Adaptive Local Iterative Filtering for Signal Decomposition

Antonio Cicone, Jingfang Liu, Haomin Zhou *

September 4, 2013

Abstract

Time-frequency analysis for non-linear and non-stationary signals is extraordinarily challenging. To capture features in these signals, it is necessary for the analysis methods to be local, adaptive and stable. In recent years, decomposition based analysis methods, such as the empiral mode decomposition (EMD) method pioneered by Huang et al., were developed by different researchers groups. These methods decompose a signal into a finite number of components on which the time-frequency analysis can be applied more effectively.

Inspired by the EMD method, and motivated to find local, adaptive and stable decompositions of non-linear and non-stationary signals, we propose an Adaptive Local Iterative Filtering (ALIF) algorithm, which uses iterative filtering strategy to achieve the decompositions. We design smooth filters with compact supports from Fokker-Planck equations. We achieve adaptivity by varying the filter lengths according to the signal itself. The performance and stability of this algorithm is demonstrated by numerical examples. Moreover, in order to have a completely local analysis for non-linear and non-stationary signals, we also propose a definition for the instantaneous frequency which depends exclusively on local properties of a signal.

1 Introduction

Data and signal analysis has become increasingly important these days. Finding features and structures of data is quite challenging especially when the data is generated by a non-linear system and that data is non-stationary. Time-frequency analysis have been substantially studied in the past, we refer to [2] and [5] for more information on this rich subject. Traditionally, Fourier spectral analysis has been commonly used. Another well known approach is based on wavelet transforms. Both of them are effective and easy to implement. However, there are some limitations. Fourier transform works well when systems are linear and the data is periodic or stationary, it can not deal with non-stationary signals or data from non-linear systems. The wavelet transform is also a linear analysis tool. Both use predetermined bases and are not designed with effective data-adaptive properties for nonliear and non-stationary signals.

In the last decade, several decomposition techniques have been proposed to analyze non-linear and non-stationary signals. All these methods share the same approach: first they decompose a signal into simpler components and then apply a time-frequency analysis to each component separately. The signal decomposition can be achieved in two ways: by iteration or by optimization.

The first iterative algorithm, the empirical mode decomposition (EMD), was introduced by Huang et al. [11] in 1998. This method aims to iteratively decompose a signal into a finite series of intrinsic mode functions (IMFs) whose instantaneous frequencies are well behaved. Then based on Hilbert transform, Huang et al. compute the instantaneous frequencies for the IMFs to construct the spectrum for the signal [10, 12, 21]. For a given signal $f(t)$, its Hilbert transform is

$$H(f)(t) = \frac{1}{\pi}\text{p.v.}\int_{-\infty}^{\infty}\frac{f(\tau)}{t-\tau}d\tau \tag{1}$$

*This work was supported by NSF Faculty Early Career Development (CAREER) Award DMS-0645266 and DMS-1042998, and ONR Award N000141310408.

provided this integral exists as a principal value. It is well known that $z(t) = f(t) + iH(f)(t)$ is an analytic function and one can write it as

$$z(t) = f(t) + iH(f)(t) = a(t)e^{i\theta(t)} \tag{2}$$

where $a(t)$, $\theta(t)$ are both real functions and $a(t)$ represents the amplitude and $\theta(t)$ the phase of $z(t)$ respectively. The instantaneous frequency $w(t)$ for the signal $f(t)$ is defined as

$$w(t) = \frac{d\theta(t)}{dt}. \tag{3}$$

However, this definition of instantaneous frequency may be controversial, mainly because it may lead to negative frequencies, which are not meaningful in practice. In order to have well behaved instantaneous frequency, i.e. the instantaneous frequency is ideally in a narrow positive interval, EMD method decomposes the signal into IMFs, whose statistical significance was studied in [22]. In Huang's original paper, an IMF is described to have two properties: 1) the number of extrema and the number of zero crossings must either equal or differ at most by one; and 2) at any point, the mean value of the upper envelope and the lower envelope is zero. The EMD algorithm Huang et al. developed to compute IMFs is called the sifting process. It has an iterative structure, which can be described as follows: Let $\mathcal{L}$ be an operator getting the moving average of a signal $f(t)$ and $\mathcal{S}$ be an operator capturing the fluctuation part $\mathcal{S}(f)(t) = f(t) - \mathcal{L}(f)(t)$. Then the first IMF is given by

$$I_1(t) = \lim_{n\to\infty} \mathcal{S}^n f(t) \tag{4}$$

Here the limit is reached so that applying $\mathcal{S}$ one more time doesn't change the signal. The subsequent IMFs are obtained one after another by

$$I_k(t) = \lim_{n\to\infty} \mathcal{S}^n(f(t) - I_1(t) - ... - I_{k-1}(t)) \tag{5}$$

The process stops when $y(t) = f(t) - I_1(t) - I_2(t) - ... - I_m(t)$ becomes a trend signal, which means it has at most one local maximum or one local minimum. So the decomposition of $f(t)$ is

$$f(t) = \sum_{j=1}^{m} I_j(t) + y(t) \tag{6}$$

In the sifting process, the moving average $\mathcal{L}(f)(t)$ is given by the mean function of the upper envelope and the lower envelope, where the upper and lower envelopes are the cubic splines connecting local maxima and local minima of $f(t)$ respectively. However, this method is not stable under perturbations since cubic splines are used repeatedly in the iteration. To overcome this issue, Huang et al. developed the Ensemble Empirical Mode Decomposition (EEMD) [23] where the IMFs are taken as the mean of many different trials. In each trial, a random perturbation is artificially added to the original signal.

Another iterative decomposition technique is Iterative Filtering (IF) method which is inspired by EMD [13]. It uses the same algorithm framework as the original EMD, but the moving average of a signal $f(n), n \in \mathbb{N}$, is derived by the convolution of $f(n)$ with low pass filters, for example the double average filter $a(k)$ given by

$$a(k) = \frac{m+1-|k|}{(m+1)^2}, \quad k = -m, -(m-1), ..., m-1, m \tag{7}$$

IF is stable under perturbation and the convergence is guaranteed for periodic and l^∞ signals using uniform filters [13, 18]. However, the convergence for general signals with non–uniform filters hasn't been explored yet. Recently, Wang et al. in [16, 17] developed the mode decomposition evolution equations which can achieve similar decompositions to the IF algorithm by using some high order partial differential equations.

A different way to decompose a signal is by optimization. Using the multicomponent amplitude modulation and frequency modulation (AM-FM) representation, which has been studied for example in [14, 19], Hou et al. developed an adaptive data analysis via sparse time-frequency representation (STFR) in [8, 7];

Daubechies et al. proposed synchrosqueezed wavelet transforms (SWT) in [3, 4] as an EMD-like tool. Both methods assume that each IMF can be written as an AM-FM function, a signal $f(t)$ is decomposed into a group of IMFs by seeking the minimizer of some functional of $f(t)$. We refer interested readers to [8, 7, 3, 4] and [20] for more descriptions of those techniques.

In this paper we present a new method, called Adaptive Local Iterative Filtering (ALIF) algorithm, to generalize the IF technique for general oscillatory signals with non-uniform filters. Following the framework of EMD and IF, ALIF algorithm can be split into an inner and outer iteration where the inner iteration is intended to extract a single IMF, while the outer iteration extracts all the IMFs from a signal. There are two main aspects in ALIF that are different from the existing IF algorithms. One is that we use Fokker-Planck equation, a second order partial differential equation (PDE), to construct smooth low pass filters which have compact support. The other is that we adapt the filter length according to the local features of the signal we want to decompose.

To effectively handle non-linear and non-stationary signals, it is highly desirable to use filters with compact support, for the simple reason that filters with long support may mix features that are far apart in a signal. This could be troublesome, especially for signals with transient information. However, the compact support low pass filters, such as the double average filters, used in the existing IF algorithms are not smooth enough. They may create artificial oscillations in subsequent IMFs, due to the non-smoothness. This motivates us to design filters from the solution of certain Fokker-Planck equations, because they are compactly supported, infinitely differentiable and vanishing to zero at both ends smoothly. Thus they can avoid creating any artificial oscillations due to the non-smoothness of the filters. More importantly, to capture the non-stationary changes in the frequency and amplitude of a signal, the length of the filters must be adapted accordingly. In this way, we have a fully adaptive method that produces decompositions for general oscillatory signals with non-uniform filters.

Moreover, we propose an alternative definition for the instantaneous frequency. In the existing instantaneous frequency analysis, Hilbert transform is used to build analytical signals. On the other hand, Hilbert transform is a global transform, which is not ideal to handle signals with transient features. To localize the analysis, we define the instantaneous frequency of an IMF, obtained by the ALIF algorithm, as the rotation speed calculated by the normalized IMF and its derivative. We show that such definition for instantaneous frequency can better capture the frequency changes in non-linear signals.

The rest of the paper is organized as follows: In Section 2, we present the convergence Theorems which we proved in [1] and propose the Adaptive Local Iterative Filtering technique based on them. In Section 3 we give a new definition of instantaneous frequency as well as a method to compute it. In Section 4 we show the numerical results on different kinds of signals.

2 Adaptive Local Iterative Filtering Techniques

In this section we present the Adaptive Local Iterative Filtering (ALIF) algorithm. This method is based on the IF technique proposed in [13]. The main differences are in the way we compute adaptively the filter length and the fact we choose filters from Fokker-Planck equations to compute the moving average of signals. We first present the ALIF Algorithm.

In the proposed algorithm, $w_n(x,t), t \in [-l_n(x), l_n(x)]$, is the filter at point x for the signal $f_n(x)$ with length $2l_n(x)$. And the selection of $w_n(x,t)$ and $l_n(x)$ will be given later in this section. ALIF algorithm contains two iterations: one capturing a single IMF and one producing all the IMFs. We call the former the inner iteration and the latter the outer iteration. The updating step of the inner iteration is

$$f_{n+1}(x) = f_n(x) - \int_{-l_n(x)}^{l_n(x)} f_n(x+t) w_n(x,t) dt \tag{8}$$

To establish the convergence theorem of the ALIF algorithm, we considered an equivalent formulation of the updating step (8 where a scaling function $g_n(x,y) : [-L, L] \to [-l_n(x), l_n(x)]$ is used. The scaling function can be linear such as $g_n(x,y) = l_n(x)y/L$, cubic like $g_n(x,y) = l_n(x)y^3/L^3$ or any other function. By means

ALIF Algorithm IMF = ALIF(f)

IMF = $\emptyset$
while the number of extrema ≥ 2 **do**
  $f_1 = f$
  while the stopping criterion is not satisfied **do**
    compute the filter length $l_n(x)$ for $f_n(x)$
    $f_{n+1}(x) = f_n(x) - \int_{-l_n(x)}^{l_n(x)} f_n(x+t)w_n(x,t)dt$
    $n = n + 1$
  end while
  IMF = IMF $\cup \{f_n\}$
  $f = f - f_n$
end while
IMF = IMF $\cup \{f\}$

of $g_n(x,y)$, we change the RHS in (8) to be

$$\int_{-l_n(x)}^{l_n(x)} f_n(x+t)w_n(x,t)dt = \int_{-L}^{L} f_n(x+g_n(x,y))W(y)dy.$$

Consequently, the equivalent formulation of (8) is given as

$$f_{n+1}(x) = f_n(x) - \int_{-L}^{L} f_n(x+g_n(x,y))W(y)dy \tag{9}$$

where $W(y), y \in [-L, L]$, is a fixed filter. This equivalent formulation is used in the theoretical analysis of the algorithm.

We define a new operator $\mathcal{T}$ as $\mathcal{T}_{w,l}(f) := \int_{-l(x)}^{l(x)} f(x+t)w(x,t)dt$. The convergence theorem for the inner iteration is

Theorem 1. *Let $f(x), x \in \mathbb{R}$, be continuous and $f(x) \in L^\infty(\mathbb{R})$. Let*

$$\varepsilon_n = \frac{\|\mathcal{T}_{w_{n+1},l_{n+1}}(f_{n+1})\|_{L^\infty}}{\|\mathcal{T}_{w_n,l_n}(f_n)\|_{L^\infty}}, \qquad \delta_n = \frac{\|\mathcal{T}_{w_{n+1},l_{n+1}}(|f_{n+1}|)\|_{L^\infty}}{\|\mathcal{T}_{w_n,l_n}(|f_n|)\|_{L^\infty}} \tag{10}$$

If $\prod_{i=1}^n \varepsilon_i \to 0$, $\prod_{i=1}^n \delta_i \to c > 0$, *as* $n \to \infty$, *Then $\{f_n(x)\}$ converges to an IMF.*

The proof of Theorem 1 can be found in [1]. In that paper, we also established a convergence theorem for the outer iteration of the ALIF algorithm.

Let the function $f(x), x \in \mathbb{R}$, be continuous and differentiable and let $f(x)$ have a finite number of extreme points in any compact interval. So $f(x)$ has at most countable extreme points. Let $x_i, i = 1, 2, \ldots, k$, be the extreme points of $f(x)$. Assume that $f(x)$ is strictly monotone in $[x_i, x_{i+1}]$, $i = 1, 2, \ldots, k-1$. Two functions $c_n^{(1)}(x)$ and $c_n^{(2)}(x)$ are defined based on $f_n(x)$ as

$$\begin{aligned} c_n^{(1)}(x) &= \int_{-L}^{L} [f_n'(x) - f_n'(g_n(x,y)+x)]W(y)dy \\ c_n^{(2)}(x) &= \int_{-L}^{L} [f_n'(x) - f_n'(g_n(x,y)+x)]h(y)W(y)dy \end{aligned} \tag{11}$$

The convergence theorem for the outer iteration of ALIF algorithm is

Theorem 2. *Assume that $f(x)$ is a differentiable function with properties described above. Using the equivalent formulation (9), if the scaling function is separable, i.e. $g_n(x, y) = l_n(x)h(y)$ and for every $n \in \mathbb{N}$*

$$\begin{aligned} c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) > 0 \quad \text{when} \quad f_n'(x) > 0 \\ c_n^{(1)}(x) + l_n'(x)c_n^{(2)}(x) < 0 \quad \text{when} \quad f_n'(x) < 0 \end{aligned} \tag{12}$$

Then the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is at most the number of extreme points of $f(x)$ if $\lim_{n\to\infty} f_n(x)$ exists.

In Theorem 2, if the number of extreme points of $f(x) - \lim_{n\to\infty} f_n(x)$ is less than the number of extreme points of $f(x)$ in the next step of the outer iteration, then the function $f(x) - \lim_{n\to\infty} f_n(x)$ is smoother than $f(x)$. If this property keeps true for each outer iteration, i.e. the number of extreme points in the remaining signal keeps decreasing, the ALIF algorithm converges for the signal $f(x)$.

2.1 Filter Lengths Computation

A crucial step in ALIF is the computation of the filter length $l_n(x)$ for a signal $f_n(x)$. The function $l_n(x)$ has to be strictly positive. It can be either a constant for every x in the domain of $f_n(x)$, this is the so called *uniform mask length* case, or $l_n(x)$ can change point by point, which is the *non-uniform mask length* case.

In the uniform mask length case $l_n(x)$ has to be a strictly positive constant function. Following [13] we compute it as

$$l_n(x) := \left\lfloor \frac{2N}{k} \right\rfloor \tag{13}$$

where N is the total number of sample points of a signal $f_n(x)$ and k is the number of its extreme points. When $l_n(x)$ is a constant function, the moving average of $f_n(x)$ is simply a convolution of $f_n(x)$ and $w_n(x)$ where $w_n(x)$ is a fixed filter with length $2l_n(x)$. The convergence of the fixed filter is shown in [13, 9, 18] and [1].

However, in this paper, we focus on the general situation of non-uniform mask length where $l_n(x)$ is a strictly positive function not necessarily constant. In practice, there is more than one way to determine $l_n(x)$. In [13], Wang et al. proposed an interpolation method to compute the filter length. Assuming that the signal $f_n(x)$ has k local extreme points, let x_i denote the position of the ith local extreme point of $f_n(x)$, where $i = 1, 4, \ldots, k$, then the filter length $l_n(x_i)$ at any x_i, $i = 2, 4, \ldots, k-1$, is given by

$$l_n(x_i) = x_{i+1} - x_{i-1} \tag{14}$$

Once the filter lengths at the extreme points are determined, the value $l_n(x)$ for any other point x is given by interpolations using the known filter length pairs $(x_i, l_n(x_i))$.

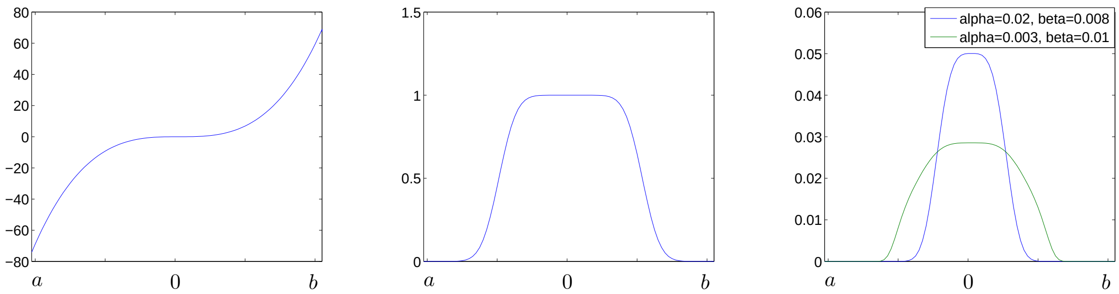
2.2 Local Filters developed from a PDE model

Another important aspect of ALIF is the low pass filters used in the inner iteration. To handle non-linear and non-stationary signals, we want to design the filters to be smooth with compact support. To produce such a filter, we make use of the diffusion process. The idea is very natural: when applying the diffusion to the data, the oscillations will eventually be eliminated and a curve will be generated as the average.

It is well known that diffusion processes are associated with PDEs. For a given diffusion equation, we can get its fundamental solution and treat it as the filter in the iterative filtering algorithm. Here we note that the well known heat equation may not be a good choice, because its solution leads to a filter with infinite support. This is not desirable. To have compact support and smoothness for the filters, we select Fokker-Planck equations to construct our filters. Let us consider the Fokker-Planck equation

$$p_t = -\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx}, \quad \alpha,\ \beta > 0 \tag{15}$$

Assume $h(x)$ and $g(x)$ are smooth enough functions such that there exist $a < 0 < b$ satisfying:



(a) $h(x)$ (b) $g^2(x)$ (c) $p(x)$

Figure 1: Coefficient functions and steady states of (17). (a) $h(x)$ is an odd function like x^3. (b) $g^2(x)$ is an even function, for instance a smooth approximation of the step function. (c) two steady states for coefficients $\alpha = 0.02$, $\beta = 0.008$ and $\alpha = 0.003$, $\beta = 0.01$ respectively.

- $g(a) = g(b) = 0, \quad g(x) > 0$ for $x \in (a, b)$
- $h(a) < 0 < h(b)$

The $(g^2(x)p)_{xx}$ term generates diffusion effect and pulls out the solution p from the center of (a, b) towards a and b while the $-(h(x)p)_x$ term transports it from a and b towards the center of the interval (a, b). When the two forces are balanced, the steady state is achieved. There exists a smooth non trivial solution $p(x)$ of the stationary problem:

$$-\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx} = 0 \tag{16}$$

satisfying $p(x) \geq 0$ for $x \in (a, b)$, and $p(x) = 0$ for $x \notin (a, b)$. That means the solution is concentrated in the interval $[a, b]$ and there is no leakage outside. So $p(x)$ is a local filter satisfying our requirements.

Based on this analysis, we shall use the solution to the initial value problem

$$\begin{aligned} p_t &= -\alpha(h(x)p)_x + \beta(g^2(x)p)_{xx} \\ p(x, 0) &= \delta\left(x - \frac{a+b}{2}\right) \end{aligned} \tag{17}$$

where $\delta(x)$ is the Dirac delta function, as the filter in our decomposition algorithm. By adjusting the functions $h(x), g(x)$ as well as coefficients α, β, we can get different shapes for the filter. In Figure 1, we plot two steady states for two different α, β, fixed $h(x)$ and $g(x)$ functions.

We can see that the larger α is, the more the weight is concentrated in the center; on the other hand, the larger β is, the more the weight is diffused and less concentrated in the center. When designing the local filter based on the Fokker-Planck equation, we first fix functions $h(x)$ and $g(x)$ and then adjust the coefficients α and β to get the filter shape we want.

Another issue is on the design of filters with same shape for different support lengths. There are at least two approaches available. One way is to solve the Fokker-Planck equation again with $h(x)$ and $g(x)$ scaled in x for every different a or b. Assume we get the steady state of (17), in order to get the filter with length from $\hat{a}$ to $\hat{b}$, we solve (18) and the steady state is the filter we want.

$$\begin{aligned} p_t = &-\alpha\left(h\left(\hat{a}\frac{x-(a+b)/2}{a} + \frac{\hat{a}+\hat{b}}{2}\right)p\right)_x \\ &+ \beta\left(g^2\left(\hat{a}\frac{x-(a+b)/2}{a} + \frac{\hat{a}+\hat{b}}{2}\right)p\right)_{xx} \\ p(x, 0) = &\ \delta\left(x - \frac{\hat{a}+\hat{b}}{2}\right) \end{aligned} \tag{18}$$

The other way is to solve the Fokker-Planck equation for a fixed a and b only once and take a special interpolation of the steady state to get the filter with the desired length.

Given the numerical solution $p(x)$ for the steady state of (17) at discrete points $\{x_1, x_2, ..., x_{2s+2}\}$, where s is a large natural number and $x_1 = a$, $x_{2s+2} = b$, the value p_i, $i = 1, 2, ..., 2s + 1$, represents the weight of the filter in the interval $[x_i, x_{i+1})$ and the sum of weights in all the intervals equals 1. Assume we need to compute again the numerical solution for the steady state of (17), but this time for an interval $[\hat{a}, \hat{b}]$ with a different discretization $\{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{2n+2}\}$, where we assume that $n < s$ and $\hat{x}_1 = \hat{a}$, $\hat{x}_{2n+2} = \hat{b}$. We call this new numerical solution $w(x)$. The idea is to make use of the previously computed values p_i to approximate the new numerical solution $w(x)$. We start mapping the interval $[\hat{a}, \hat{b}]$ to the interval $[a, b]$ by linear scaling. As a result, the points $\{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{2n+2}\}$ are mapped into $\{y_1, y_2, ..., y_{2n+2}\}$ with $y_1 = a$, $y_{2n+2} = b$. The values of the filter $w_j, j = 1, 2, ..., 2n + 1$, in the interval $[\hat{a}, \hat{b}]$ will be the weights in the intervals $[y_j, y_{j+1}) \subseteq [a, b]$, with $j = 1, 2, ..., 2n + 1$, where the weight w_j in the interval $[y_j, y_{j+1})$ is given by

$$w_j = \int_{y_j}^{y_{j+1}} p(x)dx \tag{19}$$

The integral can be approximated by the Riemann sum based on the discrete points x_i and the weights p_i, $i = 1, 2, ..., 2s + 1$. If y_j falls between two points x_{m_1-1}, x_{m_1} and y_{j+1} falls between two points x_{m_2}, x_{m_2+1}, then the filter weight w_j will be

$$w_j = p_{m_1-1}(x_{m_1} - y_j) + \sum_{i=m_1}^{m_2-1} p_i + p_{m_2}(y_{j+1} - x_{m_2}) \tag{20}$$

Using this special interpolation method, the shapes of filters with different lengths are the same. Moreover, the filter length can be any positive real number, also a non integer one.

In summary, the former approach generates the solution of the PDE. However solving PDEs numerically is much slower than interpolating existing filters. The latter approach is easier to implement. In our simulations, we use the interpolation strategy to get filters with same shape with different lengths.

3 A Different View of Instantaneous frequency

The instantaneous frequency in EMD is defined based on Hilbert transform. However, the Hilbert transform is a global operator, which is not ideal for a local time-frequency analysis. In this section, we present a new definition of instantaneous frequency which uses only local information.

Before we move on to the new definition of the instantaneous frequency, we first review the behavior of a system of second ordinary differential equations (ODEs) in the polar coordinates, where the rotation speed is corresponding to one equation in the system. Consider the linear system of ODEs $(\dot{x} = \cos t, \dot{y} = -\sin t)$. By changing the coordinate from (x, y) to (r, θ) by $x = r\cos\theta$, $y = -r\sin\theta$, this linear system can be written as $(\dot{\theta} = 1, \dot{r} = 0)$. The rotation speed of this ODE system is given directly as $\dot{\theta}$ in the first equation. As $\dot{\theta}$ is a constant, the rotation speed is a constant, which is consistent with the previous analysis. Let us consider also the case of a non-linear ODE such as the van der Pol oscillator given as $\ddot{x} + \alpha(x^2 - 1)\dot{x} + x = 0, \alpha > 0$. The standard first-order form of it is $(\dot{x} = y, \dot{y} = -x - \alpha(x^2 - 1)y)$. Using polar coordinates $x = r\cos\theta$, $y = -r\sin\theta$, this non-linear system can be written as $(\dot{r} = -\alpha(r^2\cos^2\theta - 1)r\sin^2\theta, \dot{\theta} = 1 + \alpha(r^2\cos^2\theta - 1)r\sin\theta\cos\theta)$. The second equation gives the rotation speed $\dot{\theta}$ which is not a constant any more. When $\alpha << 1$, $\dot{\theta}$ is positive.

Let $f(t)$ be an IMF that represents some pattern of oscillations. We treat this $f(t)$ as the x coordinate of some second order ODE, we can get the frequency naturally as the derivative of the phase angle in the polar coordinate $\dot{\theta}$. It is not necessary to derive the corresponding ordinary differential equation, instead we can compute the rotation speed directly. The procedure contains two steps: first $f(t)$ is mapped to $\theta(t)$ in the polar coordinate and second the rotation speed $\dot{\theta}$ is computed. We mainly to deal with step one since the second step is just to take derivatives. When mapping $f(t)$ to $\theta(t)$, we should get rid of the impact of r since

r and θ are independent in the polar coordinate. The way we do it is to normalize both $f(t)$ and $f'(t)$ by their envelopes. Thus after normalization, $(f(t), f'(t))$ is the unit circle or a perturbation of the unit circle if the normalization is not perfect. In the latter case, although the perturbation is not a perfect unit circle, we can still see its rotation standing at the center. Thus the rotation speed $\dot{\theta}$ is a positive function.

Based on these observations, we propose a new local definition of the instantaneous frequency as the rotation speed of the phase angle.

Let $f(t)$ be a function satisfying the IMF requirements, there exists an envelope function $q(t)$ of $f(t)$ such that

$$F_1(t) := f(t)/q(t) \in [-1, 1] \tag{21}$$

Considering the derivative of $f(t)$, there exists an envelope function $r(t)$ of $f'(t)$ such that

$$F_2(t) := f'(t)/r(t) \in [-1, 1] \tag{22}$$

Functions $q(t)$ and $r(t)$ can be simply taken as the cubic splines connecting the local extrema in $f(t)$ and $f'(t)$ respectively. If we define

$$F(t) = F_1(t) + iF_2(t) \tag{23}$$

then $F(t)$ corresponds to a curve in $[-1, 1] \times [-1, 1]$ on the complex plane. $F(t)$ is a perturbation of the unit circle and we define the angle for the rotation of $F(t)$ as

$$\theta(t) = -\arctan \frac{F_2(t)}{F_1(t)} \tag{24}$$

and the instantaneous frequency for $f(t)$ as

$$w(t) = \frac{d\theta(t)}{dt} \tag{25}$$

The idea here is to map an IMF and its derivative into a perturbation of the unit circle first; then the instantaneous frequency is defined as the rotation speed measured on the perturbation of the unit circle.

If the magnitude of the IMF $f(t)$ or its derivative change significantly in a short time, the envelopes $q(t)$ and $r(t)$ used in (21) and (22) to normalize $f(t)$ and $f'(t)$, respectively, might not follow closely the changes of the IMF and its derivative. This may cause unexpected errors in the instantaneous frequency. So it is advisable to identify and properly handle these sudden changes when constructing the envelopes. To address this problem we can make use of the essentially non-oscillatory (ENO) technique for shock capturing developed in computational fluid dynamics [6, 15].

The ENO technique is a divide et impera method. First, based on the differences of consecutive extrema of a given signal, we detect possible sudden changes in the magnitude using a preselected threshold. If a sudden change is detected between two consecutive extreme points we compute the differences of consecutive sample points in between those extrema. The sudden change is assumed to happen where the left difference differs most from the right difference. We use this point to divide the signal into two parts and to construct two separate envelopes one for the left and one for the right hand side of the signal.

Let us consider two test examples of the computation of the instantaneous frequency using both the new definition and the definition based on Hilbert transform.

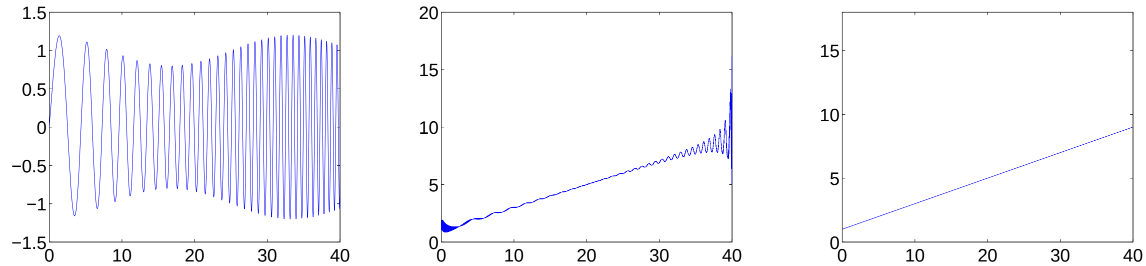
Test 1 In Figure 2, the signal is given by

$$f(t) = (1 + 0.2\cos(0.06\pi t))\sin[(1 + 0.1t)t], \quad t \in [0, 40] \tag{26}$$

The amplitude of the signal changes slowly, both frequency analysis methods show the gradual change in the instantaneous frequency as shown in Figure 2. However, when there is a significant change in the amplitude of the signal, instantaneous frequencies defined by two approaches are different.

Test 2 In Figure 3 the signal is generated by

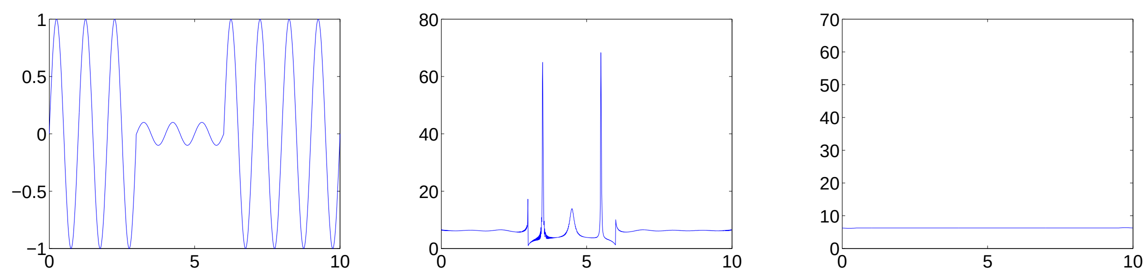
$$f(t) = a(t)\sin(2\pi t), \text{where} \quad a(t) = 1 - 0.9\chi_{[3,6]}, \quad t \in [0, 10] \tag{27}$$



(a) (b) (c)

Figure 2: Test 1. (a) the signal defined in (26). The oscillation gradually becomes faster and the amplitude changes mildly. (b) instantaneous frequency computed using Hilbert transform. It shows the gradual change in the instantaneous frequency but has some oscillations. (c) instantaneous frequency computed using the proposed method. It also shows the gradual change in the instantaneous frequency and almost has no oscillations.

It has a sudden change in the amplitude at time 3 and 6. Except these two positions, the signal is a constant frequency signal. We expect the time frequency analysis to return an instantaneous frequency which is almost a constant. Using the instantaneous frequency definition based on Hilbert transform, the transitory change in the amplitude affects faraway positions and this leads to strange behaviors in the instantaneous frequency. On the other hand, using the proposed method together with the ENO technique we get an instantaneous frequency which is almost a constant.



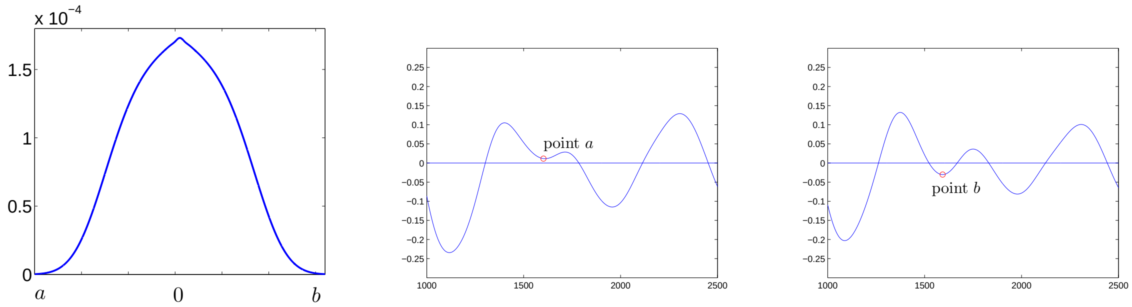
(a) (b) (c)

Figure 3: Test 2. (a) the signal defined in (27). There are sudden changes in the amplitude. (b) instantaneous frequency computed using Hilbert transform; the instantaneous frequency is inconsistent with people's expectation. (c) instantaneous frequency computed using the proposed method. The instantaneous frequency is almost a constant function.

4 Numerical Experiments

The ALIF algorithm is given in Section 2, where we propose also an adaptive technique to compute the filter length $l_n(x)$. We use the designed filter given in Figure 4a which is based on the PDE model described in Section 2.2, where h(x), g(x) are the functions shown in Figure 1 and the coefficients are $\alpha = 0.005$, $\beta = 0.09$. Filters with different lengths are computed using the special interpolation method introduced in Section 2.2.

For the stopping criterion, we define $I_{k,n} = \mathcal{S}_k^n(x - I_1 - ... - I_{k-1})$ where $\mathcal{S}_k$ denotes the operator $\mathcal{S}$ used to obtain the kth IMF. $\mathcal{S}$ captures the fluctuation part of a signal as described in the introduction Section.



(a) (b) (c)

Figure 4: (a) the local filter we use in the numerical implementations. (b) After 3 iterations the local minimum point a is above 0. (c) After 5 iterations the local minimum point b is below 0. Other parts do not change significantly from (b) to (c).

We define then

$$SD := \frac{\|I_{k,n} - I_{k,n-1}\|_2}{\|I_{k,n-1}\|_2} \tag{28}$$

So we can either stop the process when the value SD reaches a certain threshold as suggested in [11] and [13] or we can put a limit on the maximal number of iterations.

It is also possible to adopt different stopping criteria for different IMFs. Considering for instance a noisy signal, we may use a loose stopping criterion for the first few IMFs to reduce the number of noise components and use a more restrictive stopping criterion for the remaining IMFs to detect finer differences in the patterns of the following components. In the implementation, we use as stopping criterion $SD \leq \delta$ where δ is usually set to be 0.08.

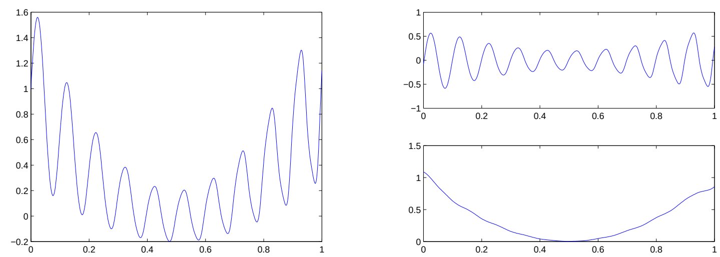
As observed previously, a good decomposition method should capture all the finest oscillations around a moving average. That means the IMFs should satisfy at least this condition: all the local maximal values are positive and all the local minimal values are negative, as shown in Figure 4c. Using an iterative filtering method and tuning the stopping criterion we can get an IMF that looks like Figure 4b to be like Figure 4c.

We test the proposed ALIF algorithm on both artificial signals and real-life data sets. Among the proposed examples, the first four are artificial signals and the last two are real-life signals. We also show that the ALIF algorithm is stable under perturbations for both simulated and real-life signals.

Example 1 We test the ALIF algorithm on the non-stationary frequency modulated signal

$$f(t) = 4(t-0.5)^2 + (2(t-0.5)^2 + 0.2)\sin\left((20\pi + 0.2\cos(40\pi t))t\right), t \in [0,1] \tag{29}$$

From Figure 5, we see that the ALIF algorithm decompose $f(t)$ into two components. The first is the frequency modulated signal $(2(t-0.5)^2 + 0.2)\sin\left((20\pi + 0.2\cos(40\pi t))t\right)$ and the second is the so called trend $4(t-0.5)^2$.



(a) (b)

Figure 5: (a) the signal given in (29); (b) the two components in the decomposition.

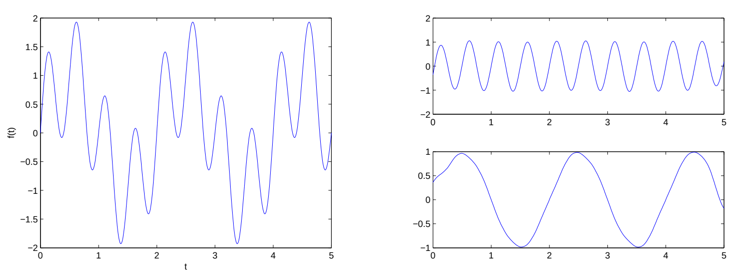
Example 2 We study a signal and its perturbation with white noise to demonstrate the stability of the proposed algorithm. The signals are

$$f_1(t) = \sin \pi t + \sin 4\pi t, t \in [0, 5] \tag{30}$$

$$f_2(t) = \sin \pi t + \sin 4\pi t + n_2(t), t \in [0, 5] \tag{31}$$

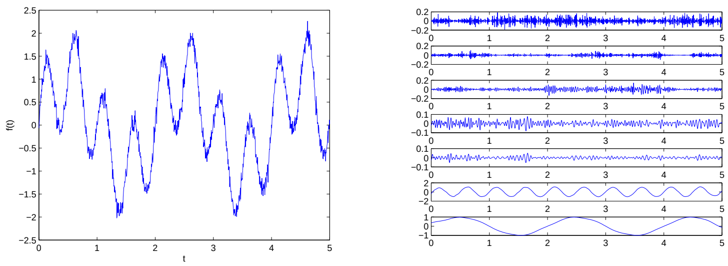
$$f_3(t) = \sin \pi t + \sin 4\pi t + n_3(t), t \in [0, 5] \tag{32}$$

where $n_2(t)$ and $n_3(t)$ are white noise: $n_2(t) \sim N(0, 0.01)$ and $n_3(t) \sim N(0, 1)$ for each t. We apply the ALIF algorithm on $f_1(t)$, $f_2(t)$ and $f_3(t)$. From Figure 6, we see that $f_1(t)$ is separated into two IMFs, which correspond to the components $\sin 4\pi t$ and $\sin \pi t$ respectively. $f_2(t)$ is decomposed into seven IMFs as shown in Figure 7. The first few IMFs come from the impact of noise and the last two IMFs reveal the two sinusoidal functions $\sin(4\pi t)$ and $\sin(\pi t)$. $f_3(t)$ is decomposed into nine IMFs and only the last seven are shown in Figure 8. Similar to the result of $f_2(t)$, the first few IMFs come from the impact of noise and the last two IMFs reveal the two sinusoidal functions $\sin(4\pi t)$ and $\sin(\pi t)$. Via this example, the ALIF algorithm is shown to be robust to white noise. What is more important, denoising is automatically achieved by getting rid of the first few IMFs.



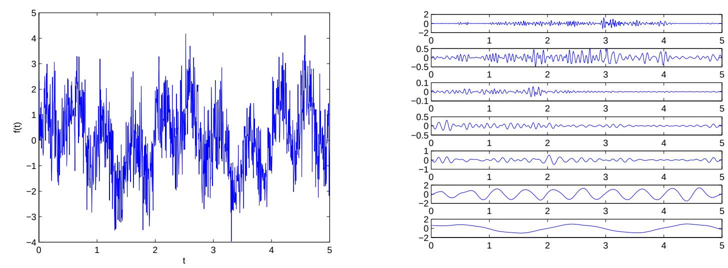
(a) (b)

Figure 6: (a) the signal $f_1(t)$ given in (30); (b) the components in the decomposition.



(a) (b)

Figure 7: (a) the signal $f_2(t)$ given in (31); (b) the components in the decomposition.



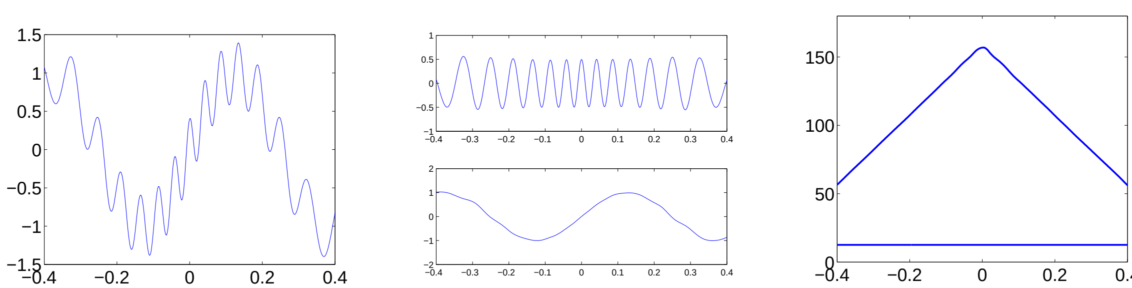
(a) (b)

Figure 8: (a) the signal $f_3(t)$ given in (32); (b) the last seven among nine components in the decomposition.

Example 3 We test the ALIF algorithm on the highly non-stationary signal

$$f(t) = \sin(4\pi t) + 0.5\cos(50\pi|t| - 40\pi t^2), \quad t \in [-0.4, 0.4] \tag{33}$$

After the decomposition, we compute the instantaneous frequency of each component by the method proposed in Section 3. As shown in Figure 9, $f(t)$ is separated into two IMFs. One has a varying instantaneous frequency and the other has a constant instantaneous frequency.



(a) (b) (c)

Figure 9: (a) the signal given in (33); (b) the components obtained from the decomposition; (c) the instantaneous frequency for the two components in (b).

Example 4 When there are sudden changes or jumps in the amplitude of a signal the computation of the moving average can be compromised in positions nearby the sudden changes. To avoid such negative influence we can make use of the essentially non-oscillatory (ENO) technique for shock capturing developed in computational fluid dynamics [6, 15]. The idea is to use only one side information near this sudden change in the ALIF algorithm. We achieve this in two steps. In the first step we detect positions of jumps in a signal and we treat the left and right sides of a jump as two signals. In the second step, for the left signal, we extend the right boundary of the signal so that the moving average can be computed; for the right one, we extend the left boundary of it. The moving average of the whole signal is the catenation of the moving averages from the left side and the right side.

In particular to detect the positions of possible jumps, we compute the variation of the signal at two consecutive sample points. If the value of the variation is sufficiently larger than an average variation, a jump is regarded to exist in between the two sample points. Then the signal $f(t)$ is separated into two parts by splitting at the detected jump: the left part $f_L(t)$ and the right part $f_R(t)$. Assume we use uniform filters with filter length $2m$ and assume the position of the jump is t_0. The left part $f_L(t)$ only provides complete data to compute the moving average $\mathcal{L}(f)(t)$ up to $t_0 - m$. To compute the moving average for $t \in [t_0 - m, t_0]$, the right boundary of $\mathcal{L}(f_L)(t)$ need to be extended properly. One possible way to do so is based on the least square fitting. For the ordered set of points

$$\{(t_0 - 2m, \mathcal{L}(f)(t_0 - 2m)), (t_0 - 2m + 1, \mathcal{L}(f)(t_0 - 2m + 1)), ..., (t_0 - m, \mathcal{L}(f)(t_0 - m))\},$$

we could find a linear function as the least squares solution. We compute the values of this linear function at $t = t_0 - m + 1, t_0 - m + 2, ..., t_0$ and extend the moving average $\mathcal{L}(f_L)(x)$ for $t \in [t_0 - m, t_0]$. We apply the same procedure properly modified to compute the moving average $\mathcal{L}(f_R(t))(t)$ for $t \in [t_0, t_0 + m]$. Finally we catenate $\mathcal{L}(f_L)(x)$ and $\mathcal{L}(f_R(t))(t)$ to produce the moving average $\mathcal{L}(f)(t)$.

As an example we consider the following signal

$$x(t) = f(t) + 0.3\sin(4\pi t) + n(t), t \in [0, 10], \tag{34}$$

where $n(t) \sim N(0, 0.1^2),\ t \in R$, and $f(t)$ is the step function

$$f(t) = \begin{cases} -0.5 & \text{if} \quad 0 \le t < 3.5 \ \text{ or } \ 6.75 < t \le 10 \\ 0.5 & \text{if} \quad 3.5 \le t \le 6.75 \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Using ALIF algorithm with uniform filters, we get the decomposition of $x(t)$ shown in Figure 11. The first 4 components are the reflections of the white noise $n(t)$. The 5th component is the IMF corresponding to the sinusoidal function $0.3\sin(4\pi t)$. The subsequent components are somehow related to the step function $f(t)$. The 7th and the 8th components are mainly related to the sudden changes in the magnitude of $x(t)$ since they have obvious larger amplitude at the same positions where the step function $f(t)$ jumps. The 9th components looks like a smoothed version of the step function $f(t)$. It has the same trend as the step function but loses the transient jumping behaviour. The result is like this since we use the signals behaviours from both sides near the jumps in ALIF algorithm although the signal are severely non-stationary from one side of the jump to the other side.

Using ALIF algorithm with an ENO approach, we get the decomposition of $x(t)$ shown in Figure 12. Similarly, the first four components are derived due to the impact of noise. The 5th component is corresponding to the sinusoidal function $0.3\sin(4\pi t)$. The subsequent components are related to the step function $f(t)$, where the last component is much more close to a step function compared with the last component in Figure 11.

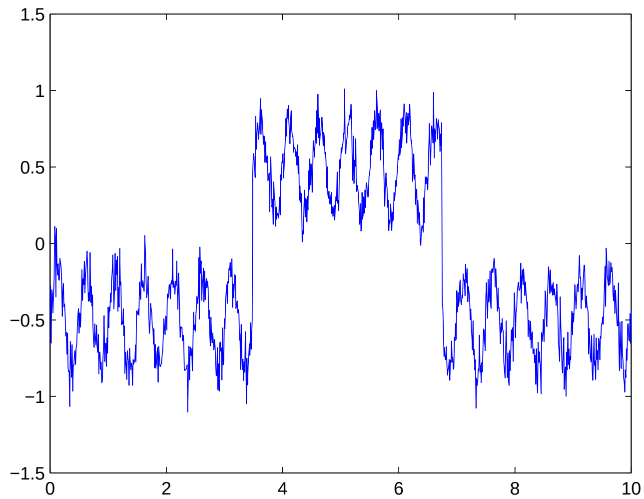


Figure 10: The signal given by (34). There are two sudden changes in the amplitude of this signal.

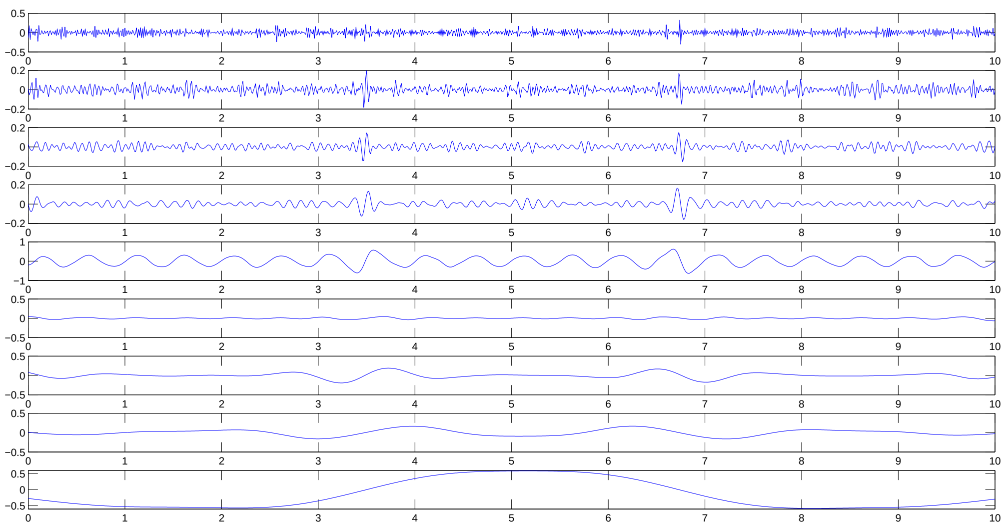


Figure 11: The decomposition of the signal (34) by IF algorithm. The last component lose the representation of the step function.

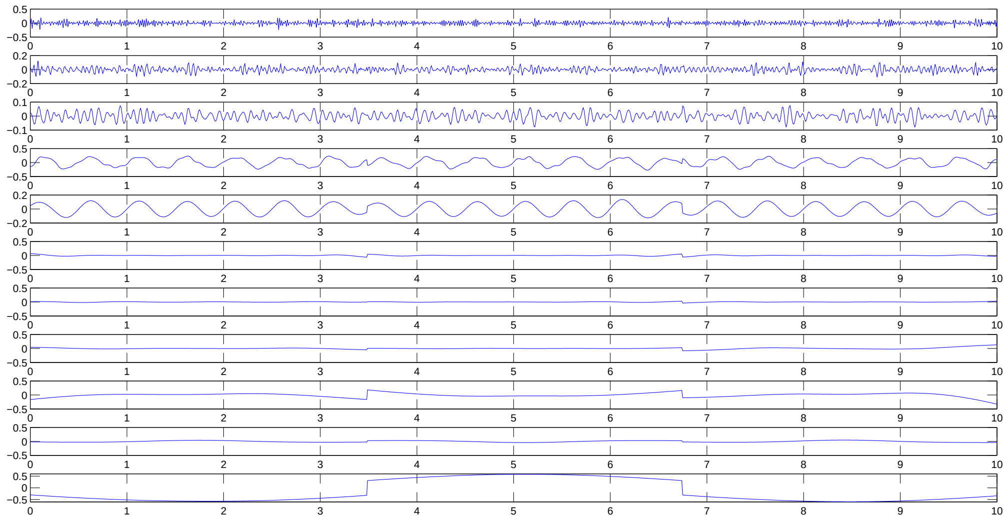
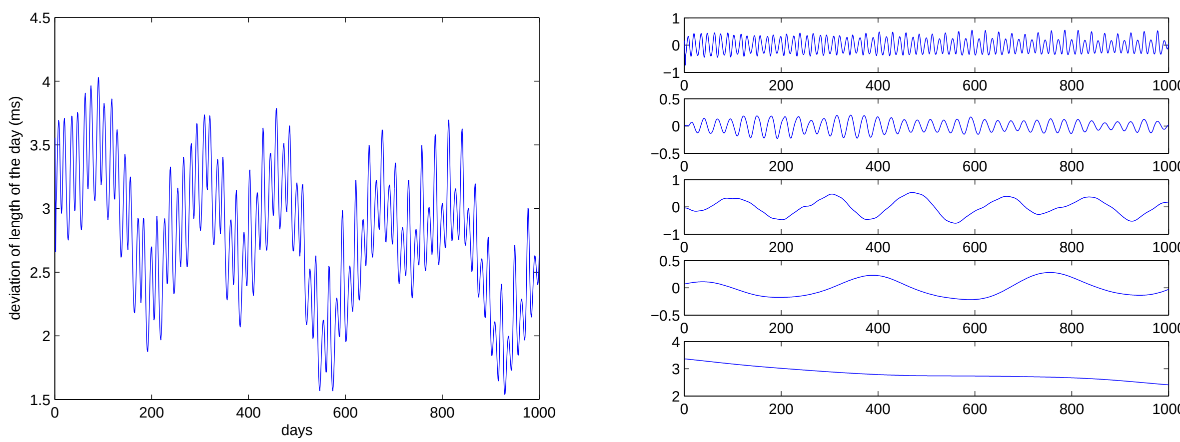


Figure 12: The decomposition of the signal (34) by IF algorithm using only one side information when there are sudden changes in the amplitude. The last component reveals the step function.

Example 5 We apply the ALIF algorithm on some real world data. The first one is the deviation of the length of the day (LOD) data[1] for 1000 days from 01/01/1973 to 09/28/1976. This data is decomposed into 5 components as shown in Figure 13 where 4 of them are IMFs and the last one is the trend. From the four IMFs, we can see very regular patterns: the half monthly change pattern, the monthly change pattern, the half yearly change pattern as well as the yearly change pattern.

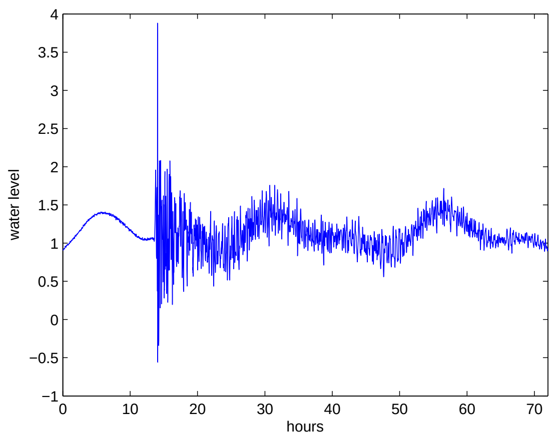


(a) (b)

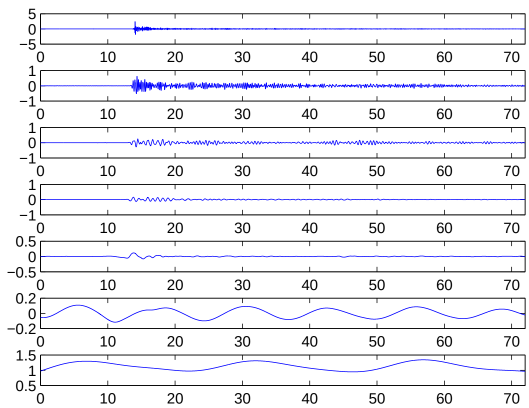
Figure 13: Length of day (LOD) signal and its decomposition. (a) the LOD signal; (b) the 5 components in the decomposition.

[1] LOD data set `http://hpiers.obspm.fr/eoppc/eop/eopc04/eopc04.62-now`

Example 6 We apply the ALIF algorithm to the water level data[2] observed at Kawaihae, Hawaii, HI for 72 hours from March 11, 2011 to March 13, 2011 when the 2011 Honshu earthquake and tsunami occurred. The data is decomposed into several components where the first three IMFs represent the impact of the tsunami and the last two components reveal the basic wave height. From Figure 14 we see that the algorithm captures transient signals as well as the regular patterns.



(a)



(b)

Figure 14: (a) the given wave height signal; (b) the 7 components in the decomposition.

5 Conclusion

In this paper we propose the Adaptive Local Iterative Filtering (ALIF) algorithm with the purpose of designing a local, adaptive and stable iterative filtering method.

The adaptivity of the ALIF algorithm is achieved by means of a filter length who is changed pointwise. The locality is guaranteed by the local filter we design based on a PDE model. The stability of the ALIF algorithm is shown in Section 4. We also present a new definition of instantaneous frequency which makes use only of local properties of a given signal, allowing for a completely local time-frequency analysis.

References

[1] A. Cicone, J. Liu, and H. M. Zhou. The convergence of iterative filtering methods.

[2] L. Cohen. *Time-frequency analysis: theory and applications*. Prentice-Hall, Inc., 1995.

[3] I. Daubechies, J. Lu, and H.-T. Wu. Synchrosqueezed wavelet transforms: A tool for empirical mode decomposition. *arXiv preprint arXiv:0912.2437*, 2009.

[4] I. Daubechies, J. Lu, and H.-T. Wu. Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis*, 30(2):243–261, 2011.

[5] K. Gröchenig. *Foundations of time-frequency analysis*. Birkhäuser Boston, 2000.

[6] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, 71(2):231–303, 1987.

[7] T.Y. Hou and Z. Shi. Adaptive data analysis via sparse time-frequency representation. *Adv. in Adap. Data Anal.*, 3(1):1–28, 2011.

[2] Honshu earthquake and tsunami data set `http://oldwcatwc.arh.noaa.gov/previous.events/03-11-11_Honshu/index.php`

[8] T.Y. Hou, M.P. Yan, and Z. Wu. A variant of the emd method for multi-scale data. *Adv. in Adap. Data Anal.*, 1(4):483–516, 2009.

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

[10] N.E. Huang, Z. Shen, and S.R. Long. A new view of nonlinear water waves: The hilbert spectrum 1. *Annual Review of Fluid Mechanics*, 31(1):417–457, 1999.

[11] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.C. Yen, C.C. Tung, and H.H. Liu. 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, 1998.

[12] N.E. Huang, Z. Wu, S.R. Long, K.C. Arnold, X. Chen, and K. Blank. On instantaneous frequency. *Adv. Adapt. Data Anal*, 1(2):177–229, 2009.

[13] L. Lin, Y. Wang, and H. Zhou. Iterative filtering as an alternative algorithm for empirical mode decomposition. *Advances in Adaptive Data Analysis*, 1(4):543–560, 2009.

[14] P.J. Loughlin and B. Tacer. Comments on the interpretation of instantaneous frequency. *Signal Processing Letters, IEEE*, 4(5):123–125, 1997.

[15] Chi-Wang Shu. High order eno and weno schemes for computational fluid dynamics. In *High-order methods for computational physics*, pages 439–582. Springer, 1999.

[16] Y. Wang, G.-W. Wei, and S. Yang. Iterative filtering decomposition based on local spectral evolution kernel. *Journal of scientific computing*, 50(3):629–664, 2012.

[17] Y. Wang, G.-W. Wei, and S. Yang. Mode decomposition evolution equations. *Journal of scientific computing*, 50(3):495–518, 2012.

[18] Y. Wang and Z. Zhou. On the convergence of iterative filtering empirical mode decomposition. In *Excursions in Harmonic Analysis, Volume 2*, pages 157–172. Springer, 2013.

[19] D. Wei and A.C. Bovik. On the instantaneous frequencies of multicomponent am-fm signals. *Signal Processing Letters, IEEE*, 5(4):84–86, 1998.

[20] H.-T. Wu, P. Flandrin, and I. Daubechies. One or two frequencies? the synchrosqueezing answers. *Advances in Adaptive Data Analysis*, 3(01n02):29–39, 2011.

[21] Z. Wu and N.E. Huang. A study of the characteristics of white noise using the empirical mode decomposition method. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2046):1597, 2004.

[22] Z. Wu and N.E. Huang. Statistical significance test of intrinsic mode functions. *Hilbert–Huang Transform and Its Applications*, pages 107–127, 2005.

[23] Z. Wu and N.E. Huang. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. in Adap. Data Anal.*, 1(1):1–41, 2009.