what is called the Fast Resampled IF method (FRIF).

The rest of this paper is structured as follows. Section 2 is a review of the IF method, its properties, and the theory behind it, and actually presents some new results on its stability. Section 3 exhibits the ALIF algorithm, with a discussion on why the method lacks results and a proper theoretical analysis, and introduces the new SALIF method. Here we also compare the features of all three algorithms, stressing their strength and weaknesses. Section 4 is dedicated to the RIF algorithm, its analysis, properties, and acceleration via FFT, in what is called the FRIF technique. In this section, we show how RIF combines the convergence and stability of IF with the flexibility of ALIF, and how it can be made computationally efficient. In Section 5 we compare those algorithms on artificial and real data, reporting the efficiency and accuracy of each method. Eventually, in Section 6, we draw conclusions and suggest future lines of research.

2 Iterative Filtering

Throughout this document, a signal is supposed to be a real function $g : \mathbb{R} \to \mathbb{R}$. The Iterative Filtering (IF) method mimics the EMD algorithm in the application of a moving average that captures the main trend of the signal and allows us to decompose it into simple IMFs. If we call $\mathcal{L}(g)$ the moving average, then both EMD and IF algorithms extract the first IMF as

$$\mathcal{S}(g) = g - \mathcal{L}(g), \qquad \mathrm{IMF}_1 = \lim_{m\to\infty} \mathcal{S}^m(g), \tag{1}$$

where $\mathcal{S}$ is called the 'sifting operator'. The moving average $\mathcal{L}(g)$ is designed in a way that identifies and flattens the most oscillating component of g, so that $\mathcal{S}(g)$ can extract it. The sequence $\mathcal{S}^m(g)$ thus refines the component until convergence to the wanted mode function. It is important to recall here that IF convergence of the limit in (1) has been proved [14, 19], and more details are given in the following section. For EMD, even though many attempts have been made over the years [14–16], this proof is still missing.

Repeating iteratively the same procedure on $r = g - \mathrm{IMF}_1$, we can extract all the IMFs until r becomes a trend signal, meaning that it possesses at most two extrema.

In the EMD method, the moving average operator $\mathcal{L}(g)$ depends completely on the shape of a given signal g, so it changes at each computation of $\mathcal{S}(g)$ inside the algorithm.

In the IF method a fixed filter $k(x)$ is chosen independently from the signal. Here we report the definition of filter.

Definition 1 A filter $k(x)$ is an even, nonnegative, bounded and measurable real function with compact support and unit mass, meaning $\int_{\mathbb{R}} k(z)\ \mathrm{d}z = 1$.

For each IMF, a length L is computed based on the signal, so that the resulting $\mathcal{L}(g)$ can be rewritten as the convolution of g with a scaled version of $k(x)$, i.e.

$$\mathcal{L}(g)(x) = \int_{\mathbb{R}} g(z) k\left(\frac{x-z}{L}\right) \frac{1}{L}\ \mathrm{d}z,$$

We point out that the length L can potentially be recomputed at every iteration m in the sequence $S^m(g)$. However, in IF and its generalizations, for each IMF the length L is computed only in the first iteration and kept unchanged in the subsequent ones.

A generalization of the IF method is called Adaptive Local Iterative Filtering (ALIF), where the moving average utilizes the convolution with a family of filters $k_x(z)$ as in

$$\mathcal{L}(g)(x) = \int_{\mathbb{R}} g(z) k_x(x-z)\ \mathrm{d}z,$$

where $k_x(z)$ has support contained in $[-\ell(x), \ell(x)]$, i.e. it varies with x, and it is derived from a fixed filter $k(z)$. The function $\ell(x)$ will be detailed in Section 3.1.

Following [19], we can rewrite the same expression as

$$\mathcal{L}(g)(x) = \int_{-L}^{L} g(x + t(x,z))k(z)\ \mathrm{d}z, \tag{2}$$

where $t(x,z)$ is a measurable function. The last formulation will be reconsidered in Section 4 and it will be shown to encapsulate also a different algorithm, the Resampled IF method.

The family of filters $k_x(z)$, or equivalently the function $t(x,z)$, is recomputed for each IMF, in order to extract different components of the same signal. In the following sections we report the most common choice for the filters, analyzing their properties and presenting an overview of the resulting methods and the rationale behind them.

2.1 Continuous IF

The IF method separates amplitude modulated IMFs from the signal which are quasi-stationary in frequency, starting from the highest frequencies.

Given a signal $g : \mathbb{R} \to \mathbb{R}$ in $L^2(\mathbb{R})$, the main idea of the method is to use a filter $k(z)$ so that the convolution $k \star g$ smooths out the most oscillating component. Notice that since $k(z)$ is nonnegative, and has unit mass and compact support, the convolution can be seen as a local averaging of the signal. The result is thus subtracted from the signal to capture the fluctuating part, and the iterated sifting on the resulting signal progressively refines the quality of the extracted IMF.

Following (1), we iteratively apply the sifting $\mathcal{S}(\cdot)$ to the signal through convolution with the filter $k_L(z)$ as in

$$\mathcal{L}(g)(x) = \int_{\mathbb{R}} g(z)k_L(x-z)\mathrm{d}z, \qquad \mathcal{S}(g) = g - \mathcal{L}(g), \tag{3}$$

for an opportune length L dependent on g, where $k_L(z) := k(z/L)/L$ and $k(z)$ is a fixed filter. Notice that $k_L(z)$ is still a filter by Definition 1 and by taking the Fourier Transform, we get

$$\widehat{\mathcal{S}(g)} = \widehat{g} - \widehat{\mathcal{L}(g)} = \widehat{g}(1 - \widehat{k_L}) \implies \widehat{\mathcal{S}^m(g)} = \widehat{g}(1 - \widehat{k_L})^m. \tag{4}$$

Once $\mathcal{S}^m(g)$ converges, it is stored as IMF_1, and the process restarts with the remaining signal $g \leftarrow g - \mathrm{IMF}_1$. The method thus extracts IMFs of progressively lower frequencies, until the remaining signal is a trend signal, which corresponds, conventionally, to a signal which contains 2 or less extrema [8].

We report in Algorithm 1 an overview of the IF method. At the end of the title line of Algorithm 1, we report on the left of the equality sign the output of the code, the variable IMFs, the name of the algorithm, IF, and in between parenthesis the inputs, which are g and δ variables. The same applies to all subsequent pseudocode algorithms.

Algorithm 1 (IF Algorithm) IMFs = IF(g, δ)

```
Inputs: $g$ real $L^2$ function, $\delta > 0$ stopping parameter
Output: IMFs is a set of $L^2$ simple oscillatory functions
IMFs = $\{\}$
initialize the remaining signal $r = g$
while the number of extrema of $r$ is $\geq 2$ do
    compute the length function $L$, depending on $r$
    $g_1 = r$
    $m = 1$
    while $\|g_m - g_{m-1}\| > \delta$ do
        $g_{m+1} = g_m - \int_{\mathbb{R}} g_m(z) k\left(\frac{x-z}{L}\right) \frac{\mathrm{d}z}{L}$
        $m = m + 1$
    end while
    IMFs = IMFs $\cup \{g_m\}$
    $r = r - g_m$
end while
```

All the algorithms in this document have the same structure as Algorithm 1, as all methods perform an iterative extraction of IMFs from a signal (referred to as 'outer loop', corresponding to the 'while' condition in line 3) through an iterative sifting process for each individual component (referred to as 'inner loop', corresponding to the 'while' condition in line 7).

The filter $k(z)$ is fixed independently from the signal so that the convergence of the inner loop is assured. This has been proved for filters $k(z)$ with certain conditions on their Fourier transform $\widehat{k}(\xi) := \int_{\mathbb{R}} k(z) e^{-\mathrm{i}2\pi z\xi}\, \mathrm{d}z$.

Theorem 1 *[14, Corollary 3.2] Given a filter $k(z)$, if $|1 - \widehat{k}(\xi)| < 1$ or $\widehat{k}(\xi) = 0$, then $\mathcal{S}^m(g)$ in (3) converges when $m \to \infty$ for any $g(z) \in L^2(\mathbb{R})$ and $L > 0$.*

Since k has compact support, its Fourier transform $\widehat{k}(\xi)$ is an analytic even function over $\mathbb{R}$ with a finite number of zeros in any compact real subset. Additionally, $\widehat{k}(0) = 1 = \|\widehat{k}(z)\|_\infty$, so the condition of Theorem 1 is satisfied, for example, by a double convoluted filter $k = \omega \star \omega$, where ω is a generic filter, since $\widehat{k} = |\widehat{\omega}|^2$.

Call IMF$_j$ the components generated in order by Algorithm 1. If r_j is the remaining signal that is being considered in the j-th step of the outer loop, then IMF$_j = \mathcal{S}^{m_j}(r_j)$, where m_j is the number of iterations of the inner loop, determined by the stopping criterion. Given the tolerance $\delta > 0$, we can rewrite the stopping criterion as

$$\|\mathcal{S}^{m+1}(r_j) - \mathcal{S}^m(r_j)\|_{L^2} < \delta. \tag{5}$$

From Theorem 1 we know that the stopping criterion is always met for a big enough number of iterations, and we can give a bound on m_j and a better description of IMF_j.

Theorem 2 *[20, Theorem 2] Given a double convoluted filter $k(z)$, a fixed tolerance $\delta > 0$, and a signal $g(z) \in L^2(\mathbb{R})$, let $\mathcal{S}(g)$ be the sifting operation in (3) with any $L > 0$. If $m(g)$ is the smallest positive integer such that*

$$\frac{m^m}{(m+1)^{m+1}} < \frac{\delta}{\|g\|_{L^2}}, \tag{6}$$

then $\|\mathcal{S}^{M+1}(g) - \mathcal{S}^M(g)\|_{L^2} < \delta$ for all $M \geq m(g)$. As a consequence the j-th component IMF generated by Algorithm 1 is the inverse Fourier transform of $(1 - \widehat{k}(\xi))^{m_j}\widehat{r_j}(\xi)$ where r_j is the remaining signal that is being considered in the j-th step of the outer loop, and $m_j \leq m(r_j)$ is the number of iterations in the inner loop.

Note that the left hand side of (6) is decreasing in m, and for big m it is approximately m^{-1}. This shows in particular that $m_i = O(\|r_i\|_{L^2}/\delta)$.

Both Theorems 1 and 2 are based on the relation (4). Since $\widehat{k}$ is analytic and $\widehat{k}(0) = 1$, there must exist a minimal positive frequency ξ_0 that is a zero of $\widehat{k}$. From (4) we see that as m goes to infinity, the sifting operator $\mathcal{S}^m$ tends to cancel all frequencies of the signal that are not close to a zero of $\widehat{k_L}$ and in particular all frequencies lower than ξ_0. Since $\widehat{k_L}(\xi) = \widehat{k}(L\xi)$, the first positive zero of $\widehat{k_L}$ is ξ_0/L, so we can compute the length L by solving $\xi_0/L = \xi^*$, where ξ^* represents the highest frequency of the signal. $\mathcal{S}^m$ will thus extract the oscillating components of the signal with frequencies near ξ^*, and obtain the first IMF.

The computation of the filter length L is an important step of the algorithm, since it assures the extraction of the highest frequencies. Following [8], when using the double averaging filter one can choose $L = 2\nu/h$ where ν is a tuning parameter that is usually fixed at 1.6 and h is the number of extreme points of the signal on $[0, 1]$. This is an approximation of the average highest frequency contained in g. Another way is to fix a tolerance $\eta > 0$ and take $L = \xi_0/\xi^*$ where ξ^* is the greatest frequency for which $|\widehat{g}(\xi)| > \eta$. Nonetheless, we can prove that Algorithm 1 produces only a finite number of 'relevant' IMFs with norm greater than a fixed tolerance $\eta > 0$.

Theorem 3 *[20, Theorem 3] Let $g \in L^2(\mathbb{R})$ be a bounded signal with $\widehat{s}$ bounded and supported inside an interval $[b, B]$, $k(z)$ a double convoluted filter, and $\delta > 0$, $\eta > 0$ fixed tolerances. Then Algorithm 1 produces only a finite number of IMFs with $\|IMF\|_\infty > \eta$.*

The theorem does not prove that there are only a limited number of IMFs, but from the proof in [20] one can extract something more that is worth citing.

Proposition 4 *Suppose that the hypotheses of Theorem 3 are met and consider the notations of Theorem 2. Then there exists a scalar $\gamma > 0$ such that if $I_j := \{\xi : (1-\widehat{k_{L_j}}(\xi))^{m_j} > 1-\gamma\}$ and*

$$\widehat{IMF_j}^{TH} = \chi_{I_j}\widehat{r}_j + (1-\widehat{k_{L_j}}(\xi))^{m_j}(1-\chi_{I_j})\widehat{r}_j,$$

then for every generated IMF_j we have $\|IMF_j - IMF_j^{TH}\| \le \eta/2$. As a consequence, if L_j is chosen as ξ_0/ξ^* where ξ^* is the greatest frequency for which $|\widehat{r_j}(\xi)| > \eta/2$, and if we stop the Algorithm 1 when $\|\widehat{r}\|_\infty \le \eta/2$, then the method produces a finite number of IMFs.

In the hypothesis of Theorem 3 we have required that $\widehat{g}$ has compact support, and for any j, $|\widehat{r}_j| \le |\widehat{g}|$ from Theorem 2. This means that in Algorithm 1, the lengths L_j are also bounded in an interval, from which the sets I_j of Proposition 4 have measure greater than a same nonzero constant. As a consequence, the latest result tells us that a finite number of I_j form a partition of the whole support of $\widehat{g}$, and $\widehat{\mathrm{IMF}_j} \approx \widehat{\mathrm{IMF}_j}^{TH} \approx \widehat{g}|_{I_j}$, so the IMFs represent an approximated banded splitting of the frequencies of the signal. Since I_j is a neighbourhood of the highest frequency detected in $\widehat{r}_j$, then IMF_j presents progressively smaller frequencies for increasing j.

Eventually, we can prove a more general version of Theorem 3, if we define the 'relevant' IMFs as those with big L^2 norm.

Proposition 5 *Let $g \in L^2(\mathbb{R})$, $k(z)$ a double convoluted filter, and $\delta > 0$, $\eta > 0$ fixed tolerances. Then Algorithm 1 produces only a finite number of IMFs with L^2 norm greater than η.*

Proof From Theorem 2 we know

$$\widehat{\mathrm{IMF}_j} = (1-\widehat{k_{L_j}})^{m_j}\left[1-(1-\widehat{k_{L_{j-1}}})^{m_{j-1}}\right]\dots\left[1-(1-\widehat{k_{L_1}})^{m_1}\right]\widehat{g},$$

that we rewrite as $\widehat{\mathrm{IMF}_j} = s_j \cdot \widehat{g}$, where s_j is a nonnegative function bounded by 1 and one can prove by induction that

$$\sum_{j=1}^{M} s_j = 1 - \left[1-(1-\widehat{k_{L_M}})^{m_M}\right]\dots\left[1-(1-\widehat{k_{L_1}})^{m_1}\right] \le 1, \quad \forall M \ge 1.$$

Since $0 \le s_j^2 \le s_j \le 1$, one concludes that the same holds for the sum of s_j^2. As a consequence,

$$\begin{aligned}\sum_j \|\mathrm{IMF}_j\|_{L^2}^2 &= \sum_j \|\widehat{\mathrm{IMF}_j}\|_{L^2}^2 = \sum_j \|s_j\widehat{g}\|_{L^2}^2 \\ &= \sum_j \int_0^\infty s_j(x)^2|\widehat{g}(x)|^2 dx \\ &= \int_0^\infty |\widehat{g}(x)|^2 \sum_j s_j(x)^2 dx\end{aligned}$$

$$\leq \int_0^\infty |\widehat{g}(x)|^2 dx = \|\widehat{g}\|_{L^2}^2 = \|g\|_{L^2}^2 < \infty,$$

from which we conclude that the algorithm produces only a finite number of IMFs with L^2 norm greater than a fixed tolerance $\eta > 0$. □

Given a double convoluted filter, we know that the conditions of Theorem 1 are satisfied, and in particular $|1 - \widehat{k_L}(\xi)| \leq 1$, so from (4) we infer that the sifting operator is a linear contraction on the Fourier transform of its argument. This is enough to prove that the inner loop Algorithm 1 behaves well under perturbation. Actually, if the perturbation doesn't change the choices for the lengths and the number of iterations in the inner loops, one can prove that the resulting perturbed IMFs are very close to the original ones.

Corollary 6 *Let $h \in L^2(\mathbb{R})$ be a perturbation of the signal $g \in L^2(\mathbb{R})$. If $k(z)$ is a filter such that either $|1 - \widehat{k}| < 1$ or $\widehat{k} = 0$, then for the sifting operation $\mathcal{S}(g)$ in (3) with any $L > 0$ we have*

$$\|\mathcal{S}^m(g+h) - \mathcal{S}^m(g)\|_{L^2} \leq \|h\|_{L^2}.$$

Let now L_j, m_j, IMF_j be the length, number of inner iteration and IMF generated by Algorithm 1 with input $g(z)$ in the j-th outer loop. If L_j, m_j coincide with those generated by the same algorithm with input $g(z) + h(z)$, and if we call the corresponding IMFs as IMF_j^*, then

$$\sum_j \|IMF_j^* - IMF_j\|_{L^2}^2 \leq \|h\|_{L^2}^2.$$

Proof From (4) and the linearity of the sifting operator,

$$\begin{aligned}\|\mathcal{S}^m(g+h) - \mathcal{S}^m(g)\|_{L^2} = \|\mathcal{S}^m(h)\|_{L^2} = \|\widehat{\mathcal{S}^m(h)}\|_{L^2} \\ = \|(1 - \widehat{k_L})^m \widehat{h}\|_{L^2} \leq \|\widehat{h}\|_{L^2} = \|h\|_{L^2}.\end{aligned}$$

From the proof of Proposition 5, we know that $\widehat{\mathrm{IMF}_j} = s_j \cdot \widehat{g}$ and $\widehat{\mathrm{IMF}_j^*} = s_j \cdot (\widehat{g} + \widehat{h})$, where s_j is a nonnegative function bounded by 1 and $\sum_j s_j \leq 1$. As a consequence,

$$\sum_j |\widehat{\mathrm{IMF}_j} - \widehat{\mathrm{IMF}_j^*}| = \sum_j s_j |\widehat{h}| \leq |\widehat{h}|$$

and thus

$$\begin{aligned}\sum_j \|\mathrm{IMF}_j^* - \mathrm{IMF}_j\|_{L^2}^2 = \sum_j \|\widehat{\mathrm{IMF}_j} - \widehat{\mathrm{IMF}_j^*}\|_{L^2}^2 \\ \leq \left\| \sum_j |\widehat{\mathrm{IMF}_j} - \widehat{\mathrm{IMF}_j^*}| \right\|_{L^2}^2 \leq \|\widehat{h}\|_{L^2}^2 = \|h\|_{L^2}^2.\end{aligned}$$

□

The IF method has also been empirically shown (but never formally proved) to be able to extract amplitude modulated components of the signals, i.e., of the form $s(x) := 2a(x)\cos(\xi_s x + \phi)$ for slow-varying smooth functions $a(x)$. A tentative explanation can be given by looking at its Fourier transform

$$\widehat{s}(z) = [\widehat{2a(x)\cos(}\xi_s x)](z) = \left[\widehat{a}(y) \star (\delta_{\xi_s} + \delta_{-\xi_s})(y)\right](z) = \widehat{a}(z - \xi_s) + \widehat{a}(z + \xi_s)$$

and noticing that $\widehat{a}$ is a fast decaying function. Since the IF algorithm extracts components with frequency in a neighborhood of $\pm\xi_s$, it is thus reasonable that it also captures a good approximation of the whole component $s(z)$. Further studies are needed on this subject, so we postpone its analysis to a future work.

2.2 Discrete IF

In the discrete setting, the IF algorithm has the advantage of a fast implementation based on FFT, in what is called Fast Iterative Filtering (FIF), and an advanced theoretical analysis [18, 20].

In practical applications we always study the signal $g(x)$ on an interval, say $[0, 1]$. Outside this interval, the signal is usually not known, so we have to impose some boundary conditions, discussed for example in [35, 36]. In particular, in [36] the authors show how any signal can be pre-extended and made periodic at the boundaries, for example, by reflecting the signal on both sides and making it decay. Therefore, for simplicity and without losing generality, we will assume that the signals to be decomposed are 1-periodic. For a more detailed discussion on this matter we refer the interested reader to [35, 36].

In a discrete setting, a signal is usually given as a vector of sampled values $\boldsymbol{g} = [g_i]_{i=0,\ldots,n-1}$ where $g_i = g(x_i)$ and $x_i = i/n$ for $i \in \mathbb{Z}$. As a consequence, one can discretize the IF moving average (3) with a simple quadrature formula

$$\mathcal{L}(g)(x_i) = \int_{\mathbb{R}} g(z) k_L(x_i - z)\ \mathrm{d}z \approx \frac{1}{nM} \sum_{j\in\mathbb{Z}} g_j k_L(x_i - x_j). \tag{7}$$

Since the filter $k_L(z)$ has compact support, the above formula is always well defined. Here M is a normalizing constant depending on $k_L(z)$ and n defined as

$$M := \frac{1}{n} \sum_{j\in\mathbb{Z}} k_L(x_i - x_j) \approx \int_{\mathbb{R}} k_L(z)\ \mathrm{d}z = 1$$

ensuring that the quadrature formula actually performs a local convex combination of the signal points g_i, akin to the averaging operation performed by the convolution in the continuous case.

We have previously seen that L is inversely proportional to the target frequency of the extracted IMF, and if we fix $k(z)$ to be a filter with $\xi_0 = 1$ as first positive zero of its Fourier transform, then $1/L \leq 2\pi$ usually indicates that we already have a slowly oscillating signal g with two or less extrema, i.e., a trend signal. From now on, both in the arguments and in results, we always suppose that $1/L > 2\pi$. As a consequence, (7) can be expressed through a

Hermitian circulant matrix K with first row

$$\boldsymbol{k}_1 := \frac{1}{nLM}\left[k(0),\, k\left(\frac{1}{nL}\right), \dots,\, k\left(\frac{s}{nL}\right),\, 0 \dots 0,\, k\left(\frac{s}{nL}\right), \dots,\, k\left(\frac{1}{nL}\right)\right]$$

where $s = \lfloor nL \rfloor < \lfloor n/6 \rfloor$. The sampling vector of $\mathcal{S}(g)$ on the points $x_0, x_1, \dots, x_{n-1}$, that we indicate as $\mathcal{S}(\boldsymbol{g})$, is thus rewritten as a matrix-vector multiplication

$$\mathcal{S}(g)(x_i) = g_i - \mathcal{L}(g)(x_i) \implies \mathcal{S}(\boldsymbol{g}) = (I - K)\boldsymbol{g}. \tag{8}$$

We can see in Algorithm 2 a full overview of the method.

Algorithm 2 (Discrete IF Algorithm) $\boldsymbol{IMFs} = \mathrm{dIF}(\boldsymbol{g}, \delta)$

```
**Inputs:** $\boldsymbol{g} \in \mathbb{R}^n$ discretized signal, $\delta > 0$ stopping parameter
**Output:** IMFs is a set of discretized simple oscillatory components in $\mathbb{R}^n$
$\boldsymbol{IMFs} = \{\}$
initialize the remaining signal $\boldsymbol{r} = \boldsymbol{g}$
while the number of extrema of $\boldsymbol{r}$ is $\geq 2$ do
    compute $L$ and the matrix $K$
    $\boldsymbol{g}_1 = \boldsymbol{r}$
    $m = 1$
    while $\|\boldsymbol{g}_m - \boldsymbol{g}_{m-1}\| > \delta$ do
        $\boldsymbol{g}_{m+1} = (I - K)\boldsymbol{g}_m$
        $m = m + 1$
    end while
    $\boldsymbol{IMFs} = \boldsymbol{IMFs} \cup \{\boldsymbol{g}_m\}$
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{g}_m$
end while
```

When the filter k is double convoluted, the matrix K can actually be proved to acquire many crucial properties.

Proposition 7 ([20, Corollary 1, Corollary 3]) *Given a double-convoluted filter k, then the IF matrix K in (8) and $I - K$ are both Hermitian, circulant and positive semidefinite matrices with spectrum belonging to the interval $[0, 1]$. In particular, the limit*

$$\lim_{m\to\infty} (I - K)^m \boldsymbol{g}$$

converges for any vector $\boldsymbol{g}$.

In particular, this ensures the convergence of the sequence $\boldsymbol{g}_m$ generated by the inner loop of Algorithm 2, since $\boldsymbol{g}_m = (I - K)^m \boldsymbol{g}_1$.

Analogously to the continuous case, the stopping criterion of the inner loop is set as the step in which $\|\boldsymbol{g}_{m+1} - \boldsymbol{g}_m\|$ falls below a certain tolerance $\delta > 0$

and we can provide a bound on the number of iterations of the corresponding inner loop. For alternative stopping criteria see for example [8].

Theorem 8 *Given a fixed tolerance $\delta > 0$, and a discrete signal $\boldsymbol{g}$, let $\mathcal{S}(\boldsymbol{g})$ be the sifting operation in (8) where K is any $n \times n$ Hermitian matrix with spectrum in $[0,1]$ and eigendecomposition $K = UDU^T$. Suppose $m(\boldsymbol{g})$ is the smallest positive integer m such that*

$$\frac{m^m}{(m+1)^{m+1}} < \max\left\{\frac{\delta}{\|\boldsymbol{g}\|}, \frac{\delta}{\sqrt{n-1-p}\|U^T\boldsymbol{g}\|_\infty}\right\}, \tag{9}$$

where p is the number of eigenvalues 1 in K. Then $\|\mathcal{S}^{M+1}(\boldsymbol{g}) - \mathcal{S}^M(\boldsymbol{g})\| < \delta$ for all $M \geq m(\boldsymbol{g})$.

As a consequence, if we consider a double convoluted filter, then the j-th IMF generated by Algorithm 2 is $(I-K)^{m_j}\boldsymbol{r}_j$ where $\boldsymbol{r}_j$ is the remaining signal that is being considered in the j-th step of the outer loop, and $m_j \leq m(\boldsymbol{r}_j)$ is the number of iterations in the inner loop.

Proof The bound on the right hand side of (9) with $\|U^T\boldsymbol{g}\|_\infty$ is proved in [20, Theorem 5]. From the hypothesis, $I - K$ is Hermitian and its spectrum is contained in $[0,1]$. Since the real function $f(\lambda) := (1-\lambda)^m\lambda$ has maximum on $[0,1]$ given by $\lambda = (m+1)^{-1}$, we get

$$\|\mathcal{S}^m(\boldsymbol{g}) - \mathcal{S}^{m+1}(\boldsymbol{g})\| = \|(I-K)^m K\boldsymbol{v}\| = \|U(I-D)^m DU^T\boldsymbol{g}\| \leq \frac{m^m}{(m+1)^{m+1}}\|\boldsymbol{g}\|.$$

The relation (9) then follows from the observation that its left hand side is decreasing in m. From Proposition 7 if $k(z)$ is a double convoluted filter, then in Algorithm 2 the matrix K is Hermitian and its spectrum is contained in $[0,1]$. □

The IF method is fast since K is a circulant matrix, thus the multiplication $(I-K)\boldsymbol{g}_m$ can be performed very efficiently through an FFT. Actually, in [20] we can find an even faster implementation, the so called FIF (Fast Iterative Filtering) algorithm, based on the relation

$$\mathrm{DFT}(\boldsymbol{g}_{m+1}) = (1 - \mathrm{DFT}(\boldsymbol{k}_1))^{\circ m} \circ \mathrm{DFT}(\boldsymbol{g}_1),$$

where $\circ$ stands for the Hadamard (or element-wise) product between vectors, DFT and iDFT stand for Discrete Fourier Transform and its inverse, and $\boldsymbol{k}_1$ is the first row of K. Since the stopping criterion

$$\|\boldsymbol{g}_{m+1} - \boldsymbol{g}_m\|_2 = \|\mathrm{DFT}(\boldsymbol{g}_{m+1}) - \mathrm{DFT}(\boldsymbol{g}_m)\|_2 \leq \delta$$

can be checked on $\mathrm{DFT}(\boldsymbol{g}_m)$, we can further accelerate the method by computing the DFTs of $\boldsymbol{g}_1$ and $\boldsymbol{k}_1$ and $\boldsymbol{g}_m = \mathrm{iDFT}(\mathrm{DFT}(\boldsymbol{g}_m))$ outside the inner loop, thus avoiding iterated computations of Fourier transforms. We report here the natural consequence of the latter arguments.

Proposition 9 *[20, Corollary 4] In Algorithm 2 the j-th extracted IMF is*

$$iDFT\left((1 - DFT(\boldsymbol{k}_j))^{\circ m_j} \circ DFT(\boldsymbol{r}_j)\right)$$

where $\boldsymbol{r}_j$ is the remaining signal that is being considered in the j-th step of the outer loop, m_j is the number of iterations in the inner loop and $\boldsymbol{k}_j$ is the first row of the matrix K generated in the j-th step of the outer loop.

A full overview of the FIF method will be given later on, when we will generalize it into the Fast Resampled IF (FRIF). As a consequence of the last result, we can again prove that the number of generated 'relevant' IMF is finite.

Proposition 10 *Let $\boldsymbol{g}$ be a discrete signal, $k(z)$ a double convoluted filter, and $\delta > 0$, $\eta > 0$ fixed tolerances. Then Algorithm 1 produces only a finite number of IMFs with norm greater than η.*

The proof uses that DFT$(\boldsymbol{k}_1)$ is the vector of eigenvalues for K and Proposition 7, but the resulting argument is totally analogous to the continuous counterpart Proposition 5, so we omit it. For the same reasons, we report here the discrete version of Corollary 6 without proof. In particular it shows that even the discrete version of IF behaves well under small perturbations of the signal.

Corollary 11 *Let $\boldsymbol{h}$ be a perturbation of the signal $\boldsymbol{g}$, and let $\mathcal{S}(\boldsymbol{g})$ be as in (8) where K is any $n \times n$ Hermitian matrix with spectrum in $[0, 1]$. Then*

$$\|\mathcal{S}^m(\boldsymbol{g} + \boldsymbol{h}) - \mathcal{S}^m(\boldsymbol{g})\| \leq \|\boldsymbol{h}\|.$$

Let now L_j, m_j, $\boldsymbol{IMF}_j$ be the length, number of inner iteration and IMF generated by the j-th outer loop of Algorithm 2 with input $\boldsymbol{g}$, $\delta > 0$ and a double convoluted filter $k(z)$. If L_j, m_j coincide with those generated by the same algorithm with input $\boldsymbol{g} + \boldsymbol{h}$, and if we call $\boldsymbol{IMF}_j^*$ the corresponding IMFs, then

$$\sum_j \|\boldsymbol{IMF}_j^* - \boldsymbol{IMF}_j\|^2 \leq \|\boldsymbol{h}\|^2.$$

Proposition 9 tells us that the method tends to isolate the components of $\boldsymbol{g}$ with frequencies near the zeros of DFT$(\boldsymbol{k}_1)$, that correspond to the zero eigenvalues of K. When the number of sampling points n is big, the DFT of the samples approximate the FFT of continuous functions well enough, letting us to follow the same paradigm of continuous IF for the choice of the length L. In particular, we want again that $\xi_0 = L\xi^*$ where ξ_0 is the smallest positive zero of $\widehat{k}(\xi)$ and ξ^* is the highest relevant frequency of DFT$(\boldsymbol{g})$ (notice that we may not have access to the continuous signal, but only to its sampling). Alternatively, L can be empirically estimated from the number h of extreme points in $\boldsymbol{g}$ and computed as $L = 2\lfloor n\nu/h\rfloor/n$ where ν is a tuning parameter that is usually fixed at 1.6.

3 ALIF and Variations

The IF method has been shown efficient and robust in extracting components that are approximately stationary in frequency, but it comes short when dealing with frequency modulations. Suppose that $g : \mathbb{R} \to \mathbb{R}$ is a signal in $L^2(\mathbb{R})$ composed by a finite number of components as in

$$g(x) = e(x) + \sum_{j=1}^{s} a_j(x) \cos(\alpha_j(x)) \tag{10}$$

where $e(x)$ is a low-norm noise signal, the amplitudes $a_j(x)$ and the instantaneous frequencies $f_j(x) := \alpha'_j(x)$ (see [19]) are continuous real nonnegative functions. Moreover suppose that $f_j(x)$ is decreasing in j for any fixed $x \in \mathbb{R}$. In this case, the IF algorithm would detect a frequency ξ^* near the maximum value of $f_1(x)$, and would end up extracting only a portion of the relative component. The focus on ξ^* is determined by the length L of the filter, so here we would need a different length for each different point $x \in \mathbb{R}$ in some way linked to the function $f_1(x)$. In physical terms, since the dominant frequency of the signal $g(x)$ changes depending on x, we need to adapt the moving average locally to the signal.

This reasoning has led to the formulation in [19] of the Adapted Local Iterative Filtering (ALIF) method, in which the signal is convoluted with an opportune filter that can change depending on the point and that emulates an adapted moving average. In other words, we need a family of filters $k_x(z)$ so that the moving average of the signal $g(x)$ is

$$\mathcal{L}(g)(x) = \int_0^1 g(z) k_x(x - z) \, \mathrm{d}z,$$

where ideally the length $\ell(x)$ of $k_x(z)$ depends on the greatest instantaneous frequency of $g(x)$, represented by $f_1(x)$. Now we describe the algorithm resulting from the most common choice for the filter family $k_x(z)$.

3.1 Linear Adaptive Local IF

When we talk about the ALIF method, we usually refer to Linear ALIF. Given now a signal $g(z)$, let us fix a filter $k(z)$ with support $[-1, 1]$ and nonnegative Fourier transform $\widehat{k}(\xi)$ with ξ_0 as smallest positive zero. Analogously to IF, if we know the highest instantaneous frequency $f(x)$ of $g(z)$ (i.e., $f_1(x)$ in (10)), then we can compute the positive length function $\ell(x) := \xi_0/f(x)$ and form the filter family $k_x(z)$ as

$$k_x(z) := k\left(\frac{z}{\ell(x)}\right) \frac{1}{\ell(x)},$$

following the same kind of stretching already performed in IF for $k_L(z)$. Notice that now $k_x(z)$ is a filter with support $[-\ell(x), \ell(x)]$ for every $x \in [0,1]$. The resulting sifting operation will thus be

$$\mathcal{S}(g)(x) = g(x) - \int_{\mathbb{R}} g(z) k\left(\frac{x-z}{\ell(x)}\right) \frac{\mathrm{d}z}{\ell(x)} \tag{11}$$

The ALIF algorithm then follows the same steps as IF (Algorithm 1). After a length function $\ell(x)$ is computed from the signal, the sifting operation (11) is then applied to $g(x)$ iteratively until convergence to an IMF, where the convergence is determined by a stopping criterion, usually based on the norm of the difference $g_{m+1} - g_m$ or the number of iterations themselves. The IMFs are thus iteratively extracted by repeating the same procedure on the remaining signal $g(x) - \text{IMF}$ until it becomes a trend signal. In Algorithm 3 we report an overview of the method. The operation $\mathcal{S}(g) = g - \mathcal{L}(g)$ is designed to catch the fluctuation part of the signal locally presenting the highest frequency, as indicated by the length function. The identification of $\ell(x)$ in signals containing noise is always possible through a preliminary time-frequency representation (TFR) algorithm and then using the acquired information to design the optimal $\ell(x)$ (see [37] for a comprehensive review of modern TFR techniques). This procedure is really important for ALIF algorithm, but it is also a research topic per se. This is why, from now on, we assume that the length function can be computed accurately and we postpone the analysis of how actually compute it to a future work.

Algorithm 3 (ALIF Algorithm) IMFs = ALIF(g, δ)

```
Inputs: $g$ real $L^2$ function, $\delta > 0$ stopping parameter
Output: IMFs is a set of $L^2$ simple oscillatory functions
IMFs = {}
initialize the remaining signal $r = g$
while the number of extrema of $r$ is $\geq 2$ do
    compute the length function $\ell(x)$, depending on $r$
    $g_1 = r$
    $m = 1$
    while $\|g_m - g_{m-1}\| > \delta$ do
        $g_{m+1} = g_m - \int_{\mathbb{R}} g_m(z) k\left(\frac{x-z}{\ell(x)}\right) \frac{\mathrm{d}z}{\ell(x)}$
        $m = m + 1$
    end while
    IMFs = IMFs $\cup \{g_m\}$
    $r = r - g_m$
end while
```

Conceptually, the ALIF method separates non-stationary components of the signal starting from the highest frequencies. For example, on real data, the method first extract high frequency noise IMFs, and then starts to produce clean components. The method, albeit being very powerful and having already been utilized in a variety of applications, still lacks a theoretical analysis proving the convergence of the inner loop $\lim_{m\to\infty}\mathcal{S}^m(g)$, except for a few notorious cases [19, 32, 34]. Moreover, the method is also missing a rigorous study of the produced IMFs. Their quantity or quality cannot be predicted a priori at this point. The local adaptation of the filter through the length function $\ell(x)$ makes the nice analysis of IF impossible to reproduce in this context. Nevertheless, some tentative studies have been conducted when dealing with discretized signals.

In the next sections, we report some of the available convergence results for the discrete version of the algorithm.

3.2 Discrete ALIF

Using the same notations as Section 2.2, let $\boldsymbol{g} = [g_0\ g_1\ \dots\ g_{n-1}]$ be the sampling $g_i = g(x_i)$ on the points $x_i = i/n$, and suppose we have no other information on the continuous signal $g(x)$. In particular, we do not know $g(x)$ outside the interval $[0,1]$, but we can always suppose through an opportune extension that the function is continuous and supported on this interval (see [35, 36]). As a consequence, one can discretize the relation (11) with a simple quadrature formula

$$\mathcal{S}(g)(x_i) = g(x_i) - \int_0^1 g(z)k_{x_i}(x_i - z)\ \mathrm{d}z \approx g_i - \frac{1}{M_i n}\sum_{j=0}^{n-1} g_j k_{x_i}(x_i - x_j),$$

where M_i is the normalizing constant

$$M_i = \max\left\{1, \frac{1}{n}\sum_{j=0}^{n-1} k_{x_i}(x_i - x_j)\right\} \approx \max\left\{1, \int_{x_i-1}^{x_i} k_{x_i}(z)\ \mathrm{d}z\right\} = 1$$

ensuring that the moving average is discretized into a sub-convex combination (recall that the signal is supposed to be 0 outside $[0,1]$). In turn, this lets us write the sampling vector of $\mathcal{S}(g)$ as a matrix-vector multiplication. In fact, if we assume all the indexes start from zero,

$$\mathcal{S}(\boldsymbol{g}) := [\mathcal{S}(g)(x_i)]_{i=0,\dots,n-1} = (I - K)\boldsymbol{g}, \tag{12}$$

$$K_{i,j} = \left[\frac{1}{M_i n} k_{x_i}(x_i - x_j)\right]_{i,j=0}^{n-1},$$

where K is, by construction, a nonnegative matrix, whose rows add up to 1 since each of them represents the discretization of a filter function k_{x_i}. This is also the reason why K is a row stochastic matrix. In the Linear ALIF paradigm, we fix a filter $k(z)$ (usually, double convoluted) and compute a length function

$\ell(x)$ that usually depends on the relative positions of local extrema in $g(z)$ if the signal does not contain noise, or through more complex TFR algorithms. We thus produce our family of filters $k_x(z) = k(z/\ell(x))/\ell(x)$ and iteratively perform the discrete moving average and sifting operation as shown in (12). The resulting method is reported in Algorithm 4.

Algorithm 4 (Discrete ALIF Algorithm) $\boldsymbol{IMFs} = \text{dALIF}(\boldsymbol{g}, \delta)$

```
**Inputs:** $g \in \mathbb{R}^n$ discretized signal, $\delta > 0$ stopping parameter
**Output:** IMFs is a set of discretized simple oscillatory components in $\mathbb{R}^n$
$\boldsymbol{IMFs} = \{\}$
initialize the remaining signal $\boldsymbol{r} = \boldsymbol{g}$
**while** the number of extrema of $\boldsymbol{r}$ is $\geq 2$ **do**
    compute $\ell(x)$ and the matrix $K$
    $\boldsymbol{g}_1 = \boldsymbol{r}$
    $m = 1$
    **while** $\|\boldsymbol{g}_m - \boldsymbol{g}_{m-1}\| > \delta$ **do**
        $\boldsymbol{g}_{m+1} = (I - K)\boldsymbol{g}_m$
        $m = m + 1$
    **end while**
    $\boldsymbol{IMFs} = \boldsymbol{IMFs} \cup \{\boldsymbol{g}_m\}$
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{g}_m$
**end while**
```

Notice that in the inner loop of the algorithm we again impose a stopping condition, usually for when the norm of $\boldsymbol{g}_{m+1} - \boldsymbol{g}_m$ gets too small. Even so, it is evident that the convergence of the internal loop is not guaranteed, since it depends on the spectral properties of the matrix $I - K$. Since

$$\boldsymbol{g}_{m+1} = \mathcal{S}^m(\boldsymbol{g}_1) = (I - K)^m \boldsymbol{g}_1,$$

we find that a necessary condition for the convergence with any initial $\boldsymbol{g}_1$ is

$$|1 - \lambda_i(K)| \leq 1, \qquad \forall i = 1, \ldots, n. \tag{13}$$

If the zero eigenvalue of K has equal geometric and algebraic multiplicities, and it is the only eigenvalue for which $|1-\lambda_i(K)| = 1$, then the condition is also sufficient. Recent studies [32, 34] show that for large n and continuous functions $k(z)$, $\ell(x)$, almost all eigenvalues of the matrix K belong or are close to the interval $[0, 1]$, but this is still not enough to establish the convergence of the method. In fact, it has been ascertained experimentally (see for example [34]) that the relation (13) may not hold, especially with a fast changing function $\ell(x)$. The most common problem seems to lie in the presence of eigenvalues in K with negative real part.

Despite all the drawbacks, the method has empirically shown to be robust, producing significant IMFs and managing to decompose the signals made from both artificial and real data, in a successful way.

In the next section, we propose a modification to the matrix K of ALIF that ensures the convergence of the inner loop, and that is designed to produce similar IMFs.

3.3 Stable ALIF

Suppose now that the first inner loop of discrete ALIF (Algorithm 4) converges to $\boldsymbol{IMF}_1 = \mathcal{S}^{m_1}(\boldsymbol{g}) = (I-K)^{m_1}(\boldsymbol{g})$, where m_1 is the first index for which the stopping criterion is satisfied. If the initial $\boldsymbol{g}$ had more than one distinct component, then m_1 must be big enough in order to identify the correct frequency. In this case, the algorithm must produce an approximated projection of the signal $\boldsymbol{g}$ on the null space of K, including also smaller components from eigenspaces close to the null space.

As stated before, for the inner loop of the ALIF to converge, we need that $I-K$ has eigenvalues in the unit open complex ball, or equal to 1, that is K must at least have eigenvalues with nonnegative real part. Since this is not always attained, we instead opt to substitute K with a different matrix. In what follows, the norms $\|\cdot\|$, $\|\cdot\|_1$, $\|\cdot\|_\infty$, are respectively the induced 2, 1 and ∞ norm on the matrices.

Proposition 12 *Given a $n\times n$ complex matrix $K\neq 0$, let $A = c^2K^TK$ with $\|cK\|\leq 1$ and $c\neq 0$ a real scalar. Then A is a positive semidefinite matrix with spectrum inside $[0,1]$ and $\ker(A)=\ker(K)$. Moreover, if $\lambda_n(M)$ has the smallest absolute value among the eigenvalues of M then*

$$\lambda_n(A)\leq c^2|\lambda_n(K)|^2.$$

In particular if K is row stochastic, the proposition holds with $c^{-1} = \max\{1,\sqrt{\|K\|_1}\}$.

Proof The matrix c^2K^TK is positive semidefinite and $1\geq \|cK\|^2 = c^2\|K\|^2 = c^2\|K^TK\|$, so its spectrum is contained in $[0,1]$. Moreover, $\ker(K)\subseteq\ker(MK)$ for any matrix M, but the singular values of K^TK are the square of those in K (it can be proved using the SVD of K), so their kernels have the same dimension. Since $c\neq 0$, we conclude that $\ker(c^2K^TK)=\ker(K)$.

In positive semidefinite matrices, the singular values are equal to the eigenvalues, so $\lambda_n(A) = \sigma_n(c^2K^TK) = c^2\sigma_n(K)^2$ where $\sigma_n(M)$ is the smallest singular value of M. The relation $\sigma_n(K)^2\leq|\lambda_n(K)|^2$ is a classic corollary of the equivalent definition $\sigma_n(K)^2 = \min_{\|y\|=1} y^*K^*Ky$.

Suppose now that $c^{-1} = \max\{1,\sqrt{\|K\|_1}\}$. From the classic inequality $\|M\|^2\leq\|M\|_1\|M\|_\infty$, and since K is row stochastic, we find that

$$0 < c\|K\| \leq c\sqrt{\|K\|_1}\leq 1.$$

□

From this result we can see that $A := c^2K^TK$ still presents exactly the same zero eigenvalues of K, so we are able to extract the relative components from the signal by considering the sifting operation $\mathcal{S}(\boldsymbol{g}) = (I - c^2K^TK)\boldsymbol{g}$. Moreover, Theorem 8 proves that the iterated sifting $\mathcal{S}^m(\boldsymbol{g})$ always converges and gives a bound on the number of iterations needed for getting $\|\mathcal{S}^m(\boldsymbol{g}) - \mathcal{S}^{m+1}(\boldsymbol{g})\|$ lower than a fixed tolerance $\delta > 0$.

Corollary 13 *If $A = c^2K^TK$ respects the hypotheses of Proposition 12 and $\mathcal{S}(\boldsymbol{v}) := (I - A)\boldsymbol{v}$, then $\lim_{m\to\infty} \mathcal{S}^m(\boldsymbol{v})$ converges for every $\boldsymbol{v}$. Moreover, given $\delta > 0$ there exists a smallest m_0 for which*

$$\|\mathcal{S}^m(\boldsymbol{v}) - \mathcal{S}^{m+1}(\boldsymbol{v})\| \leq \delta, \quad \forall m \geq m_0$$

and m_0 is upper bounded by the minimum m for which

$$\frac{m^m}{(m+1)^{m+1}} < \max\left\{\frac{\delta}{\|\boldsymbol{v}\|}, \frac{\delta}{\sqrt{n-1-p}\|U^T\boldsymbol{v}\|_\infty}\right\},$$

where p is the dimension of the eigenspace of A relative to the eigenvalue 1, and $A = UDU^T$ is an eigendecomposition of the matrix A.

We call the resulting method Stable ALIF (SALIF), and report the full method in Algorithm 5.

Algorithm 5 (Stable Discrete ALIF Algorithm) $\boldsymbol{IMFs} = \mathrm{SALIF}(\boldsymbol{g}, \delta)$

```
Inputs: $\boldsymbol{g} \in \mathbb{R}^n$ discretized signal, $\delta > 0$ stopping parameter
Output: IMFs is a set of discretized simple oscillatory components in $\mathbb{R}^n$
$\boldsymbol{IMFs} = \{\}$
initialize the remaining signal $\boldsymbol{r} = \boldsymbol{g}$
while the number of extrema of $\boldsymbol{r}$ is $\geq 2$ do
    compute $\ell(x)$, the matrix $K$ and the scalar $c$
    $\boldsymbol{g}_1 = \boldsymbol{r}$
    $m = 1$
    while $\|\boldsymbol{g}_m - \boldsymbol{g}_{m-1}\| > \delta$ do
        $\boldsymbol{g}_{m+1} = (I - c^2K^TK)\boldsymbol{g}_m$
        $m = m + 1$
    end while
    $\boldsymbol{IMFs} = \boldsymbol{IMFs} \cup \{\boldsymbol{g}_m\}$
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{g}_m$
end while
```

Even though the method lets us extract the components relative to the kernel of K, at the same time, we lose control over the components relative to eigenspaces near the null one. In [32, 34], it has been shown that under special conditions on $\ell(x)$ and big enough n, the matrix K is close to a Hermitian matrix $\widetilde{K}$. As already stated, this is not enough to ensure the convergence of the method, and the estimated difference $K - \widetilde{K}$ is too large to bring forth a formal justification of the method, but it gives some insight on Algorithm 5.

From Proposition 12, we see that the order of the smallest eigenvalue of $K^T K$ is approximately the square of the smallest one in K. If K were a Hermitian matrix, this same relation would hold for every eigenvalue, thus raising the number of components near a fixed zero eigenvalue. As a consequence, the algorithm requires more iterations to attain the same accuracy as ALIF, since it must separate eigenspaces that are now closer. In particular, it takes a bigger index m to get $\lambda_i((I - A)^m) = (1 - \lambda_i(A))^m$ close to zero since $\lambda_i(A) \sim \lambda_i(K)^2 \ll \lambda_i(K)$ when $\lambda_i(K)$ is already small.

Experimental evidences support this theory, showing how SALIF takes more iterations than ALIF to converge to a clear component. Moreover, the iterative step in the SALIF algorithm $\boldsymbol{g}_{m+1} = (I - K^T K)\boldsymbol{g}_m$ takes at least double the number of flops with respect to the respective step in the ALIF algorithm, and since the number of iterations is usually much smaller than n, computing $K^T K$ or its eigendecomposition beforehand does not in general improve the speed of the method. As we will show in the experiments, ref. Section 5, the stability of SALIF is probably the main factor leading to more accurate component extraction than ALIF, at the cost of an increased computational cost.

The SALIF method is called ‘stable’ for a number of reasons. First, as we already seen in Corollary 13, the convergence of the inner loop is always guaranteed. Secondly, we can again prove that the method is robust under small perturbations. In fact, even with a perturbed matrix K, we can retain the same conclusions of Proposition 12 by an opportune choice of the scalar c ($\|cK\| \leq 1$ is still enough). Moreover, Corollary 11 can be generalized to the SALIF method, showing how a perturbation in the signal propagates into the IMFs. To prove it, we first need an auxiliary result, that also shows how SALIF produces only a finite number of relevant IMFs, i.e. with norm greater than a fixed tolerance $\eta > 0$.

Proposition 14 *Let $\boldsymbol{g}$ be a discrete signal and $\delta > 0$, $\eta > 0$ fixed tolerances. Then the IMFs produced by Algorithm 5 satisfy*

$$\sum_j \|\boldsymbol{IMF}_j\|^2 \leq \|\boldsymbol{g}\|^2.$$

In particular, the algorithm outputs only a finite number of IMFs with norm greater than η.

Proof Suppose that at the j-th step of the outer loop of Algorithm 5, m_j is the number of iterations of the inner loop, $\boldsymbol{r}_j$ the remainder signal being considered and $A = c^2 K^T K$ the generated matrix, so that

$$\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \boldsymbol{IMF}_j, \qquad \boldsymbol{IMF}_j = (I - A)^{m_j} \boldsymbol{r}_j.$$

If $A = UDU^T$ is an eigendecomposition of A, we know from Proposition 12 that D has all diagonal elements nonnegative and upper bounded by 1. As a consequence,

$$\begin{aligned}\|\boldsymbol{IMF}_j\|^2 &= \|(I-A)^{m_j}\boldsymbol{r}_j\|^2 = \|U(I-D)^{m_j}U^T\boldsymbol{r}_j\|^2\\ &= \|U^T\boldsymbol{r}_j - \left[I-(I-D)^{m_j}\right]U^T\boldsymbol{r}_j\|^2\\ &\le \|U^T\boldsymbol{r}_j\|^2 - \|\left[I-(I-D)^{m_j}\right]U^T\boldsymbol{r}_j\|^2\\ &= \|\boldsymbol{r}_j\|^2 - \|U\left[I-(I-D)^{m_j}\right]U^T\boldsymbol{r}_j\|^2\\ &= \|\boldsymbol{r}_j\|^2 - \|\boldsymbol{r}_j - (I-A)^{m_j}\boldsymbol{r}_j\|^2 = \|\boldsymbol{r}_j\|^2 - \|\boldsymbol{r}_{j+1}\|^2,\end{aligned}$$

where the inequality comes from the relation $(1-x)^2 \le 1-x^2$ that holds for all $x \in [0,1]$. As a consequence the norm of $\boldsymbol{r}_j$ is decreasing in j and

$$\sum_j \|\boldsymbol{IMF}_j\|^2 \le \sum_j \|\boldsymbol{r}_j\|^2 - \|\boldsymbol{r}_{j+1}\|^2 \le \|\boldsymbol{r}_1\|^2 = \|\boldsymbol{g}\|^2.$$

□

Proposition 15 *Let $\boldsymbol{h}$ be a perturbation of the signal $\boldsymbol{g}$. Then for the sifting operation $\mathcal{S}(\boldsymbol{g}) := (I-A)\boldsymbol{g}$ with any A Hermitian matrix with spectrum in $[0,1]$ and any m, we have*

$$\|\mathcal{S}^m(\boldsymbol{g}+\boldsymbol{h}) - \mathcal{S}^m(\boldsymbol{g})\| \le \|\boldsymbol{h}\|.$$

Let now $\ell(x)_j$, m_j, $\boldsymbol{IMF}_j$ be the length function, the number of inner iterations and the IMF generated by the j-th outer loop of Algorithm 5 with input $\boldsymbol{g}$. If $\ell(x)_j$, m_j coincide with those generated by the same algorithm with input $\boldsymbol{g}+\boldsymbol{h}$, and if we call $\boldsymbol{IMF}_j^*$ the corresponding IMFs, then

$$\sum_j \|\boldsymbol{IMF}_j^* - \boldsymbol{IMF}_j\|^2 \le \|\boldsymbol{h}\|^2.$$

Proof Let $A = UDU^T$ an eigencomposition of A.

$$\begin{aligned}\|\mathcal{S}^m(\boldsymbol{g}+\boldsymbol{h}) - \mathcal{S}^m(\boldsymbol{g})\| &= \|\mathcal{S}^m(\boldsymbol{h})\| = \|(I-A)^m\boldsymbol{h}\|\\ &= \|U(I-D)^mU^T\boldsymbol{h}\| \le \|U^T\boldsymbol{h}\| = \|\boldsymbol{h}\|.\end{aligned}$$

For the second part, if we fix all $\ell(x)_j$, m_j generated by Algorithm 5 with input $\boldsymbol{v}$, then it is possible to prove by induction that the generated IMFs are linear in the input $\boldsymbol{v}$. In other words, calling $\boldsymbol{IMF}_j(\boldsymbol{v})$ such IMFs, we have $\boldsymbol{IMF}_j(\boldsymbol{g}+\boldsymbol{h}) = \boldsymbol{IMF}_j(\boldsymbol{g}) + \boldsymbol{IMF}_j(\boldsymbol{h})$ for every j. Notice now that the results in Proposition 14 hold even when we fix $\ell(x)_j$, m_j in the algorithm, since all generated matrices A still respect Proposition 12. As a consequence,

$$\begin{aligned}\sum_j \|\boldsymbol{IMF}_j^* - \boldsymbol{IMF}_j\|^2 &= \sum_j \|\boldsymbol{IMF}_j(\boldsymbol{g}+\boldsymbol{h}) - \boldsymbol{IMF}_j(\boldsymbol{g})\|^2\\ &= \sum_j \|\boldsymbol{IMF}_j(\boldsymbol{h})\|^2 \le \|\boldsymbol{h}\|^2.\end{aligned}$$

□

Notice that the proofs of the last two results can be repurposed as proofs of the corresponding results for the discrete IF algorithm. Instead, the opposite is not possible since the IF method benefits from the representation through the DFT (or the Fourier transform in the continuous case) that SALIF lacks completely. In particular, the set of IMFs produced by IF can be seen in the frequency domain as a partition of the frequencies into semi-stationary components, described in the discrete case by the classic Fourier base. In SALIF, though, we want to extract frequency modulated modes, that are not prone to be described as a proper combination of frequencies.
To summarize, we have

- the IF algorithm has a well developed theory in the continuous case, its discrete version always converges and can be efficiently implemented as FIF, but cannot capture non-stationary components with quickly varying frequencies,
- the ALIF algorithm has been empirically shown to be flexible enough to extract fully non-stationary components, but an analysis of the continuous version has never been conducted and the convergence in the discrete case is not guaranteed,
- the SALIF algorithm is always convergent and empirically it has shown to produce a more accurate decomposition than the ALIF algorithm, while withholding its ability to extract non-stationary components, but it is very slow, it lacks a full characterization of the generated decomposition both from a mathematical and physical point of view.

In the next section, we show how to design an alternative method, that is flexible enough to perform non-stationary analysis on the signals, but at the same time fast and provably convergent.

4 Resampled Iterative Filtering

The linear ALIF method makes use of a length function $\ell(x)$ to locally stretch a fixed filter $k(z)$ so that the convolution with the signal $g(z)$ smooths out the high-frequency oscillatory behaviour. The idea behind the Resampled Iterative Filtering (RIF) algorithm is to set a fixed length for the filter $k(z)$ and instead modify the signal through a global resampling function. In a sense, we want to locally stretch the signal, making the component of higher frequency approximately stationary, so that we are able to identify it through the fast IF algorithm.

As a clarifying example, suppose the signal $g(z)$ is a linear combination of M non-stationary components $\cos(\alpha_j(z))$ such that $\alpha_1'(z) > \alpha_2'(z) > \cdots > \alpha_M'(z) > \epsilon > 0$ are all continuous functions. Here the functions $\alpha_j'(z)$ are called instantaneous frequencies for the components (see [19]), and since $\alpha_j'(z) > \epsilon > 0$, then all the $\alpha_j(z)$ are C^1 and invertible functions. In order to extract the highest frequency component $\cos(\alpha_1(z))$ the ALIF method stretches the filter accordingly to $\ell(z) := \xi_0/\alpha_1'(z)$, whereas the RIF method applies the resampling $G(y) := \alpha_1^{-1}(y)$ to the signal, obtaining $\widetilde{g}(y) := g(G(y))$ that is now a linear combination of the components $\cos(\alpha_j(G(y)))$. Notice that the first component is now the simple stationary signal $\cos(y)$ with frequency 1, and all the others have instantaneous frequencies

$$(\alpha_j \circ G)'(y) = \alpha_j'(G(y))G'(y) < \alpha_1'(G(y))G'(y) = (\alpha_1 \circ G)'(y) = 1.$$

We have seen in Section 2 that the IF algorithm operates similarly to a band-pass filter, by isolating a neighbourhood of the wanted frequency from the rest, and in particular from the lower frequencies. The idea here is thus to use the IF method on the resampled signal $\widetilde{g}(y)$ in order to extract the first high-frequency stationary component, and thus separate it from the components with lower instantaneous frequencies as in $\widetilde{g}(y) = I_1(y) + \widetilde{r}_1(y)$, where $I_1(x)$ approximates a multiple of $\cos(y)$. A final resampling $H(z) := G^{-1}(z) = \alpha_1(z)$ brings back all the components to the original coordinates, $g(z) = \widetilde{g}(H(z)) = I_1(H(z)) + \widetilde{r}_1(H(z)) = IMF_1(z) + r_1(z)$, where $IMF_1 \sim \cos(\alpha_1(x))$ is now marked and stored as the first IMF of the signal, and the method is then iteratively applied to the reminder signal $r_1(z)$.

Following the steps described for the above example, we can now formalize the RIF algorithm, where the resampling function $G(y)$ is computed starting from the length function $\ell(z)$ given by the ALIF algorithm with a filter $k(t)$ with $\xi_0 = 1$. In particular, calling $\alpha'(z)$ the instantaneous frequency of the first extracted component, we know that $\ell(z) = 1/\alpha'(z)$ and we want that $(\alpha \circ G)'(y) = 1$, thus we can take $G(y) = \alpha^{-1}(y + \alpha(0))$ and compute it as

$$\ell(z) = G'(G^{-1}(z)) \quad \text{and} \quad G^{-1}(z) = \int_0^z \frac{1}{\ell(x)}\,\mathrm{d}x. \tag{14}$$

The full RIF algorithm is reported as Algorithm 6.

From the algorithm it is evident that, after the resampling, the steps are the same as the IF algorithm. In fact, we always extract almost stationary IMFs from the resampled signal, and then we operate the inverse sampling to obtain the respective IMFs for the original signal. The observation that the proposed RIF algorithm extracts almost stationary IMFs from the resampled signal is enough to infer that the RIF internal loop always converge to some IMF thanks to Theorem 2. In the next section we see how these properties carry over to the discrete case.

We point out that $G(x)$ depends on $\ell(x)$, so it must be computed every time we want to extract a new component. Notice that in the inner loop RIF is sifting the resampled signal $\widetilde{g} = g \circ G$, and this is equivalent, from a theoretical point of view, to applying on the original signal a resampled filter obtained via the inverse sampling $H := G^{-1}$.

$$\begin{aligned}
\mathcal{S}_{RIF}(g)(x) &= \widetilde{g}(H(x)) - \int_{\mathbb{R}} \widetilde{g}(y) k\left(H(x) - y\right) \mathrm{d}y \\
&= g(x) - \int_{\mathbb{R}} g(G(y)) k\left(H(x) - y\right) \mathrm{d}y
\end{aligned}$$

Algorithm 6 (Resampled IF Algorithm) IMFs = RIF(g, δ)

```
**Inputs:** $g$ real $L^2$ function, $\delta > 0$ stopping parameter
**Output:** IMFs is a set of $L^2$ simple oscillatory functions
IMFs = $\{\}$
initialize the remaining signal $r = g$
**while** the number of extrema of $r$ is $\geq 2$ **do**
    compute $\ell(x)$ and derive the resampling $G(y), G^{-1}(x)$ as in (14) and the resampled signal $h = r \circ G$
    $h_1 = h$
    $m = 1$
    **while** $\|h_m - h_{m-1}\| > \delta$ **do**
        $h_{m+1} = h_m - \int_{\mathbb{R}} h_m(y)k(x-y)\mathrm{d}y$
        $m = m + 1$
    **end while**
    IMFs = IMFs $\cup \{h_m \circ G^{-1}\}$
    $r = r - h_m \circ G^{-1}$
**end while**
```

$$= g(x) - \int_{\mathbb{R}} g(z)k\left(H(x) - H(z)\right)H'(z)\mathrm{d}z. \tag{15}$$

It can be also shown that RIF is actually a particular ALIF method, since

$$\begin{aligned}\int_{\mathbb{R}} g(G(y))k\left(H(x) - y\right)\mathrm{d}y &= \int_{\mathbb{R}} g(G(H(x) - w))k\left(w\right)\mathrm{d}w \\ &= \int_{\mathbb{R}} g(x + [G(H(x) - w) - x])k\left(w\right)\mathrm{d}w.\end{aligned}$$

with $t(x, y) = G(H(x) - y) - x$ in (2). Moreover, if we approximate at its first order $G(H(x) - w) \approx G(H(x)) - wG'(H(x)) = x - w\ell(x)$, where we used (14), then

$$\begin{aligned}\int_{\mathbb{R}} g(G(H(x) - w))\,k(w)\ \mathrm{d}w &\approx \int_{\mathbb{R}} g(x - w\ell(x))\,k(w)\ \mathrm{d}w \\ &= \int_{\mathbb{R}} g(x - r)\,k\left(\frac{r}{\ell(x)}\right)\frac{\mathrm{d}r}{\ell(x)} \\ &= \int_{\mathbb{R}} g(r)\,k\left(\frac{x - r}{\ell(x)}\right)\frac{\mathrm{d}r}{\ell(x)} \qquad (16)\end{aligned}$$

where (16) shows that Linear ALIF is a first-order approximation of RIF, and since RIF is a convergent method, we could ask whether it produces the same output as Linear ALIF. The answer is provided in the following result.

Proposition 16 *Suppose that $k(z)$ is a filter with $k(0) \neq 0$, $\xi_0 = 1$ and at least in $C^1(\mathbb{R})$. Moreover suppose that the resampling function $H(z) = G^{-1}(z)$ is at least in*

$C^2(\mathbb{R})$. In this case, the sifting operator in the RIF continuous algorithm 6 and the one in the ALIF continuous algorithm 3 coincide if and only if $\ell(x)$ is a constant function.

Proof Given a signal $g(x)$, in Linear ALIF the sifting operator is

$$\begin{aligned}\mathcal{S}_{ALIF}(g)(x) &= g(x) - \int_{\mathbb{R}} g(y) k\left(\frac{x-y}{\ell(x)}\right) \frac{1}{\ell(x)} \mathrm{d}y \\ &= g(x) - \int_{\mathbb{R}} g(y) k\left(H'(x)(z-y)\right) H'(x) \mathrm{d}y\end{aligned}$$

where $1/\ell(x) = (G^{-1})'(x) = H'(x)$ from (14). In RIF, the sifting operator is

$$\mathcal{S}_{RIF}(g)(x) = g(x) - \int_{\mathbb{R}} g(z) k\left(H(x) - H(z)\right) H'(z) \mathrm{d}z.$$

The two operators coincide for every starting signal $g(x)$ if and only if

$$k\left((x-y)H'(x)\right) H'(x) = k(H(x) - H(y))H'(y), \quad \forall x, y$$

and it is easily verified that this is true if $\ell(x)$ is constant, and therefore $H(z)$ is a linear function.

To prove the opposite implication, we now derive in y.

$$-k'\left((x-y)H'(x)\right) H'(x)^2 = -k'(H(x)-H(y))H'(y)^2 + k(H(x)-H(y))H''(y), \quad \forall x, y.$$

Substituting $x = y$, we conclude

$$-k'\left(0\right) H'(x)^2 = -k'(0)H'(x)^2 + k(0)H''(x)$$

and in particular, $H''(x) = 0$. As a consequence $H'(x)$ and $\ell(x) = 1/H'(x)$ are constant functions. □

4.1 Non-stationary Error Bounds

Let us suppose that the signal $g(x)$ is a linear combination of non-stationary purely oscillating components as in

$$g(x) := \sum_{j=1}^{M} a_j g_j(x), \qquad g_j(x) = \cos(\alpha_j(x)) \tag{17}$$

where $\alpha_1'(x) > \alpha_2'(x) > \cdots > \alpha_M'(x) > \epsilon > 0$ are all continuous functions and $|a_j| \leq P$ for any j. Moreover, suppose that $g(x)$ and all its components $g_j(x)$ are 1-periodic, and as a consequence $\alpha_j(1) - \alpha_j(0) = 2\pi s_j$ for any j, with integer numbers $s := s_1 > s_2 > \cdots > s_m > 0$. Following the RIF Algorithm 6 applied to $g(x)$, we find that in the first outer loop, we try to isolate the highest frequency component of

$$h(y) := \sum_{j=1}^{M} a_j h_j(y), \qquad h_j(y) = \cos(\alpha_j(\alpha_1^{-1}(y))), \qquad h_1(y) = \cos(y)$$

by applying the IF Algorithm 1. Notice that $h(y)$ and all $h_j(y)$ are now periodic on $[0, 2\pi s]$. With an abuse of notation, let us compress the domain to $[0, 1]$ with a linear change of variables $2\pi sz = y$, so that

$$h(z) := \sum_{j=1}^{M} a_j h_j(z), \qquad h_j(z) = \cos(\alpha_j(\alpha_1^{-1}(2\pi sz)) = \cos(\beta_j(z)),$$

are now all 1-periodic and $h_1(z) = \cos(2\pi sz)$. Moreover, we have already proved in the last section that for $j > 1$, $\beta_j'(z) < \beta_1'(z) = 2\pi s$, so the instantaneous frequencies of h_j are all bounded by $2\pi s$. By Theorem 2, in the case $h(z)$ is an $L^2(\mathbb{R})$ function, the inner loop of the algorithm with filter $k(z)$ returns a signal whose Fourier Transform is $(1 - \widehat{k})^m \widehat{h}$. In the case $h(z)$ is periodic, an analogous relation holds, where $\widehat{k}$ and $\widehat{h}$ are now distributions, or better, infinite vectors of their Fourier coefficients, and all the operations are to be intended elementwise.

Proposition 4 and the above formula show that the IF Algorithm extracts from $h(z)$ mainly components with frequency close to the first zero ξ_0 of the Fourier Transform of the filter $k(z)$, that we can choose thanks to the length L. In particular the iterative sifting tends to cancel all frequencies below $\xi_0 = 2\pi s$, thus the Algorithm will extract as the first IMF the component $h_1(z)$ plus, at most, the coefficients of the components $h_j(z)$ with frequency greater or equal than ξ_0.

In the case of stationary components, the non zero Fourier coefficients in $\widehat{h}_j(z)$ for $j > 1$ are all with index strictly less than s, so that the algorithm is able to correctly extract the first component $\widehat{h}_1(z)$. When the components are non-stationary, $\widehat{h}_j(z)$ may be non-zero also for high frequencies, thus the extracted IMF gets modified. We now give an estimate for the errors incurred in this method. The technical details are postponed to the appendix A.

Theorem 17 *Suppose $\beta : \mathbb{R} \to \mathbb{R}$ is a C^1 function with $\beta'(x) \in [a, b]$ where $0 < a < b$ and $R := b - a$. Suppose that $\beta(x + 1) = \beta(x) + 2k\pi$ for some positive integer k and for all $x \in \mathbb{R}$. Let $f(x) := \cos(\beta(x))$ and let $f(x)_N$ be the partial sum of its Fourier series with coefficients up to N as in*

$$f(x)_N = \sum_{j=-N}^{N} e^{i2\pi jx} d_j := \sum_{j=-N}^{N} e^{i2\pi jx} \int_0^1 \cos(\beta(y))e^{-i2\pi jy} dy.$$

If $G := 2\pi N - b > 0$, then

$$\|f(x) - f(x)_N\|_2^2 \le \min\left\{ \left(\frac{b}{G + b + 2\pi}\right)^2, \frac{R^2}{\pi^3 G} \right\}.$$

If we apply now the above theorem with $j > 1$, $f(z) = h_j(z)$ and $N = s-1$, we find that $P\|f(x) - f(x)_N\|_2$ is a bound on the perturbation of the IMF caused by the j-th component h_j, and it is proportional to both

$$\frac{b}{G+b+2\pi} = \frac{\max_z \beta_j'(z)}{2\pi s} = \max_x \frac{\alpha_j'(x)}{\alpha_1'(x)}$$

and to

$$R = \max_z \beta_j'(z) - \min_z \beta_j'(z) = 2\pi s \left(\max_x \frac{\alpha_j'(x)}{\alpha_1'(x)} - \min_x \frac{\alpha_j'(x)}{\alpha_1'(x)} \right).$$

As a consequence h_j affects the first IMF with an error that is inversely proportional to the relative distance of the instantaneous frequencies of the original components $g_1(x)$ and $g_j(x)$, and directly proportional to how close the resampled component $h_j(z)$ is to a stationary signal.

We would like to stress that these are loose bounds, since the method extracts only selected frequencies near the zeros of $\widehat{k}$, whose norm is way less than all the higher-or-equal-than-ξ_0 frequencies.

4.2 Fast Resampled Iterative Filtering

First of all we review how to possibly implement a discrete version of RIF. As with IF, we suppose the signal $g(x)$ is only known on $[0,1]$ and that it is a 1-periodic C^1 function. Consider the IF sifting operator on the resampled signal

$$g_{m+1}(G(y)) = g_m(G(y)) - \int_{\mathbb{R}} g_m(G(z))\, k(y-z)\mathrm{d}z,$$

where $h_m = g_m \circ G$ has domain $[0,M]$ and $G:[0,M] \to [0,1]$ is C^1. Recall that from (14), $M = G^{-1}(1) = \int_0^1 \ell(z)^{-1}\,\mathrm{d}z$, where $\ell(z)^{-1}$ represents the highest instantaneous frequency of the signal. Since a signal with two or less extrema on $[0,1]$ is considered a trend signal, we can at least assume $\ell(z)^{-1} > 2\pi$ and as a consequence $M > 2\pi$.

We thus consider the regular grid $x_i := Mi/n$ for $i = 0,\dots,n-1$. and extend the signal cyclically on the real line, meaning that $h_m(sM+x) := h_m(x)$ for every $s \in \mathbb{Z}$ and every $x \in [0,M)$. The quadrature rule on the discretization points yields

$$h_{m+1}(x_i) \approx h_m(x_i) - \frac{M}{nP} \sum_{j\in\mathbb{Z}} h_m(x_j) k(x_i - x_j),$$

where P is a renormalizing constant given by

$$P := \frac{M}{n} \sum_{j\in\mathbb{Z}} k(x_i - x_j) \approx \int_{\mathbb{R}} k(z)\,\mathrm{d}z = 1$$

Notice that the above formula coincides with the IF moving average with length $L = 1/M$, and can be expressed through a Hermitian circulant matrix

K with first row

$$\boldsymbol{k}_1 = \frac{M}{nP}\left[k(0),\, k\left(\frac{M}{n}\right), \ldots,\, k\left(s\frac{M}{n}\right), 0 \ldots 0,\, k\left(s\frac{M}{n}\right), \ldots,\, k\left(\frac{M}{n}\right)\right]$$

where $s = \lfloor n/M \rfloor \leq n/6$. If $\boldsymbol{h}_m$ is the vector $[h_m(x_i)]_{i=0,\ldots,n-1}$, the sifting operator becomes

$$\boldsymbol{h}_{m+1} = (I - K)\boldsymbol{h}_m$$

where $I-K$ is still a Hermitian and circulant matrix, so that the matrix vector multiplication can be performed efficiently through a FFT. In particular, from Proposition 9,

$$\boldsymbol{h}_{m+1} = \mathrm{iDFT}\left((1 - \mathrm{DFT}(\boldsymbol{k}_1)) \circ \mathrm{DFT}(\boldsymbol{h}_m)\right),$$

where $\circ$ stands for the Hadamard (or element-wise) product between vectors, $1 = \mathrm{DFT}(e_1)$ is the all-ones vectors, coming from $\mathrm{DFT}(I\boldsymbol{h}_m) = \mathrm{DFT}(e_1) \circ \mathrm{DFT}(\boldsymbol{h}_m)$, and DFT, iDFT stand for Discrete Fourier Transform and its inverse, respectively. Moreover, since

$$\mathrm{DFT}(\boldsymbol{h}_{m+1}) = (1 - \mathrm{DFT}(\boldsymbol{k}_1)) \circ \mathrm{DFT}(\boldsymbol{h}_m) = (1 - \mathrm{DFT}(\boldsymbol{k}_1))^{\circ m} \circ \mathrm{DFT}(\boldsymbol{h}_1)$$

we only need to perform two DFTs $\mathrm{DFT}(\boldsymbol{k}_1)$, $\mathrm{DFT}(\boldsymbol{h}_1)$, find the index m satisfying the stopping condition, compute the $m+1$ Hadamard products and then perform an iDFT outside of the inner loop. Calling $\widehat{\boldsymbol{h}_m} := \mathrm{DFT}(\boldsymbol{h}_m)$, notice that $\|\widehat{\boldsymbol{h}_{m+1}} - \widehat{\boldsymbol{h}_m}\| = \|\boldsymbol{h}_{m+1} - \boldsymbol{h}_m\|$, so the usual stopping criterion $\|\boldsymbol{h}_{m+1} - \boldsymbol{h}_m\| \leq \delta$ can be checked directly on $\widehat{\boldsymbol{h}_m}$, thus avoiding iterated computations of Fourier transforms.

The resulting method is reported in Algorithm 7.

Notice that while the internal loop only consists of Hadamard multiplications among vectors, and its convergence and stability properties can be analysed with the same tools used for the IF algorithm, as in Proposition 7, Theorem 8 and the first part of Corollary 11, in the outer loop we perform operations that may lead to a loss in accuracy of the method. We can thus adopt a spline interpolation to mitigate the accuracy loss, and even in this case, the computational cost of the outer loop is still $O(n \log n)$ operations due to the Fourier transforms.

From Proposition 7 and Theorem 8, one can state analogous convergence results to the Discrete IF Algorithm.

Corollary 18 *Given a double-convoluted filter $k = \omega \star \omega$, then the inner loop of the RIF Algorithm 6 converges for any initial function $h(x)$. In particular, the limit*

$$\lim_{m\to\infty} \hat{\boldsymbol{h}}_{\boldsymbol{m}} = \lim_{m\to\infty} \hat{\boldsymbol{k}}_{\boldsymbol{1}}^{m} \circ \hat{\boldsymbol{h}}_{\boldsymbol{1}}$$

Algorithm 7 (Fast Resampled Iterative Filtering) $\boldsymbol{IMFs} = \mathrm{FRIF}(\boldsymbol{g}, \delta)$

```
Inputs: $\boldsymbol{g} \in \mathbb{R}^n$ discretized signal, $\delta > 0$ stopping parameter
Output: IMFs is a set of discretized simple oscillatory components in $\mathbb{R}^n$
IMFs = {}
initialize the remaining signal $\boldsymbol{r} = \boldsymbol{g}$
while the number of extrema of $\boldsymbol{r}$ is $\geq 2$ do
    compute $\ell(x)$ based on a time-frequency representation of $\boldsymbol{r}$, the resampling function $G^{-1}(x) = \int \ell(t)^{-1}\,\mathrm{d}t$ via numerical integration, and the constant $M$ as $M = G^{-1}(1)$
    compute the resampled signal $\boldsymbol{h}$ through interpolation of $\boldsymbol{r}$ on the points $G(y_i)$ where $y_i = Mi/n$ and the filter $\boldsymbol{k}_1$ that capture the highest frequency contained in $\boldsymbol{h}$.
    $\boldsymbol{h}_1 = \boldsymbol{h}$
    $\hat{\boldsymbol{h}}_1 = \mathrm{DFT}(\boldsymbol{h}_1)$, $\hat{\boldsymbol{k}}_1 = 1 - \mathrm{DFT}(\boldsymbol{k}_1)$
    $m = 1$
    while $\|\widehat{\boldsymbol{h}}_{m+1} - \widehat{\boldsymbol{h}}_m\| > \delta$ do
        $\hat{\boldsymbol{h}}_{m+1} = \hat{\boldsymbol{k}}_1 \circ \hat{\boldsymbol{h}}_m$
        $m = m + 1$
    end while
    $\boldsymbol{h}_m = \mathrm{iDFT}(\hat{\boldsymbol{h}}_m)$
    compute the IMF $\boldsymbol{I}$ through interpolation of $\boldsymbol{h}_m$ on the points $G^{-1}(z_i)$ where $z_i = i/n$
    IMFs = IMFs $\cup \{\boldsymbol{I}\}$
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{I}$
end while
```

converges for any vector $\hat{\boldsymbol{h}}_{\mathbf{1}}$*, and the stopping condition is met for a* m *upper bounded by* $m(\boldsymbol{h}_{\mathbf{1}})$*, the smallest positive integer such that*

$$\frac{m^m}{(m+1)^{m+1}} < \max\left\{\frac{\delta}{\|\hat{\boldsymbol{h}}_{\mathbf{1}}\|}, \frac{\delta}{\sqrt{n-1-p}\,\|\hat{\boldsymbol{h}}_{\mathbf{1}}\|_\infty}\right\},$$

where p *is the number of elements* 1 *in* $\hat{\boldsymbol{k}}_{\mathbf{1}}$*.*

We have seen that the inner loops of the FRIF algorithm is provably convergent, and that its computational time is comparable with the FIF method. It's worthwhile also to mention that the interpolations may introduce a loss in accuracy in the solution. One can formulate a different, but equivalent, version of the continuous algorithm that does not require a resampling of the signal, since, as shown in (15), the sifting operator can be rewritten as

$$\mathcal{S}_{RIF}(g)(x) = g(x) - \int_{\mathbb{R}} g(z) k\left(H(x) - H(z)\right) H'(z) \mathrm{d}z.$$

The operator can be thus discretized into a diagonalizable matrix with all the eigenvalues real, nonnegative and less than 1, for which a result similar to Proposition 7 can be proved. The resulting algorithm thus avoids the need to interpolate the signal two times per IMF, but the resulting matrix is not circulant (or even Hermitian), so we lose the fast implementation that was possible in Algorithm 7.

In the numerical examples, we will also show that nonetheless the FRIF algorithm produces sensible decompositions, but first let us address another property of the method.

4.3 Anti-Aliasing Property

In the discrete setting, the resampling of the signal $g(x)$ may in theory come with an undersampling of the highest frequencies, leading to aliasing effects. Here we show that in the FRIF algorithm, this is actually not a problem.

Suppose that the signal can be split into components as in (17), i.e. $g(x)$ presents a decomposition in components $g_j(x)$ which have instantaneous frequencies $I_1(x), I_2(x), I_3(x), \ldots$, where $I(x) := I_1(x)$ has the highest instantaneous frequency among all the components, and recall that from (14) we have

$$G^{-1}(z) = \int_0^z I(x)\,\mathrm{d}x, \qquad M = G^{-1}(1) = \int_0^1 I(x)\,\mathrm{d}x.$$

The resampled signal $\widetilde{h}(x) = g(G(x))$ has thus domain $[0, M]$, but in the discrete setting we treat it as a signal over $[0, 1]$, so we are actually working with

$$h(x) := \widetilde{h}(Mx) = g(G(Mx)).$$

The signal $h(x)$ presents now a new decomposition in components $h_j(x) = g_j(G(Mx))$ with instantaneous frequency given by $I_j(G(Mx))G'(Mx)M$, and in particular the first component has now frequency

$$\begin{aligned} I(G(Mx))G'(Mx)M &= I(G(G^{-1}(z)))G'(G^{-1}(z))M \\ &= \frac{I(z)M}{(G^{-1})'(z)} = M = \int_0^1 I(x)dx \end{aligned}$$

that is surely less than $\|I(x)\|_\infty$. Moreover, since $G(z)$ is increasing and $I(x) \geq I_j(x)\,\forall x, j$, then

$$I(G(Mx))G'(Mx)M \geq I_j(G(Mx))G'(Mx)M, \qquad \forall x$$

meaning that $h_1(x)$ has still the biggest instantaneous frequency among the $h_j(x)$.

This allows to infer that the greatest instantaneous frequency of the resampled signal $h(x)$ is less than the greatest instantaneous frequency of the original signal $g(x)$, despite being evaluated in FRIF on the same number of points. The resampling thus does not create artificial high frequency components, and the FRIF algorithm does not suffer from aliasing problems.

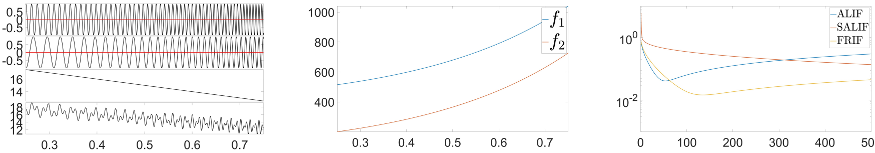

Fig. 1: Example 1. Left panel: the components f_1 and f_2, respectively first and second row,the trend, third row, and the signal f, bottom row. Central panel: exponential instantaneous frequencies of f_1 and f_2. Right panel: relative error in norm 2 between the ground truth and IMF$_1$ produced by ALIF, SALIF, and FRIF algorithms.

5 Numerical Experiments

In this section we show and compare the performances of all the reviewed techniques. In order to study the signals and their decompositions in time-frequency, we will rely on the so called IMFogram, a recently developed algorithm [38], which allows to represent the frequency content of all IMFs. The IMFogram proves to be a robust, fast and reliable way to obtain the time-frequency representation of a signal, and it has been shown to converge, in the limit, to the well know spectrogram based on the FFT [17].

The following tests have been conducted using MATLAB® R2021a installed on a 64–bit Windows 10 Pro computer equipped with a 11th Gen Intel© Core® i7-1165G7 at 2.80GHz processor and 8GB RAM. All tested examples and algorithms are freely available at [1].

5.1 Example 1

We consider the artificial signal f, plotted in the left panel, bottom row, of Figure 1, which contains two nonstationary components with exponentially changing frequencies f_1 and f_2, plus a trend f_3. In particular

$$\begin{aligned} f_1(x) &= \cos(20e^{t\pi} + 120\pi t) \\ f_2(x) &= \cos(20e^{t\pi} + 20\pi t) \\ f_3(x) &= -10x + 20 \end{aligned}$$

where x varies in [0, 1] and is sampled over 10^4 points.

The f_1 and f_2 components and f signal are plotted in the left panel of Figure 1, whereas f_1 and f_2 frequencies are shown in the central panel.

In Table 1 we report the computational time required by ALIF, SALIF and FRIF with a fixed stopping criterion based on (5). In the same table we summarize the performance of the three techniques in terms of inner loop

[1] www.cicone.com

iterations required to produce the two IMFs and the relative error measured as ratio between the norm 2 of the difference between the computed IMF and the corresponding ground truth, and the norm 2 of the ground truth itself.

Example 1	ALIF	SALIF	FRIF
time(s)	16.3293	26.9395	**0.9107**
err_1	0.040260	0.117824	**0.006535**
err_2	1.051461	0.117842	**0.006543**
err_3	0.049352	0.000084	**0.000017**
num of iter IMF_1	61	175	80
num of iter IMF_2	500	155	4

Table 1: performance of various techniques when applied on Example 1, measured as relative errors in norm 2 and number of iterations.

From Table 1 results it is clear that FRIF proves to converge quickly to a really accurate solution. In fact, it takes less than a second to produce a decomposition which has a relative error which is order of magnitudes smaller than the ones produced using ALIF and SALIF methods. Furthermore ALIF and SALIF decompositions require more than 16 and 26 seconds, respectively, to converge. This is confirmed by the results shown in the right panel of Figure 1, where we compare the norm 2 relative error of the IMF_1 obtained using ALIF, SALIF, and FRIF algorithms for subsequent steps in the inner loops when we remove the stopping condition. ALIF initially tends toward the right solution. At 35 steps the relative error reach the minimum value of 0.0262, and then, after that, the instabilities of the method show up and drive the solution far away from the correct one. SALIF, instead, is clearly convergent, in fact the solution is moving steadily to the exact one. However SALIF converge rate is slow, as proven by the relative error which is slowly decaying. In fact, after 500 inner loop steps, the relative error is still around 0.0179. Finally, FRIF quickly converge to a really good approximation of the right solution, at 73 steps the error is minimal with a relative error of 0.0064. After this step, the relative error restarts growing due to the chosen stopping criterion. It is important to remember that, in general, the ground truth is not known. This is the reason why the stopping criterion adopted in these techniques does not rely on the ground truth knowledge. Hence, as a consequence, FRIF, ALIF, and SALIF do not necessarily stop when the actual best approximation of the ground truth is achieved. For example, one can see that the ALIF algorithm doesn't stop in the computation of the second IMF of the signal. Studying what could be an ideal stopping criterion and how to tune it properly is outside the scope of this work.

5.2 Example 2

In this second example, we start from the artificial signal h which contains two nonstationary components, h_1 and h_2, and a trend h_3,

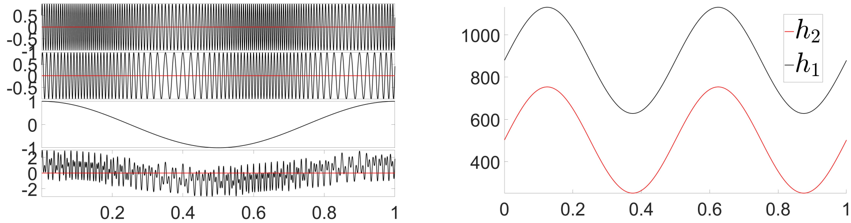

Fig. 2: Example 2. Left panel: the components h_1 and h_2, respectively first and second row, and the signal h, bottom row. Right panel: exponential instantaneous frequencies of h_1 and h_2.

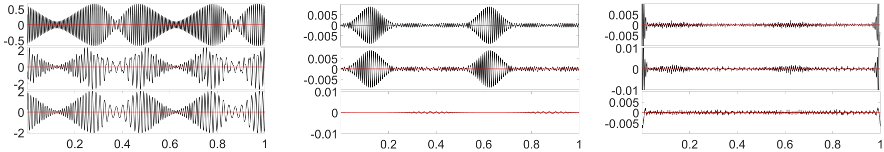

Fig. 3: Example 2. Difference between the ground truth and the derived decomposition via ALIF (left), SALIF (central), FRIF (right).

$$\begin{aligned} h_1(x) &= \cos(20\cos(4\pi t) - 160\pi t) \\ h_2(x) &= \cos(20\cos(4\pi t) - 280\pi t) \\ h_3(x) &= \cos(2\pi t) \end{aligned}$$

where x varies in $[0,\, 1]$ and is sampled over 8000 points.

The h_1, h_2, the trend component, and h signal are plotted in the left column of Figure 2, whereas h_1 and h_2 frequencies are shown in the right panel.

In Table 2 we report the performance of ALIF, SALIF and FRIF techniques. In Figure 3 we show the differences between the IMFs produced by the different methods and the known ground truth. It is evident both from the table and the figure that the proposed FRIF method outperforms the other approaches both from the efficiency and the accuracy point of view.

5.3 Example 3

In this example we show the robustness of the proposed FRIF approach to noise. To do so, we consider the signal h studied in Example 2 and we perturb it by additive Gaussian noise. In Figure 4 we plot on the left panel the perturbed signal when the signal to noise ratio (SNR) is of 8.6 dB. On the right panel we report the decomposition produced by FRIF. It is evident that the

Example 2	ALIF	SALIF	FRIF
time(s)	16.0005	20.2120	**1.0958**
err$_1$	0.457672	0.003584	**0.003426**
err$_2$	1.374017	0.003591	**0.003292**
err$_3$	1.304946	**0.000229**	0.000908
num of iter IMF$_1$	500	468	81
num of iter IMF$_2$	500	8	11

Table 2: Example 2 performance of ALIF, SALIF and FRIF, measured as relative errors in norm 2 and iteration number.

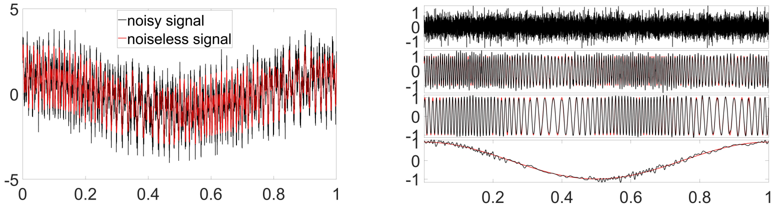

Fig. 4: Example 3. Left panel, the noisy signal compared with the noiseless signal h defined in Example 2. The SNR is around 8.6 dB. Right panel, the IMF decomposition derived by FRIF.

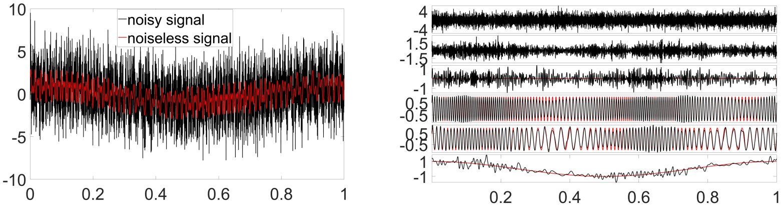

Fig. 5: Example 3. Left panel, the noisy signal with SNR around 1.3 dB compared with the noiseless signal h of Example 2. Right panel, the corresponding FRIF decomposition compared with the ground truth.

method can separate properly the random perturbation in the first row, from the deterministic components in the following three rows.

This result is confirmed even if we increase the SNR to 1.3 dB, left panel of Figure 5. It is evident from this figure that this level of noise is quite high. Nevertheless FRIF method proves to be able still to separate the deterministic signal from the additive Gaussian contribution, as shown in the left panel of Figure 5.

5.4 Example 4

We conclude the numerical section with an example based on a real life signal. We consider the recording of the sound emitted by a bat, shown in the left panel of Figure 6. In the central panel, we show the associated time-frequency plot

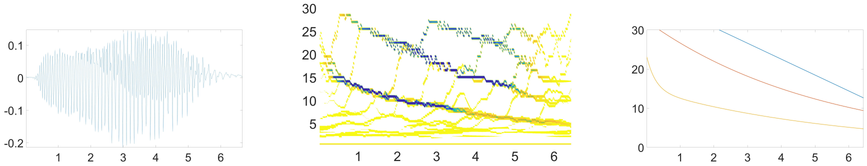

Fig. 6: Example 4. Left panel, sound produced by a bat. Central panel, the corresponding IMFogram time-frequency plot where darker color means stronger energy. Right panel, instantaneous frequency curves inferred from the IMFogram plot. In this panel, different colors represent different instantaneous frequency curves.

obtained using the IMFogram [38]. From this plot we observe that this signal appears to contain three main simple oscillatory components which present rapid changes in frequencies. Those are classical examples of the so called chirps. By using a curve extraction method, it is possible to derive from the IMFogram the instantaneous frequency curves plotted in the right panel of Figure 6. As briefly mentioned earlier, the identification of these instantaneous frequency curves is of fundamental importance for the proper functioning of FRIF, but it is also a research topic per se. In this work, we assume that they can be computed accurately and we postpone the analysis of how to compute them in a robust and accurate way to future works.

By leveraging on the extracted curves, we run FRIF algorithm and derive the decomposition shown in the left most panel of Figure 7. The first three IMFs produced correspond to the three main chirps observed in the IMFogram, which is depicted in the central panel of Figure 6. This is confirmed by running IMFogram separately on the first three IMFs produced by FRIF. The results are shown in the rightmost 3 panels of Figure 7. From these plots it becomes clear that the algorithm is able to separate in a clean way the three chirps contained in the signal.

6 Conclusions

In this work, we introduced the Resampled Iterative Filtering (RIF), and, in the discrete setting, the Stable Adaptive Local Iterative Filtering (SALIF) and Fast Resampled Iterative Filtering (FRIF), which are capable of decomposing non-stationary signals into simple oscillatory components, even in presence of fast changes in their instantaneous frequencies, like in chirps. We have analyzed them from a theoretical standpoint, showing, among other things, that it is possible to guarantee a priori their convergence. Furthermore, we have tested them using several artificial and real-life examples.

More is yet to be said about the problem. In particular, all these methods are dependent on the computation of a length function $\ell(x)$ which is, de facto, the reciprocal of the instantaneous frequency curve associated with each component contained in the signal. This function is required to guide the

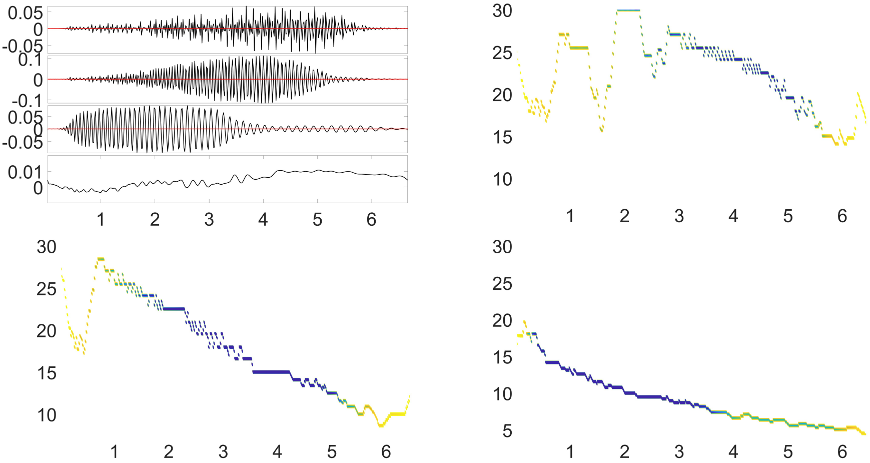

Fig. 7: Example 4. First row, left panel, IMF decomposition produced by FRIF. First row right panel and second row left and right panels, the IMFogram time-frequency plots associated with the first, second, and third row in the FRIF decomposition, respectively. In these time-frequency plots, darker color means stronger energy.

aforementioned methods, including ALIF itself, in the extraction of physically meaningful IMFs. The identification of instantaneous frequency curves associated with each component, which are contained in a given signal, is a research topic per se, and it is out of the scope of the present work. This is why we plan to study this problem in the future.

Other open problems regard the selection of an optimal stopping criterion, including its tuning, to be used in this kind of methods, an analysis of the number of relevant IMFs produced by SALIF and RIF, their local orthogonality, and a perturbation analysis for RIF. We plan to work in these directions in the future.

Finally, we plan to work on the extension of the proposed techniques to handle multidimensional and multivariate signals.

A Error Estimation for Non-Stationary Components

Here we report the proof for Theorem 17. First, let us show a more general result.

Proposition 19 *Suppose $\alpha : \mathbb{R} \to \mathbb{R}$ is a C^1 function with $\alpha'(x) \in [a, b]$ where $a < b$ and $ab \geq 0$. Call $R := b - a$ and suppose that $\alpha(1) = \alpha(0) + 2k\pi$ for some integer k.*

If now d_j is the j-th Fourier coefficient of $e^{\mathrm{i}\alpha(x)}$ as in

$$d_j := \int_0^1 e^{\mathrm{i}\alpha(x)} e^{-\mathrm{i}2\pi jx} dx,$$

and if $2\pi j > b$, then

$$|d_j| \leq \frac{1}{\pi}\frac{R}{2\pi j - b}.$$

Proof From the relation $\alpha(1) = \alpha(0) + 2k\pi$ we find that $2k\pi = \alpha(1) - \alpha(0) = \int_0^1 \alpha'(x) \in [a, b]$. Call $\psi(x) := 2\pi jx - \alpha(x)$ and notice that

$$\psi'(x) = 2\pi j - \alpha'(x) \in [q, p], \qquad p := 2\pi j - a \geq q := 2\pi j - b > 0,$$

so ψ is invertible with ψ^{-1} in C^1. We can then define the function $\varphi(y)$ as $\varphi(y) := (\psi^{-1})'(y) = 1/\psi'(\psi^{-1}(y)) \in [p^{-1}, q^{-1}]$, and by the Fourier formula,

$$d_j = \int_0^1 e^{\mathrm{i}[\alpha(x)-2\pi jx]} dx = \int_0^1 e^{-\mathrm{i}\psi(x)} dx = \int_{\psi(0)}^{\psi(1)} e^{-\mathrm{i}y}\varphi(y)dy,$$

where $\psi(1) - \psi(0) = 2\pi j - (\alpha(1) - \alpha(0)) = 2\pi j - 2\pi k$. Call $s := j - k$ and notice that $2\pi s = \int_0^1 \psi'(x) \in [q, p]$ and in particular $p \geq 2\pi s \geq q > 0$. Now

$$\begin{aligned} |d_j| = \left| \int_{-\alpha(0)}^{2\pi j - \alpha(1)} e^{-\mathrm{i}y}\varphi(y)dy \right| &= \left| \int_0^{2\pi} e^{-\mathrm{i}sz} e^{\mathrm{i}\alpha(0)}\varphi(sz - \alpha(0))sdz \right| \\ &= s\left| \int_0^{2\pi} e^{-\mathrm{i}sz}\varphi(sz - \alpha(0))dz \right| \end{aligned}$$

and the integral of the exponential is zero over $[0, 2\pi]$, therefore we can add to $\varphi(y)$ any constant without changing the result. As a consequence,

$$|d_j| = s\left| \int_0^{2\pi} e^{-\mathrm{i}sz}\left(\varphi(sz - \alpha(0)) - \frac{q^{-1} + p^{-1}}{2}\right) dz \right| = s\left| \int_0^{2\pi} e^{-\mathrm{i}sz}\phi(z)dz \right|$$

where $\phi(z) := \varphi(sz - \alpha(0)) - \frac{q^{-1}+p^{-1}}{2}$ is a real function bounded in absolute value by $\frac{q^{-1}-p^{-1}}{2}$. Suppose z_0 is the argument of $\int_0^{2\pi} e^{-\mathrm{i}sz}\phi(z)dz$ so that

$$|d_j| = se^{-\mathrm{i}z_0}\int_0^{2\pi} e^{-\mathrm{i}sz}\phi(z)dz \in \mathbb{R}$$

and its imaginary part is zero, leading to

$$\begin{aligned} |d_j| &= s\Re\left(e^{-\mathrm{i}z_0}\int_0^{2\pi} e^{-\mathrm{i}sz}\phi(z)dz\right) \\ &= s\int_0^{2\pi} \cos(sz + z_0)\phi(z)dz \\ &\leq s\frac{q^{-1} - p^{-1}}{2}\int_0^{2\pi} |\cos(sz + z_0)|dz = 2s(q^{-1} - p^{-1}). \end{aligned}$$

Using that $p \geq 2\pi s$ and $p - q = R$ we conclude that

$$|d_j| \leq \frac{R}{q\pi}.$$

□

Corollary 20 *Suppose $\alpha : \mathbb{R} \to \mathbb{R}$ is a C^1 function with $\alpha'(x) \in [a, b]$ where $0 < a < b$. Call $R := b - a$ and suppose that $\alpha(1) = \alpha(0) + 2k\pi$ for some integer k. If now d_j is the j-th Fourier coefficient of $\cos(\alpha(x))$ as in*

$$d_j := \int_0^1 \cos(\alpha(x))e^{-\mathrm{i}2\pi jx}dx,$$

and $2\pi j - b > 0$, then

$$|d_j| \le \frac{1}{\pi}\frac{R}{2\pi j - b}.$$

Proof Since $2\cos(\alpha(x)) = e^{\mathrm{i}\alpha(x)} + e^{-\mathrm{i}\alpha(x)}$ and both $\alpha(x)$ and $-\alpha(x)$ satisfy the hypotheses of Proposition 19, we can estimate d_j through the mean of the Fourier coefficients $d_{j,1}$ and $d_{j,2}$ respectively of $e^{\mathrm{i}\alpha(x)}$ and $e^{-\mathrm{i}\alpha(x)}$

$$|d_j| \le \frac{|d_{j,1}| + |d_{j,2}|}{2} \le \frac{1}{2\pi}\left(\frac{R}{2\pi j - b} + \frac{R}{2\pi j + b}\right) \le \frac{1}{\pi}\frac{R}{2\pi j - b}.$$

□

Going back to Theorem 17, notice that $f(x)$ is a C^1 periodic function with continuous derivative $f'(x)$ whose Fourier coefficients are $2\pi\mathrm{i}nd_n$ and $|f'(x)| = |\beta'(x)\sin(\beta(x))| \le b$. By Parseval Identity,

$$\|f'(x)\|_2^2 = \sum_{n=-\infty}^{+\infty}(2\pi n)^2|d_n|^2 = \int_0^1 f'(x)^2dx \le b^2.$$

The series of $n^2|d_n|^2$ thus converges, and

$$\|f - f_N\|_2^2 = \sum_{|n|>N}|d_n|^2 = \sum_{|n|>N}\frac{(2\pi n)^2|d_n|^2}{(2\pi n)^2} \le \left(\frac{b}{2\pi(N+1)}\right)^2 = \left(\frac{b}{G + b + 2\pi}\right)^2.$$

From Corollary 20, we already have a bound on d_j leading to

$$\|f - f_N\|_2^2 = \sum_{|n|>N}|d_n|^2 \le \frac{R^2}{\pi^2}\sum_{|n|>N}\frac{1}{(2\pi n - b)^2} \le \frac{R^2}{\pi^2}\int_N^\infty \frac{1}{(2\pi x + b)^2} + \frac{1}{(2\pi x - b)^2}$$

$$= \frac{R^2}{2\pi^3}\left[\frac{1}{2\pi N + b} + \frac{1}{2\pi N - b}\right] \le \frac{R^2}{\pi^3 G}.$$

References

[1] Hofmann-Wellenhof, B., Lichtenegger, H., Wasle, E.: GNSS–global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More. Springer, Berlin, Germany (2007)

[2] Li, B.N., Dong, M.C., Vai, M.I.: On an automatic delineator for arterial blood pressure waveforms. Biomedical Signal Processing and Control **5**(1), 76–81 (2010)

[3] Camporeale, E., Sorriso-Valvo, L., Califano, F., Retinò, A.: Coherent structures and spectral energy transfer in turbulent plasma: a space-filter approach. Physical review letters **120**(12), 125101 (2018)

[4] Wood, A.G., Alfonsi, L., Clausen, L.B., Jin, Y., Spogli, L., Urbář, J., Rawlings, J.T., Whittaker, I.C., Dorrian, G.D., Høeg, P., *et al.*: Variability of ionospheric plasma: results from the esa swarm mission. Space Science Reviews **218**(6), 52 (2022)

[5] De Santis, A., Marchetti, D., Pavón-Carrasco, F.J., Cianchini, G., Perrone, L., Abbattista, C., Alfonsi, L., Amoruso, L., Campuzano, S.A., Carbone, M., *et al.*: Precursory worldwide signatures of earthquake occurrences on swarm satellite data. Scientific reports **9**(1), 20287 (2019)

[6] Nuttall, L.: Characterizing transient noise in the ligo detectors. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **376**(2120), 20170286 (2018)

[7] Flandrin, P.: Time-frequency/time-scale Analysis. Academic press, Cambridge, Massachusetts, USA (1998)

[8] Lin, L., Wang, Y., Zhou, H.: Iterative filtering as an alternative algorithm for empirical mode decomposition. Advances in Adaptive Data Analysis **1**(04), 543–560 (2009)

[9] Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences **454**(1971), 903–995 (1998)

[10] Wu, Z., Huang, N.E.: Ensemble empirical mode decomposition: a noise-assisted data analysis method. Advances in adaptive data analysis **1**(01), 1–41 (2009)

[11] Yeh, J.-R., Shieh, J.-S., Huang, N.E.: Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. Advances in adaptive data analysis **2**(02), 135–156 (2010)

[12] Torres, M.E., Colominas, M.A., Schlotthauer, G., Flandrin, P.: A complete ensemble empirical mode decomposition with adaptive noise. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4144–4147 (2011). IEEE

[13] Zheng, J., Cheng, J., Yang, Y.: Partly ensemble empirical mode decomposition: An improved noise-assisted method for eliminating mode mixing. Signal Processing **96**, 362–374 (2014)

[14] Huang, C., Yang, L., Wang, Y.: Convergence of a convolution-filtering-based algorithm for empirical mode decomposition. Advances in Adaptive Data Analysis **1**(04), 561–571 (2009)

[15] Ur Rehman, N., Mandic, D.P.: Filter bank property of multivariate empirical mode decomposition. IEEE transactions on signal processing **59**(5), 2421–2426 (2011)

[16] Huang, N.E.: Introduction to the hilbert–huang transform and its related mathematical problems. Hilbert–Huang transform and its applications, 1–26 (2014)

[17] Cicone, A., Li, W.S., Zhou, H.: New theoretical insights in the decomposition and time-frequency representation of nonstationary signals: the imfogram algorithm. preprint (2021)

[18] Cicone, A.: Iterative filtering as a direct method for the decomposition of nonstationary signals. Numerical Algorithms, 1–17 (2020)

[19] Cicone, A., Liu, J., Zhou, H.: Adaptive local iterative filtering for signal decomposition and instantaneous frequency analysis. Applied and Computational Harmonic Analysis **41**(2), 384–411 (2016)

[20] Cicone, A., Zhou, H.: Numerical analysis for iterative filtering with new efficient implementations based on fft. Numerische Mathematik **147**(1), 1–28 (2021)

[21] An, X.: Local rub-impact fault diagnosis of a rotor system based on adaptive local iterative filtering. Transactions of the Institute of Measurement and Control **39**(5), 748–753 (2017)

[22] An, X., Li, C., Zhang, F.: Application of adaptive local iterative filtering and approximate entropy to vibration signal denoising of hydropower unit. Journal of Vibroengineering **18**(7), 4299–4311 (2016)

[23] An, X., Pan, L.: Wind turbine bearing fault diagnosis based on adaptive local iterative filtering and approximate entropy. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **231**(17), 3228–3237 (2017)

[24] An, X., Yang, W., An, X.: Vibration signal analysis of a hydropower unit based on adaptive local iterative filtering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science

231(7), 1339–1353 (2017)

[25] An, X., Zeng, H., Li, C.: Demodulation analysis based on adaptive local iterative filtering for bearing fault diagnosis. Measurement **94**, 554–560 (2016)

[26] Kim, S.J., Zhou, H.: A multiscale computation for highly oscillatory dynamical systems using empirical mode decomposition (emd)–type methods. Multiscale Modeling & Simulation **14**(1), 534–557 (2016)

[27] Li, Y., Wang, X., Liu, Z., Liang, X., Si, S.: The entropy algorithm and its variants in the fault diagnosis of rotating machinery: A review. IEEE Access **6**, 66723–66741 (2018)

[28] Mitiche, I., Morison, G., Nesbitt, A., Hughes-Narborough, M., Stewart, B.G., Boreham, P.: Classification of partial discharge signals by combining adaptive local iterative filtering and entropy features. Sensors **18**(2), 406 (2018)

[29] Piersanti, M., Materassi, M., Cicone, A., Spogli, L., Zhou, H., Ezquer, R.G.: Adaptive local iterative filtering: A promising technique for the analysis of nonstationary signals. Journal of Geophysical Research: Space Physics **123**(1), 1031–1046 (2018)

[30] Sharma, R., Pachori, R.B., Upadhyay, A.: Automatic sleep stages classification based on iterative filtering of electroencephalogram signals. Neural Computing and Applications **28**(10), 2959–2978 (2017)

[31] Yang, D., Wang, B., Cai, G., Wen, J.: Oscillation mode analysis for power grids using adaptive local iterative filter decomposition. International Journal of Electrical Power & Energy Systems **92**, 25–33 (2017)

[32] Cicone, A., Garoni, C., Serra-Capizzano, S.: Spectral and convergence analysis of the discrete alif method. Linear Algebra and its Applications **580**, 62–95 (2019)

[33] Cicone, A., Wu, H.-T.: Convergence analysis of adaptive locally iterative filtering and sift method. arXiv (2005.04578) (2021)

[34] Barbarino, G., Cicone, A.: Conjectures on spectral properties of alif algorithm. Linear Algebra and Its Applications **647**, 127–152 (2022)

[35] Cicone, A., Dell'Acqua, P.: Study of boundary conditions in the iterative filtering method for the decomposition of nonstationary signals. Journal of Computational and Applied Mathematics **373**, 112248 (2020)

[36] Stallone, A., Cicone, A., Materassi, M.: New insights and best practices

for the successful use of empirical mode decomposition, iterative filtering and derived algorithms. Scientific Reports **10**, 15161 (2020)

[37] Wu, H.-T.: Current state of nonlinear-type time-frequency analysis and applications to high-frequency biomedical signals. Current Opinion in Systems Biology **23**, 8–21 (2020)

[38] Barbe, P., Cicone, A., Suet Li, W., Zhou, H.: Time-frequency representation of nonstationary signals: the imfogram. Pure and Applied Functional Analysis **7**(1), 27–39 (2022)