Data Driven Time-Frequency Analysis

Thomas Y. Hou* Zuoqiang Shi†

October 8, 2011

Abstract

In this paper, we introduce a new adaptive data analysis method based on the sparse representation over a highly redundant dictionary. This method is inspired by the Empirical Mode Decomposition method (EMD) and the recently developed compressed (compressive) sensing theory, and can be applied to nonlinear and non-stationary data. The main idea is to look for the sparsest representation of multiscale data within the largest possible dictionary consisting of intrinsic mode functions of the form $\{a(t)\cos(\theta(t))\}$, where $a \in V(\theta)$, $V(\theta)$ consists of the functions smoother than $\cos(\theta(t))$ and $\theta' \geq 0$. This problem can be formulated as a nonlinear L^0 optimization problem. In order to solve this optimization problem, we propose a nonlinear matching pursuit method by generalizing the classical matching pursuit for the L^0 optimization problem. One important advantage of this nonlinear matching pursuit method is it can be implemented by Fast Fourier Transform with complexity of order $O(N \log N)$, where N is the number of data sample points. Further, this approach is very stable to noise. Extensive numerical examples will be given to demonstrate the robustness of our method and comparison will be made with the EMD/EEMD method. We also introduce another two algorithms to deal with the 'end effect'. Further, we provide a convergence analysis of our nonlinear matching pursuit method under certain scale separation assumptions. One advantage of performing such decomposition is to preserve some intrinsic physical properties of the data, such as trend and instantaneous frequency.

1 Introduction

Developing a truly adaptive data analysis method is important for our understanding of many natural phenomena. Traditional data analysis methods, such as the Fourier transform and wavelet transform, use pre-determined basis and provide an effective tool to process linear and stationary data. However, there are still some limitations in applying these methods to analyze nonlinear and nonstationary data. Time-frequency analysis has been developed to overcome the limitations of the traditional techniques by representing a signal

*Applied and Comput. Math, Caltech, Pasadena, CA 91125. *Email: hou@cms.caltech.edu.*

†Applied and Comput. Math, Caltech, Pasadena, CA 91125. *Email: shizqi@cms.caltech.edu.*

with a joint function of both time and frequency. The recent advances of wavelet analysis have opened a new path for time-frequency analysis. A significant breakthrough of wavelet analysis is the use of multi-scales to characterize signals. This technique has led to the development of several wavelet-based time-frequency analysis techniques [26, 14, 31].

An important development in the time-frequency analysis is to study instantaneous frequency of a signal. Some of the pioneering work in this area was due to Van der Pol [41] and Gabor [20], who introduced the so-called Analytic Signal (AS) method that uses the Hilbert transform to determine instantaneous frequency of a signal. This Analytic Signal method is one of the most popular ways to define instantaneous frequency. Until very recently, this method works mostly for monocomponent signals in which the number of zero-crossings is equal to the number of local extrema [2]. There were other attempts to define instantaneous frequency such as the zero-crossing method [38, 39, 32] and the Wigner-Ville distribution method [2, 29, 36, 17, 28, 35]. However most of these methods are rather restrictive. More substantial progress has been made only recently with the introduction of the EMD method [23]. The EMD method provides an effective tool to decompose a signal into a collection of intrinsic mode functions (IMF) that allow well-behaved Hilbert transforms for computation of physically meaningful time-frequency representation. We remark that the Hilbert spectral representation based on the wavelet projection has also been carried in [34].

Inspired by the EMD method and the recently developed compressed (compressive) sensing theory, we propose a data-driven time-frequency analysis method. There are two important ingredients of this method. The first one is that the basis that is used to decompose the data is derived from the data rather than determined *a priori*. This explains the name "data-driven" in our method. The second ingredient is to look for the sparsest decomposition of the signal among the largest possible dictionary consisting of intrinsic mode functions. The adoption of this data-driven basis and the search for the sparsest decomposition over this highly redundant basis make our time-frequency analysis method fully adaptive to the signal. As we are going to demonstrate later, our method can reveal some hidden physical information of the signal, such as trend and instantaneous frequency.

Our data-driven time-frequency analysis method is motivated by the observation that many multi-scale data often have an intrinsic sparse structure in the time-frequency plane, although its representation in the physical domain could be quite complicated. The challenge is that such sparse representation is valid only for certain multiscale basis which is adapted to the data and is unknown *a priori*. Finding such nonlinear multiscale basis is an essential ingredient of our data-driven analysis method. In this sense, our problem is more difficult than the compressed (compressive) sensing problem in which the basis is assumed to be known *a priori*. One way to find the adaptive basis is to learn from the data if we have a large number of data samples that share the similar physical property. This does not apply to our problem since we deal with only a single signal. We overcome this difficulty by reformulating the problem as a nonlinear optimization among the largest possible dictionary. The trade-off is that such decomposition is not unique. We need to exploit

the intrinsic sparse structure of the data to select the sparsest one among all the possible decompositions.

In our method, the dictionary is given as following:

$$\mathcal{D} = \left\{ a(t)\cos\theta(t) : \ \theta'(t) \geq 0, \ a(t) \in V(\theta) \right\}, \tag{1}$$

where $V(\theta)$ is a linear space consisting of functions smoother than $\cos\theta(t)$. The construction of $V(\theta)$ with given $\theta(t)$ will be discussed later. We then decompose the signal over this dictionary by looking for the sparest decomposition. The sparest decomposition can be obtained by solving a nonlinear optimization problem:

$$\begin{aligned} P: \qquad & \text{Minimize} \qquad\qquad M \\ & \text{Subject to:} \quad f(t) = \sum_{k=1}^{M} a_k(t)\cos\theta_k(t), \quad a_k(t)\cos\theta_k(t) \in \mathcal{D}, \quad k = 1, \cdots, M. \end{aligned} \tag{2}$$

When the signal is polluted by noise, the equality in the above constraint can be relaxed to be an inequality depending on the noise level. This optimization problem can be viewed as a nonlinear version of the L^0 minimization problem and is known to be very challenging to solve. Inspired by the compressed (compressive) sensing theory [3], we propose two nonlinear matching pursuit methods to solve this nonlinear optimization problem.

Our nonlinear matching pursuit is inspired by the linear matching pursuit method [30, 40]. We first extract an intrinsic mode function $a(t)\cos\theta(t) \in \mathcal{D}$ from the signal $f(t)$ by looking for the one which matches the signal $f(t)$ best among all the elements in $\mathcal{D}$. This would imply the following nonlinear optimization problem:

$$\text{Minimize} \quad \|f(t) - a(t)\cos\theta(t)\|_{l^2}^2, \quad \text{Subject to} \ \ a(t)\cos\theta(t) \in \mathcal{D}.$$

Denote by $r(t)$ the residual after subtracting $a(t)\cos\theta(t)$ from $f(t)$, i.e. $r(t) = f(t) - a(t)\cos\theta(t)$. We can then treat $r(t)$ as a new signal to extract the remaining IMFs. There are two important advantages of this nonlinear matching pursuit approach. The first one is that this method is very stable to noise perturbation. The second one is that it can be implemented by the Fast Fourier Transform (FFT). Thus the complexity of our algorithm is of order $O(N \log N)$ where N is the number of data sample points that we use to represent the signal. The low computational cost and the robustness to noise perturbation make this method very effective in many applications. Moreover, for data that satisfy certain scale separation conditions, we can prove that our method recovers the IMFs and their instantaneous frequencies accurately.

We perform extensive numerical experiments to test the robustness and the accuracy of the nonlinear matching pursuit for both synthetic data and some real data. Our results show that the nonlinear matching pursuit can indeed decompose a multiscale signal into a sparse collection of intrinsic mode functions. We also compare our method with the original EMD method. For the data without noise, we find that our method gives results comparable to those obtained by the EMD method. Moreover, for noisy data, our method seems to

provide better estimation of the instantaneous frequency than EMD and recently developed EEMD method [42, 44].

A common difficulty in many data analysis methods is the relatively large error produced near the boundary of the data set. For the EMD method, this source of error is referred to as the "end effect", which is primarily caused by the use of cubic spline interpolation in constructing the envelope and median of the signal. Our nonlinear matching pursuit also suffers from this end effect if we extend the signal periodically beyond the finite interval in a naive way. In order to reduce the so-called end effect, we need to extend the signal to be a periodic signal as smoothly as possible beyond the support of the signal. Alternatively, we can use other non-periodic orthonormal basis to represent the data in the θ-space. We propose two methods to alleviate the difficulty associated with the end effect. The first one is based on a l^2 projection. The second one uses a l^1-regularized least square method. Preliminary numerical study shows that these modified methods seem to give more accurate approximation over the whole interval if the data satisfy certain scale separation property.

We remark that there has been some recent progress in developing a mathematical framework for an EMD like method using synchrosqueezed wavelet transforms by Daubechies, Lu and Wu [15]. This seems to be a very promising approach. We have performed some preliminary numerical experiments to compare the performance of our method with the synchrosqueezed wavelet approach. In many cases, we find that the two methods give comparable and complementary results. We are currently exploring a hybrid approach that combines the advantages of our method with those of the synchrosqueezed wavelet approach. Our preliminary results seem quite encouraging. We will report this in a forthcoming paper.

The remaining part of the paper is organized as follows. In Section 2, we give a brief review of some existing data analysis methods such as the matching pursuit, the basis pursuit and the EMD method. We introduce our adaptive data analysis method and its fast algorithm based on FFT in Section 3. In Section 4, we present some numerical experiments to demonstrate the performance of our method. In Section 5, we introduce two methods to deal with the "end effect" of the non-periodic data. We present some preliminary error analysis of our nonlinear matching method in Section 6. Nonlinear matching pursuit methods that use other norms will be discussed in Section 7. Some conclusions are made in Section 8.

2 Brief review of the existing sparse decomposition methods

A considerable focus of activities in the recent signal processing literature has been the development of the sparse signal representations over a redundant dictionary. Among these methods, the matching pursuit [30] and the basis pursuit [9] have attracted a lot of attention in recent years due to the development of the compressed (compressive) sensing. All these methods consist of two parts: a dictionary to decompose the signal and a decomposition method to select the sparsest decomposition.

2.1 Dictionaries

A dictionary is a collection of parametrized waveforms $\mathcal{D} = \{\phi_\gamma\}_{\gamma\in\Gamma}$. Many dictionaries have been proposed in the literature. Here we review a few of them that have been used widely.

A Fourier Dictionary. A Fourier dictionary is a collection of sinusoidal waveforms. More specifically, the waveforms consist of the following two families,

$$\phi_{\omega,0} = \cos(\omega t), \quad \phi_{\omega,1} = \sin(\omega t). \tag{3}$$

For the standard Fourier dictionary, ω runs through the set of all cosines with Fourier frequencies $\omega_k = 2k\pi/n,\ k = 0, 1, \cdots, n/2$, and all sines with Fourier frequencies $\omega_k = 2k\pi/n,\ k = 1, \cdots, n/2 - 1$, where n is the number of sample points. We can also obtain an overcomplete Fourier dictionary by sampling the frequencies more finely. Let $l > 1$. We may choose $\omega_k = 2k\pi/(ln),\ k = 0, 1, \cdots, ln/2$ for cosines and $\omega_k = 2k\pi/(ln),\ k = 1, \cdots, ln/2-1$ for sines. This is an l-fold overcomplete system. In the algorithm for non-periodic data, we will use this kind of overcomplete Fourier dictionary.

A Wavelet Dictionary. A wavelet dictionary is a collection of translations and dilations of the basic mother wavelet ψ, together with translations of the scaling function φ defined below:

$$\phi_{a,b,0} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \quad \phi_{a,b,1} = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right). \tag{4}$$

For the standard wavelet dictionary, we let a, b run through the discrete collection of mother wavelets with dyadic scales $a_j = 2^j/n,\ j = j_0, \cdots, \log_2(n)-1$, and locations that are integer multiples of the scale $b_{j,k} = ka_j,\ k = 0, \cdots, 2^j - 1$, and the collection of scaling functions at the coarse scale j_0. This dictionary consists of n waveforms, which form an orthonormal basis. As in the Fourier dictionary, an overcomplete wavelet dictionary can be obtained by sampling the locations more finely.

A Time-Frequency Dictionary. A typical time-frequency dictionary is the Gabor dictionary due to Gabor (1946). In this dictionary, we take $\gamma = (\omega, \tau, \theta, \delta)$, where $\omega \in [0, \pi)$ is frequency, τ is a location, θ is a phase, and δ is the duration. We define the waveform as follows:

$$\phi_\gamma(t) = \exp\left(-\frac{(t-\tau)^2}{\delta^2}\right)\cos\left(\omega(t-\tau)+\theta\right). \tag{5}$$

Such waveforms consist of frequencies near ω and essentially vanish far away from τ.

An EMD Dictionary. We can also define a dictionary via the EMD method. In the EMD method [23], the dictionary is the collection of all Intrinsic Mode Functions (IMF), which are functions defined descriptively by enforcing the following two conditions:

1. The number of the extrema and the number of the zero crossings of the function must be equal or differ at most by one;

2. At any point of the function, the average of the upper envelope and the lower envelope defined by the local extrema should be zero (symmetric with respect to zero).

Inspired by the EMD method, we will use a variant of the EMD dictionary to construct a sparse decomposition of a signal via nonlinear optimization.

2.2 Decomposition Methods

In this subsection, we will review a few decompositon methods that can be used to give a sparse decomposition of a signal by exploiting the intrinsic sparsity structure of the signal. In recent years, there have been a lot of research activities in looking for the sparest representation of a signal over a redundant dictionary [30, 9, 16, 4, 5], i.e. we look for a decomposition of a signal f over a given dictionary $\mathcal{D} = \{\phi_\gamma\}_{\gamma\in\Gamma}$ as

$$f = \sum_{k=1}^{m} \alpha_{\gamma_k}\phi_{\gamma_k} + R^m, \tag{6}$$

with the smallest m, where R^m is the residual. Whether or not a signal can be decomposed into a sparse decomposition depends on the choice of the dictionary that we use to decompose the signal. In general, a highly redundant dictionary tends to give better adaptivity, which implies better sparsity of the decomposition. However, when the dictionary is highly redundant, the decompositions are not unique. We need to give a criterion to pick up the "best" decomposition among all the possible choices.

Matching Pursuit. In [30], Mallat and Zhang introduced a general decomposition method called the matching pursuit that exploits the sparsity of a signal. Starting from an initial approximation $\mathbf{s}^0 = 0$ and a residual $\mathbf{r}^0 = s$, it builds up a sequence of sparse approximations step by step. At stage k, the method identifies the atom that best matches the residual and then adds to the current approximation, so that $s^k = s^{k-1}+\alpha_k\phi_{\gamma k}$, where $\alpha_k =< r^{k-1}, \phi_{\gamma k} >$ and $r^k = \mathbf{s} - \mathbf{s}^k$. After m steps, one has a representation of the form (6), with residual $R^m = r^m$. A similar algorithm was proposed for Gabor dictionaries by S. Qian and D. Chen [37].

An intrinsic feature of this algorithm is that when stopped after a few steps, it yields an approximate sparse representation using only a few atoms. When the dictionary is orthogonal, the method works perfectly. If the dictionary is not orthogonal, the situation is less clear. Recently, J. Tropp and A. Gilbert proved that under some assumptions on the basis, the orthogonal matching pursuit can solve the original l_0 minimization problem [40].

Basis Pursuit. Another important class of decomposition methods is the basis pursuit, which was introduced by S. Chen, D. Donoho and M. Saunders [9]. First, we reformulate the decomposition problem in the following way. Suppose we have a discrete dictionary of

p waveforms and we collect all these waveforms as columns of an n by p matrix $\mathbf{\Phi}$. The decomposition problem (6) can be reformulated as:

$$\mathbf{s} = \mathbf{\Phi}\alpha \tag{7}$$

where $\alpha = (\alpha_\gamma)$ is the vector of coefficients in (6).

The basic idea of the basis pursuit is to find a sparse representation of the signal whose coefficients have a minimal l_1 norm, i.e. the decomposition is obtained by solving the problem

$$\min \|\alpha\|_{l^1}, \quad \text{subject to } \mathbf{\Phi}\alpha = \mathbf{s}. \tag{8}$$

Recently, the basis pursuit has received a lot of attention, since it is found that under some conditions the basis pursuit can recover the exact solution of the original l_0 minimization problem [4, 16]. There has been extensive research to obtain a sparse representation by the basis pursuit in a variety of applications. An essential component of the basis pursuit is to solve the l^1 minimization problem. The computational cost of solving this l^1 minimization is more expensive than the least-square problem in the matching pursuit, although a powerful Splitt Bregman method has been introduced by Goldstein and Osher to speed up the l^1 minimization problem considerably [21].

The EMD decomposition via a sifting process. The EMD method decomposes a signal to IMFs sequentially. The basic idea behind this approach is the removal of the local median from a signal by using a sifting process. Specifically, for a given signal, $f(t)$, one tries to decompose it as a sum of the local median $m(t)$, and an IMF. A cubic spline polynomial is used to interpolate all the local maxima to obtain an upper envelope, and to interpolate all the local minima to obtain a lower envelope. By averaging the upper and lower envelopes, one obtains an approximate median for $m(t)$. One then decides whether or not to accept the obtained $m(t)$ as our local median depending on whether $f(t)-m(t)$ gives an acceptable IMF that satisfies the two conditions that are specified in the definition of an EMD dictionary. If $f(t) - m(t)$ does not satisfy these conditions, one can treat $f(t) - m(t)$ as a new signal and construct a new candidate for the IMF by using the same procedure described above. This sifting process continues until we obtain a satisfactory IMF, which we denote as $f_n(t)$. Now we can treat $f(t) - f_n(t)$ as a new signal, and apply the same procedure to generate the second IMF, $f_{n-1}(t)$. This procedure will continue until $f_0(t)$ is either monotone or contains at most one extremum. For more details of the sifting process, we refer to [23].

Decomposition based on a nonlinear TV^3 minimization. Inspired by the EMD method, we proposed a decomposition method based on a nonlinear TV^3 minimization in our previous paper [22]. Here TV^3 is the total variation of the third order derivative of a function, defined as $TV^3(f) = \int_a^b |f^{(4)}(t)|dt$. We use a TV^3 norm because the L^1 norm or the total variation norm is not strong enough to enforce the regularity of the median or the

envelope of our decomposition. For example, the use of the total variation norm, which is very popular in imaging processing community, tends to give a decomposition whose median or envelope is piecewise constant, which is referred to as the stair-case effect. The TV^3 norm, on the other hand, gives a much smooth decomposition for both the median and the envelope. Incidentally, the minimization using the TV^3 norm tends to favor piecewise cubic polynomials such as cubic splines. Thus, our method gives results that are qualitatively similar to the EMD method which uses cubic splines to construct its median and envelope from the local extrema of the signal.

We now give a brief review of our TV^3 decomposition method. In our approach, every element in our dictionary automatically satisfies the conditions of IMF. There is no need to do any sifting or use the Hilbert transform in our method. First, we decompose a signal $f(t)$ into its local median a_0 and an IMF $a_1 \cos\theta(t)$ by solving the following nonlinear optimization problem:

$$\begin{aligned} (P) \qquad \text{Minimize} \qquad & TV^3(a_0) + TV^3(a_1), \\ \text{Subject to:} \qquad & a_0(t) + a_1(t)\cos\theta(t) = f(t), \quad \theta'(t) \geq 0. \end{aligned} \tag{9}$$

To solve this nonlinear optimization problem, we proposed the following Newton type of iterative method:

Initialization: $\theta^0 = \theta_0$.

Main Iteration:

Step 1: Update $a_0^n,\ a_1^n,\ b_1^n$ by solving the following linear optimization problem:

$$\text{Minimize} \qquad TV^3(a_0^n) + TV^3(a_1^n) + TV^3(b_1^n), \tag{10}$$

$$\text{Subject to :} \qquad a_0^n + a_1^n \cos\theta^{n-1}(t) + b_1^n \sin\theta^{n-1}(t) = f(t). \tag{11}$$

Step 2: Update the phase function θ:

$$\theta^n = \theta^{n-1} - \mu \arctan\left(\frac{b_1^n}{a_1^n}\right), \tag{12}$$

where $\mu \in [0,1]$ is chosen to enforce that θ^n is an increasing function:

$$\mu = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta_k^{n-1} - \alpha \arctan\left(\frac{b_1^n}{a_1^n}\right)\right) \geq 0\right\}. \tag{13}$$

Step 3: If $\|\theta^n - \theta^{n-1}\|_2 \leq \epsilon_0$, stop. Otherwise, go to Step 1.

In [22], we performed a number of numerical experiments and compared the results with those obtained by the EMD (or EEMD) method. Our results show that this method shares many important properties with the original EMD method. Moreover, its performance does not depend on numerical parameters such as the number of sifting or the stop criterion, which seem to have a major effect on the original EMD method.

There are two limitations of this approach. The first one is that the computational cost to solve the TV^3 minimization problem is relatively high, even if we use the Split Bregman method of Goldstein and Osher [21]. The second one is that this method is more sensitive to noise perturbation, although a nonlinear filter was introduced to alleviate this difficulty. In this paper, we introduce another decomposition method based on the nonlinear matching pursuit which is very stable to noise perturnation and has a low computational cost.

3 Sparse time-frequency decomposition method based on nonlinear matching pursuit

Our adaptive data analysis method is based on finding the sparsest decomposition of a signal by solving a nonlinear optimization problem. First, we need to construct a large dictionary that can be used to obtain a sparse decomposition of the signal. In principle, the larger the dictionary is, the more adaptive (or sparser) the decomposition is.

3.1 Dictionary

In our method, the dictionary is chosen to be:

$$\mathcal{D} = \left\{ a(t)\cos\theta(t) :\ \theta'(t) \geq 0,\ a(t) \text{ is smoother than } \cos\theta(t) \right\}. \tag{14}$$

Let $V(\theta)$ be the collection of all the functions that are smoother than $\cos\theta(t)$:

$$V(\theta) = \left\{ g(t) :\ g(t) \text{ is smoother than } \cos\theta(t) \right\}. \tag{15}$$

Then $\mathcal{D}$ can be written as

$$\mathcal{D} = \left\{ a(t)\cos\theta(t) :\ \theta'(t) \geq 0,\ a(t) \in V(\theta) \right\}. \tag{16}$$

The simplest way to construct $V(\theta)$ is as follows

$$V(\theta) = \text{span}\left\{ \cos\left(\frac{k\theta}{L_\theta}\right), \sin\left(\frac{l\theta}{L_\theta}\right) : k = 0, \cdots, \lambda L_\theta,\ l = 1, \cdots, \lambda L_\theta \right\}, \tag{17}$$

where $\lambda \leq 1/2$ is a parameter to control the smoothness of functions in $V(\theta)$ and $L_\theta = (\theta(1) - \theta(0))/2\pi$. Once the function space $V(\theta)$ is chosen, the dictionary $\mathcal{D}$ is well defined and is given below:

$$\mathcal{D} = \left\{ a(t)\cos\theta(t) :\ \theta'(t) \geq 0,\ a(t) \in V(\theta) \right\}. \tag{18}$$

In some sense, the dictionary $\mathcal{D}$ defined above can be considered as a collection of IMFs. This property makes our method as adaptive as the EMD method. Since the dictionary $\mathcal{D}$ is highly redundant, the decomposition over this dictionary is not unique. We need a criterion to select the "best" one among all possible decompositions. We assume that the data we consider have an intrinsic sparse structure in the time-frequency plane in some nonlinear

and nonstationary basis. However, we do not know this basis *a priori* and we need to derive (or learn) this basis from the data. Based on this consideration, we adopt sparsity as our criterion to choose the best decomposition. This criterion yields the following nonlinear optimization problem:

$$\begin{aligned} P: \quad & \text{Minimize} \quad M \\ & \text{Subject to:} \quad f(t) = \sum_{k=1}^{M} a_k(t)\cos\theta_k(t), \quad a_k(t)\cos\theta_k(t) \in \mathcal{D}, \quad k = 1, \cdots, M, \end{aligned} \tag{19}$$

or

$$\begin{aligned} P_\delta: \quad & \text{Minimize} \quad M \\ & \text{Subject to:} \quad \|f(t) - \sum_{k=1}^{M} a_k(t)\cos\theta_k(t)\|_2 \le \delta, \quad a_k(t)\cos\theta_k(t) \in \mathcal{D}, \quad k = 1, \cdots, M, \end{aligned} \tag{20}$$

if the signal has noise with noise level δ.

After this optimization problem is solved, we get a very clear time-frequency representation:

$$\text{Frequency:} \quad \omega_k(t) = \theta_k'(t), \quad \text{Amplitude:} \quad a_k(t). \tag{21}$$

3.2 Nonlinear matching pursuit

The above optimization problem can be seen as a nonlinear L^0 minimization problem. Thanks to the recent developments of the compressed (compressive) sensing, it is now well known that there are two kinds of methods to solve a L^0 minimization problem: the basis pursuit and the matching pursuit. Since the dictionary we adopt here has infinitely many elements, the generalization of the basis pursuit is not so straightforward. But the idea of the matching pursuit can be generalized. Applying the idea of the matching pursuit to our problem, we obtain the following algorithm:

- $r^0 = f(t)$.

Step 1: Solve the following constraint nonlinear least-square problem (P_2):

$$\begin{aligned} P_2: \quad & \text{Minimize} \quad \|r^{k-1} - a_k(t)\cos\theta_k(t)\|_2^2 \\ & \text{Subject to:} \quad \theta_k' \ge 0, \quad a_k(t) \in V(\theta_k).. \end{aligned} \tag{22}$$

Step 2: Update the residual

$$r^k = f(t) - \sum_{j=1}^{k} a_j(t)\cos\theta_j(t). \tag{23}$$

Step 3: If $\|r^k\|_2 < \epsilon_0$, stop. Otherwise, go to Step 1.

In the first step of the above iterative algorithm, we need to solve a nonlinear least-square problem P_2. To solve this problem, we use the following Gauss-Newton type method:

- $\theta_k^0 = \theta_0$.

Step 1: Solve the following linear least-square problem:

$$\begin{aligned} P_{2,l_2} \qquad & \text{Minimize} \quad \|r^{k-1} - a_k^{n+1}(t)\cos\theta_k^n(t) - b_k^{n+1}(t)\sin\theta_k^n(t)\|_2^2 \\ & \text{Subject to} \quad a_k^{n+1}(t),\ b_k^{n+1}(t) \in V(\theta_k^n).. \end{aligned} \tag{24}$$

Step 2: Update θ_k^n,

$$\theta_k^{n+1} = \theta_k^n - \lambda \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right), \tag{25}$$

where $\lambda \in [0,1]$ is chosen to make sure that θ_k^{n+1} is a monotonely increasing function.

$$\lambda = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta_k^n - \alpha \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right) \geq 0\right\}. \tag{26}$$

Step 3: If $\|\theta_k^{n+1} - \theta_k^n\|_2 < \epsilon_0$, stop. Otherwise, go to Step 1.

3.3 An algorithm based on the Fast Fourier Transform (FFT)

In the iterative algorithm, we need to solve a least square problem in each step. This is the most expensive part of the algorithm especially when the number of the data points is large. Next, we will introduce an approximate method based on FFT to solve the above least square problem.

The least-square problem that we need to solve in the iterative algorithm is described below:

$$\begin{aligned} & \text{Minimize} \quad \|r(t) - a(t)\cos\theta(t) - b(t)\sin\theta(t)\|_2^2 \\ & \text{Subject to} \quad a(t),\ b(t) \in V(\theta). \end{aligned} \tag{27}$$

In order to simplify the notations, we drop the subscript and superscript here.

We observe that if $\theta(t)$ is a linear function, e.g. $\theta(t) = t$, then $V(\theta)$ can be simply defined as

$$V(t) = \text{span}\left\{\cos\left(\frac{kt}{n}\right), \sin\left(\frac{lt}{n}\right) : k = 0, \cdots, \lambda n,\ l = 1, \cdots, \lambda n\right\}. \tag{28}$$

Then the above least-sqaure problem can be solved by projecting $r(t)$ into the following space,

$$S = \text{span}\left\{\cos\left(\frac{kt}{n}\right), \sin\left(\frac{lt}{n}\right) : k, l = (1-\lambda)n, \cdots, (1+\lambda)n\right\}. \tag{29}$$

This projection can be performed by using the Fourier Transform as follows:

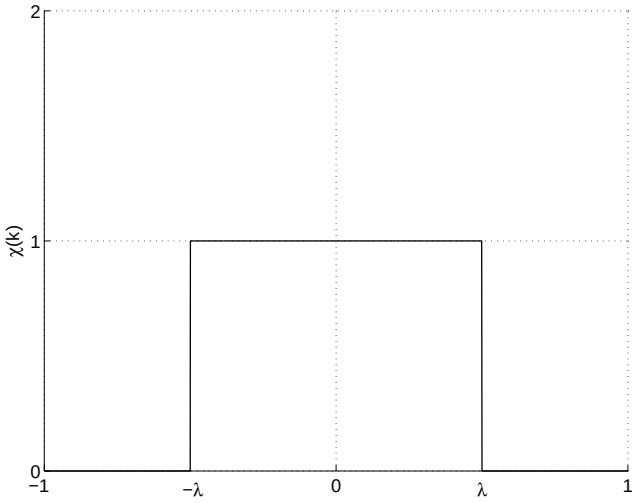

Figure 1: The low-pass filter $\chi(k)$ corresponding to the choice of $V(t)$ in (28).

Step 1: Calculate the Fourier transform of $r(t)$, denoted by $\widehat{r}(k)$.

Step 2: Compute $a(t)$ and $b(t)$ by the following formulas:

$$a(t) = 2Re\left\{\mathcal{F}^{-1}\left[\widehat{r}(k+1)\chi(k)\right]\right\}, \tag{30}$$

$$b(t) = 2Im\left\{\mathcal{F}^{-1}\left[\widehat{r}(k+1)\chi(k)\right]\right\}, \tag{31}$$

where $\chi(k)$ is a low-pass filter, which is determined by the choice of $V(t)$. Corresponding to the choice of $V(t)$ in (28), the cutoff function $\chi(k)$ is a stair function as shown in Fig. 1.

Notice that in the iterative process, the derivative of the phase function $\theta(t)$ is always monotonically increasing. Thus, we can use $\theta(t)$ as a new coordinate. In this new coordinate, $\theta(t)$ is simply a linear function of θ. Then from the previous analysis, the least-square problem can be solved by using the Fourier Transform. More specifically, for a given increasing phase function $\theta(t)$, we have the following algorithm to solve the least-square problem (27) approximately:

Step 1: Apply an interpolation method to obtain the value of $r(t)$ at a uniform mesh in the θ coordinate, $r_\theta(\theta_j)$:

$$r_\theta(\theta_j) = \text{Interpolate } (\theta(t_i), r, \theta_j), \tag{32}$$

where $\theta_j,\ j = 1, \cdots, N$ are uniformly distributed in the θ coordinate.

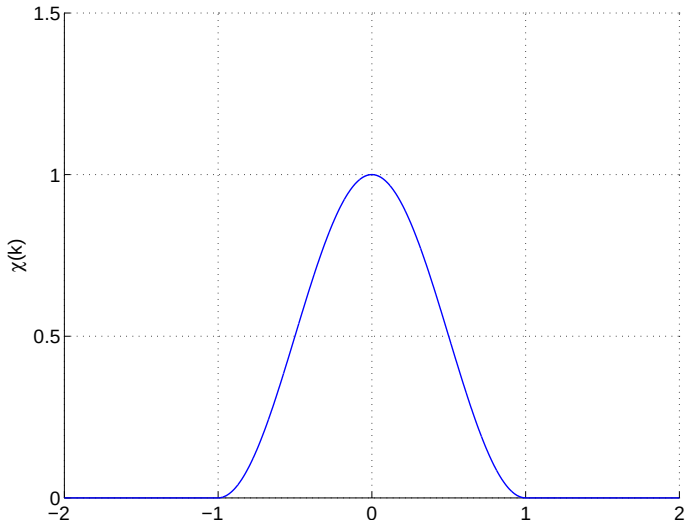

Figure 2: The low-pass filter $\chi(k)$ used in our computation.

Step 2: Apply $\chi(k)$ to the Fourier Transform of r_θ to compute the envelope $a(t)$ and $b(t)$:

$$a(t) = 2Re\left\{\mathcal{F}^{-1}\left[\widehat{r}_\theta(k+1)\chi(k)\right]\right\}, \tag{33}$$

$$b(t) = 2Im\left\{\mathcal{F}^{-1}\left[\widehat{r}_\theta(k+1)\chi(k)\right]\right\}, \tag{34}$$

where $\chi(k)$ is same as that shown in Fig. 1.

The low-pass filter $\chi(k)$ is determined by the choice of $V(\theta)$. The definition of $V(\theta)$ in (28) implies that $\chi(k)$ a stair function, which is not the best choice as a filter. We can define a different space for $V(\theta)$ by choosing an appropriate $\chi(k)$. This opens up many choices for $V(\theta)$. In this paper, we choose the following low-pass filter $\chi(k)$ (see also Fig. 2) to define $V(\theta)$:

$$\chi(k) = \begin{cases} 1+\cos\pi k, & -1 < k < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

Now, by incorporating the FFT-based solver in our iterative algorithm, we get the following FFT-based iterative algorithm:

- $\theta_k^0 = \theta_0$.

Step 1: Interpolate $r(t)$ to a uniform mesh in the θ^n coordinate to get $r_{\theta^n}(\theta_j^n)$:

$$r_{\theta^n}(\theta_j^n) = \text{Interpolate } \left(\theta^n(t_i), r, \theta_j^n\right), \tag{36}$$

where $\theta_j^n,\ j = 1, \cdots, N$ are uniformly distributed in θ^n coordinate.

Step 2: Apply $\chi(k)$ to the Fourier Transform of r_{θ^n} to compute a^{n+1} and b^{n+1} on the mesh of the θ^n coordinate:

$$a^{n+1}(\theta^n) = 2Re\left\{\mathcal{F}^{-1}\left[\widehat{r}_{\theta^n}(k+1)\chi(k)\right]\right\}, \tag{37}$$

$$b^{n+1}(\theta^n) = 2Im\left\{\mathcal{F}^{-1}\left[\widehat{r}_{\theta^n}(k+1)\chi(k)\right]\right\}. \tag{38}$$

Step 3: Interpolate $a^{n+1}(\theta^n)$ and $b^{n+1}(\theta^n)$ back to the uniform mesh of t:

$$a^{n+1}(t_i) = \text{Interpolate } \left(\theta_j^n, a^{n+1}(\theta_j^n), t_i\right), \tag{39}$$

$$b^{n+1}(t_i) = \text{Interpolate } \left(\theta_j^n, b^{n+1}(\theta_j^n), t_i\right). \tag{40}$$

Step 4: Update θ^n in the t coordinate:

$$\theta_k^{n+1} = \theta_k^n - \lambda \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right), \tag{41}$$

where $\lambda \in [0,1]$ is chosen to make sure that θ_k^{n+1} is monotonically increasing:

$$\lambda = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta_k^n - \alpha \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right) \geq 0\right\}. \tag{42}$$

Step 5: If $\|\theta_k^{n+1} - \theta_k^n\|_2 < \epsilon_0$, stop. Otherwise, go to Step 1.

In our implementation, instead updating the phase function θ, we update the instantaneous frequency $\theta'(t)$:

$$\left(\theta_k^{n+1}\right)' = (\theta_k^n)' + \lambda \frac{a_k^{n+1}\left(b_k^{n+1}\right)' - b_k^{n+1}\left(a_k^{n+1}\right)'}{\left(a_k^{n+1}\right)^2 + \left(b_k^{n+1}\right)^2}. \tag{43}$$

The phase function can be reconstructed from the instantaneous frequency by integration. The integration constant is set to 0 since this constant does not change the instantaneous frequency.

In the above algorithm, we need to perform the Fourier Transform. In general, this works well only for periodic data. For a non-periodic signal, we extend the signal by a mirror reflection and treat the extended signal as a periodic signal. This is a very crude extension. Later on in Section 5, we will discuss two other approaches to extend the data periodically outside the support of the data in a way that would minimize the end effect. The initial guess of θ can be generated by other time-frequency analysis methods, such as the synchrosqueezed wavelet transforms [15]. In the following numerical examples, we obtain our initial guess using a simple approach based on the Fourier Transform. More precisely, by estimating the wavenumber by which the high frequency components are centered around, we can obtain a reasonably good initial guess for θ. The initial guess for θ obtained in this way is a linear function. As we will see in the following numerical examples, even with these relatively rough initial guesses for θ, our algorithm still converges to the right answer with accuracy determined by the noise level.

4 Numerical Results

In this section, we will present a number of numerical experiments to demonstrate the accuracy and robustness of our method. We also compare the performance of our method with that of EMD or EEMD. A main focus of our numerical study is the robustness of the decomposition to signals that are polluted with a significant level of noise. When the signal is free of noise, we observe that the performance of our method is comparable to that of the EMD/EEMD method. However, when the signal is polluted by noise with a significant noise-to-signal ratio, our nonlinear matching pursuit method tends to give better performance than that of the EMD/EEMD method.

Throughout this section, we denote $X(t)$ as white noise with zero mean and variance $\sigma^2 = 1$. The Signal-to-Noise Ratio (SNR, measured in dB) is defined by

$$\text{SNR[dB]} = 10 \log_{10} \left(\frac{\text{var} f}{\sigma^2} \right). \tag{44}$$

We will apply our method to several different signals with increasing level of difficulty.

Example 1 The first example that we consider is a simple non-stationary signal consisting of a single IMF, which is given below

$$f(t) = \cos(60\pi t + 15 \sin(2\pi t)). \tag{45}$$

In Fig. 3, we plot the original signal on the left column and the instantanenous frequency on the right column. The curve with the red color corresponds to the exact instantaneous frequency and the one with the blue color corresponds to the one obtained by by our method. The top row corresponds to the original signal without noise, $f(t)$. The middle row corresponds to the same signal with a moderate noise level ($f(t)+X(t)$, SNR=-3.01dB) and the bottom row corresponds to the signal with large noise ($f(t) + 3X(t)$, SNR=-12.55dB). In the case when no noise is added, the instantaneous frequency obtained by our method is almost indistinguishable from the exact one, see the first row of Fig. 3. When the signal has noise, our method can still extract the instantaneous frequency and corresponding IMF with reasonable accuracy, see Fig. 3 and Fig. 4.

In Fig. 4, we compare the IMFs extracted by our method with those obtained by the EMD/EEMD method. For the signal without noise, we use the EMD method to decompose the signal. For the signal with noise, we use the EEMD method to decompose the signal. In the EEMD approach, the number of ensembles is chosen to be 200 and the standard deviation of the added noise is 0.2. In each ensemble, the number of sifting is set to 8. Even though the signal we consider has only a single IFM, the EEMD method still produces several IFMs. Among different components of IMFs that are produced by the EEMD method, we select the one that is closest to the exact IMF in l^2 norm and display it in Fig. 4.

When the signal does not have noise, both our method and the EMD method produce qualitatively the same result for this simple signal, see the first row of Fig. 4. When noise

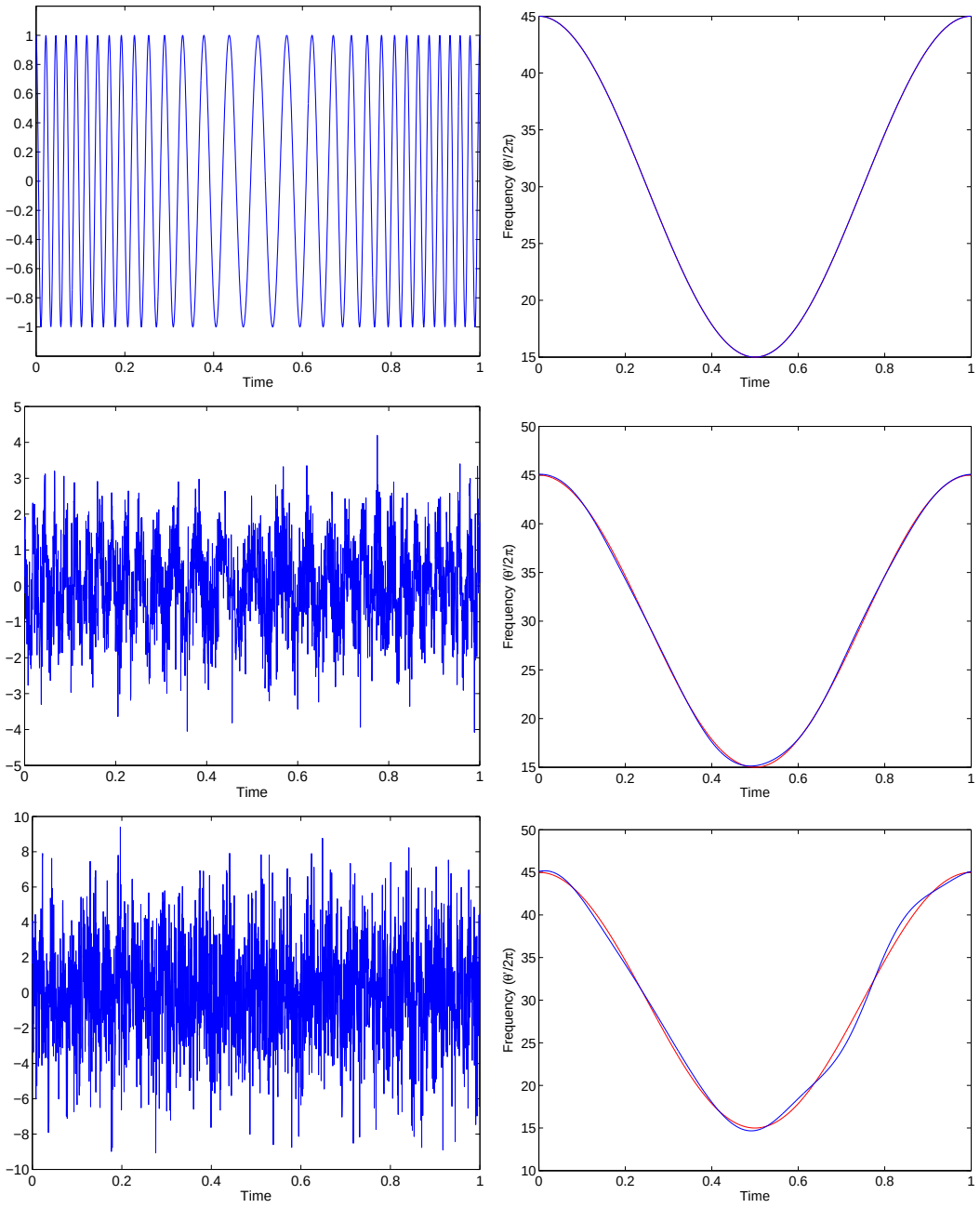

Figure 3: Top row: left: the original signal defined by (45) without noise; right: Instantaneous frequencies; red: exact frequency; blue: numerical results. Middle row: the same as the top row except white noise $X(t)$ is added to the original signal, the corresponding SNR is -3.01 dB. Bottom row: White noise $3X(t)$ is added to the original signal, the corresponding SNR is -12.55 dB.

is added, the situation is quite different. The IMFs extracted by our method still have reasonable accuracy. However, the IMF decomposed by EEMD fails to capture the phase of the exact IMF in some region. As a consequence, the accuracy of the instantaneous frequency obtained by the EEMD method is very poor (not shown here).

Example 2

Next, we consider a signal that consists of two IMFs. The frequencies of these two IMFs are well seperated. The analytical form of this signal is given below:

$$f(t) = \frac{1}{1.5 + \cos(2\pi t)} \cos(60\pi t + 15\sin(2\pi t)) + \frac{1}{1.5 + \sin(2\pi t)} \cos(160\pi t + \sin(16\pi t)). \quad (46)$$

For the signal without noise, our method gives an accurate approximation to the instantaneous frequency and the IMF, see the upper row of Fig. 5 and Fig. 6. For the signal with noise, the instantaneous frequency and IMF obtained by our method still have reasonable accuracy, see the bottom row of Fig. 5 and Fig. 7. In comparison, the IMF obtained by the EEMD method seems to produce a relatively large phase error, see Fig. 7. We do not plot the instantaneous frequency obtained by the EMD or EEMD method here since they tend to produce a lot of oscillation.

Example 3

Now, we consider a signal that consists of three IMFs by adding one more component to the signal in Example 2.

$$\begin{aligned} f(t) &= \frac{1}{1.5 + \cos(2\pi t)} \cos(60\pi t + 15\sin(2\pi t)) + \frac{1}{1.5 + \sin(2\pi t)} \cos(160\pi t + \sin(16\pi t)) \\ &\quad + (2 + \cos(8\pi t)) \cos(140\pi (t+1)^2). \end{aligned} \quad (47)$$

For this more complicated signal, our method still works well even with a large noise level, see Fig. 8.

Example 4

For the last synthetic example, we consider a more challenging signal consisting of two IMFs whose instantaneous frequencies overlap in some region. More specifically, the instantaneous frequency trajectories of the two IMFs intersect each other. Thus there is no spectrum gap (or separation) between the two instantaneous frequencies near the region of intersection. This makes it extremely difficult to separate the two IMFs accurately since they are essentially not separabe near the region of intersection of their instantaneous frequencies. The signal is generated by the analytical formula given below.

$$f(t) = \cos(20\pi t + 40\pi t^2 + \sin(2\pi t)) + \cos(40\pi t). \quad (48)$$

Fig. 9 gives the instantaneous frequencies recovered by our method for the signal without and with noise. Near the region of intersection, the two IMFs can not be decomposed

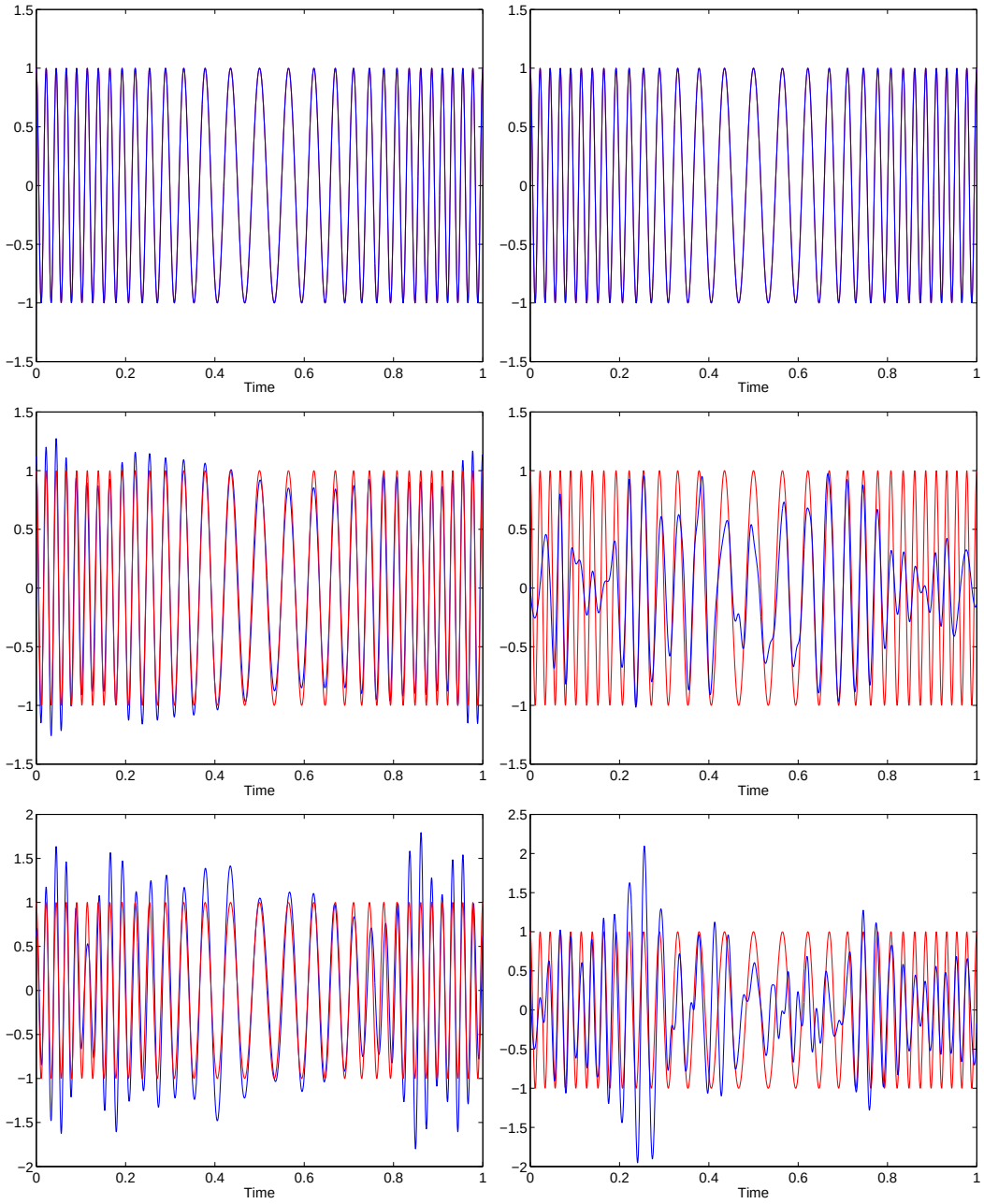

Figure 4: The IMFs extracted by our method and EMD/EEMD method. Left column: IMFs extracted by our method; Right column: IMFs obtained by EMD/EEMD method. Top row: IMFs from $f(t)$; Middle row: IMFs from $f(t) + X(t)$; Bottom row: IMFs from $f(t) + 3X(t)$. $f(t)$ is defined in (45).

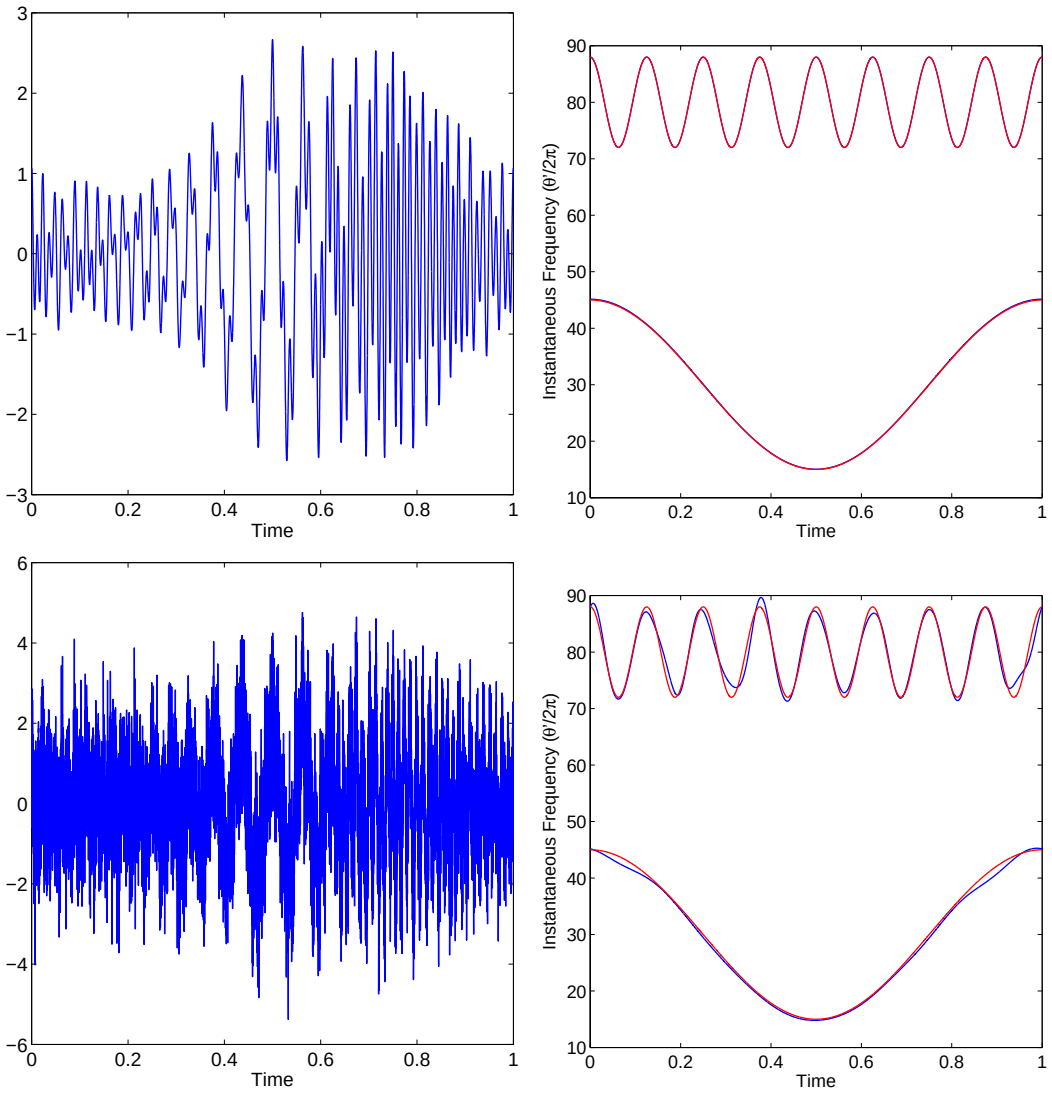

Figure 5: Upper row: left: signal defined in (46) without noise; right: Instantaneous frequencies; red: exact frequencies; blue: numerical results.Lower row: same as the upper row except white noise $X(t)$ was added to the original signal, corresponding SNR is 0.3 dB.

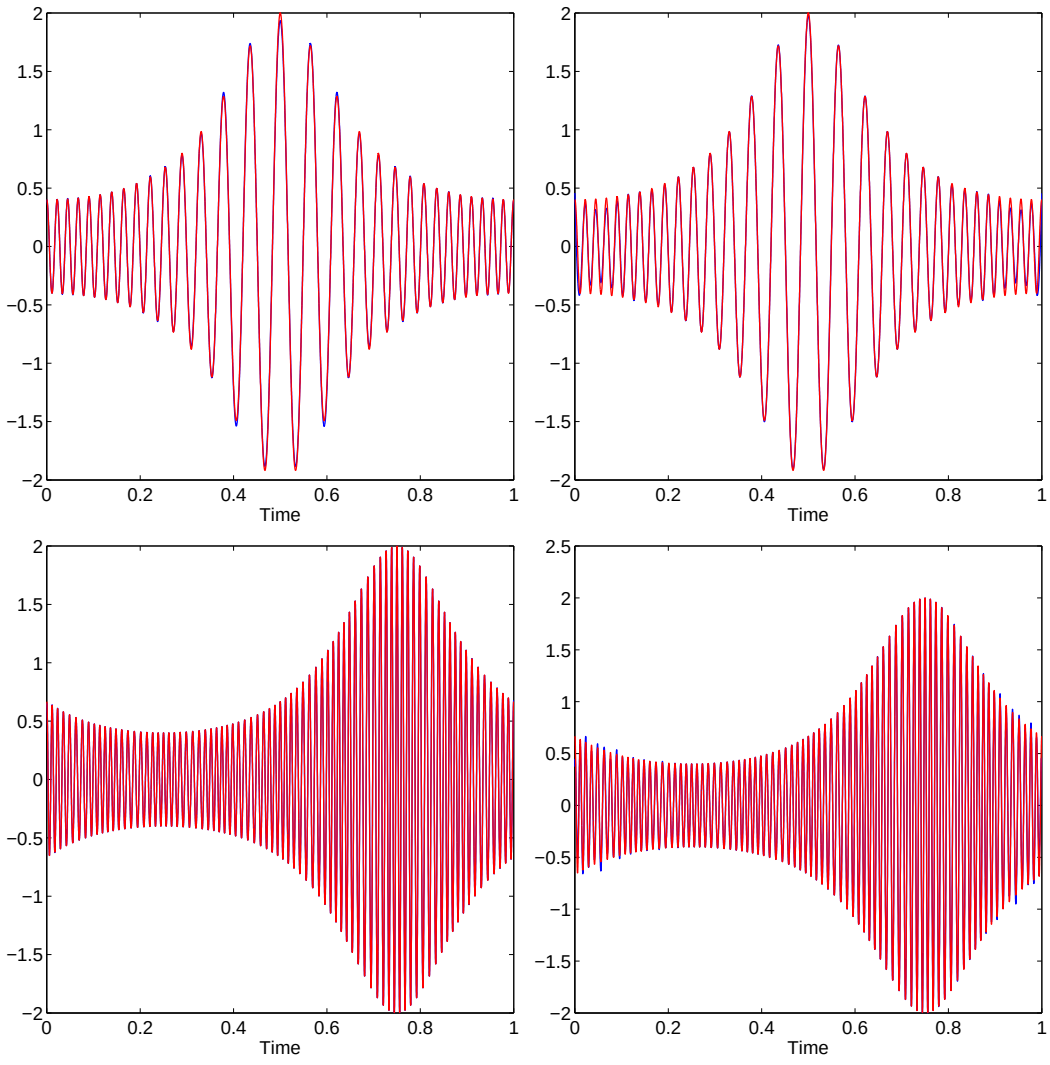

Figure 6: The IMFs extracted by our method and the EMD method for the signal $f(t)$ defined in (46). Left column: IMFs extracted by our method; Right column: IMFs obtained by EMD method. Top row: IMF with low frequency; Bottom row: IMF with high frequency.

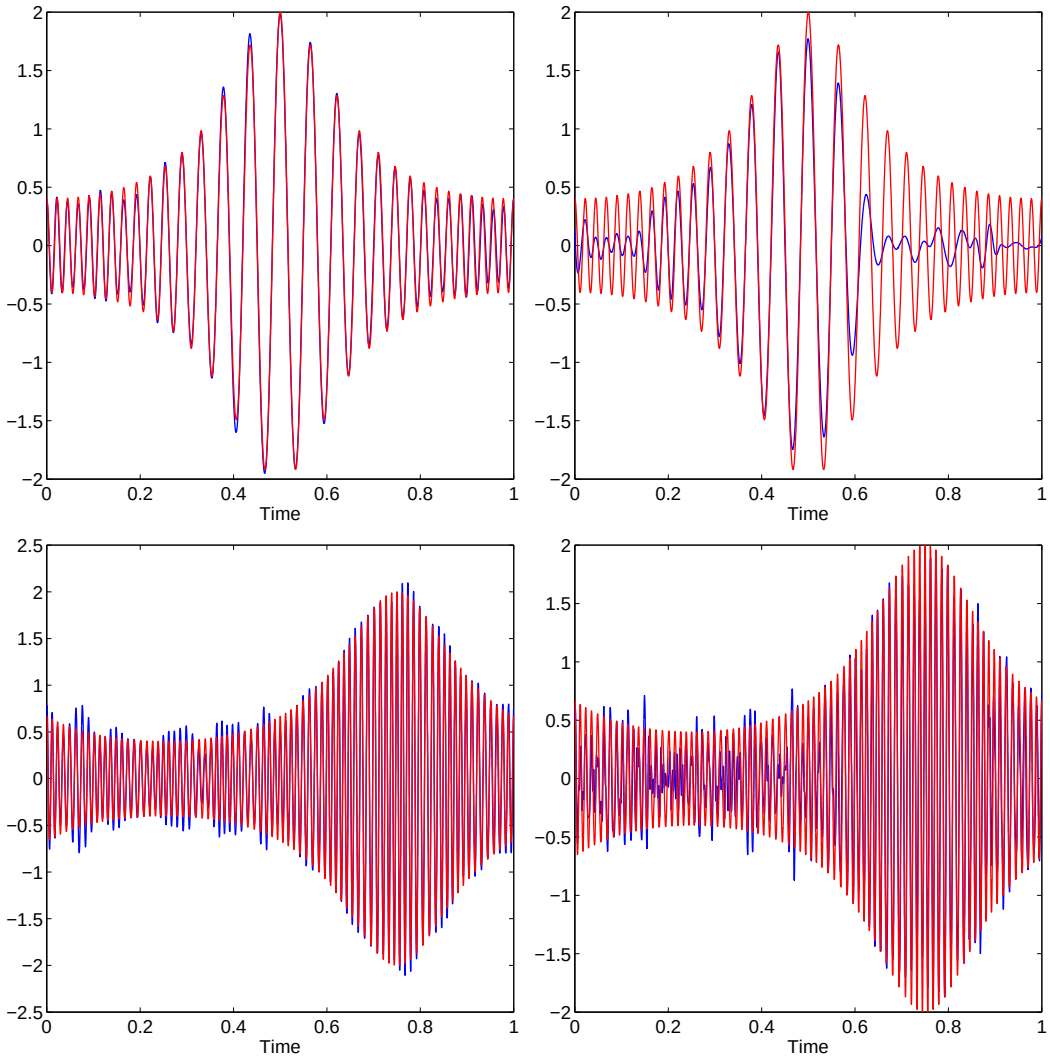

Figure 7: The IMFs extracted by our method and EEMD method from signal $f(t) + X(t)$, where $f(t)$ is defined in (46). Left column: IMFs extracted by our method; Right column: IMFs obtained by the EMD method. Top row: IMF with low frequency; Bottom row: IMF with high frequency.

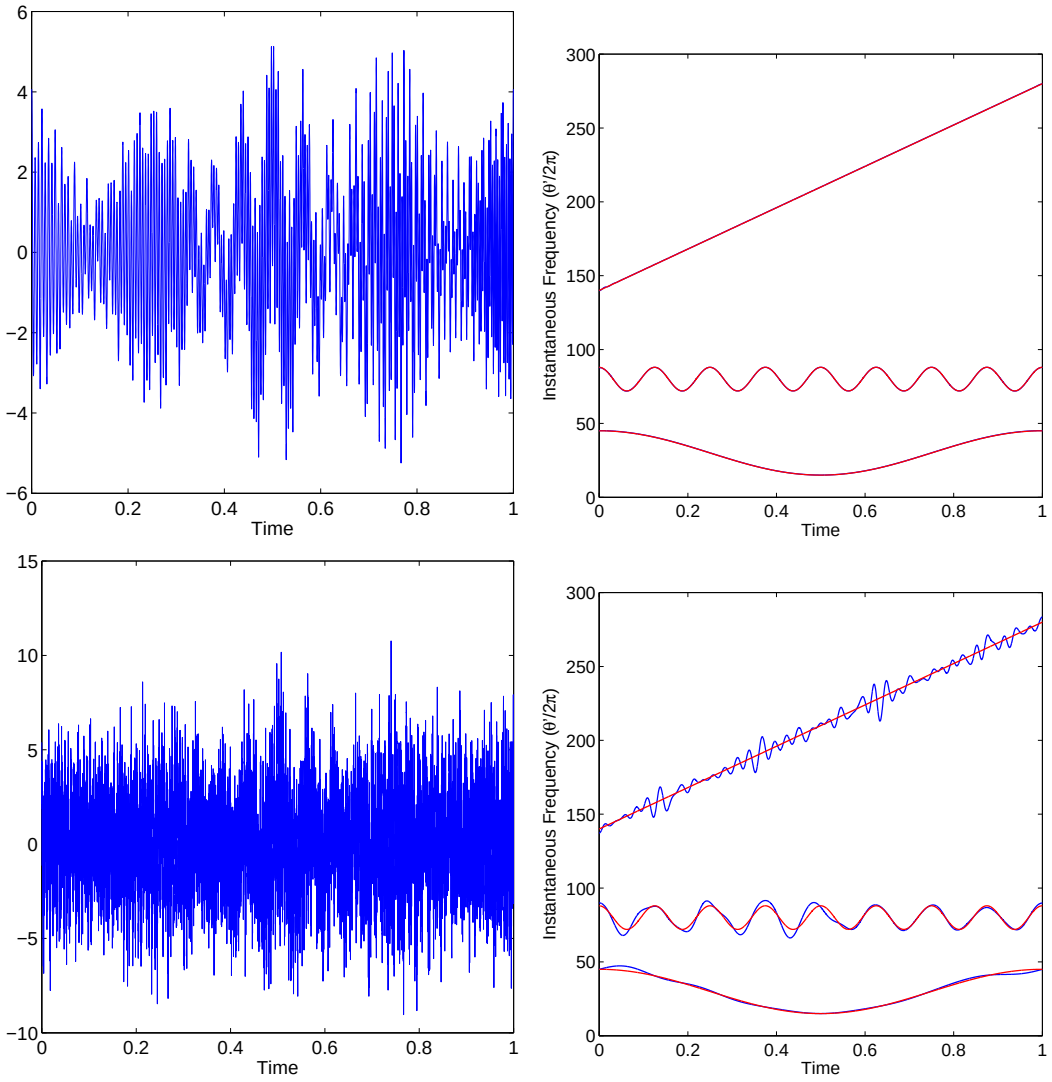

Figure 8: Upper row: left: signal defined in (47) without noise; right: Instantaneous frequencies; red: exact frequencies; blue: numerical results.Lower row: same as the upper row except white noise $2X(t)$ was added to the original signal, corresponding SNR is -0.8 dB.

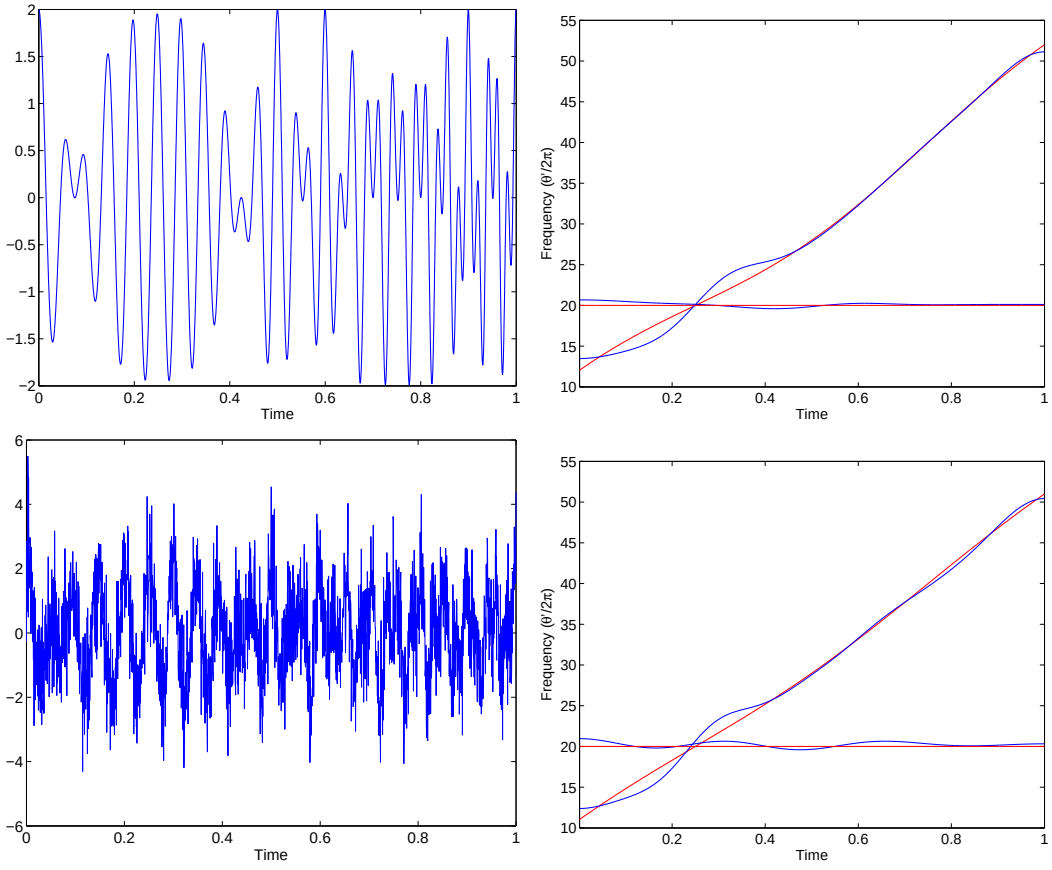

Figure 9: Upper row: left: signal defined in (48) without noise; right: Instantaneous frequencies; red: exact frequencies; blue: numerical results. Lower row: same as the upper row except white noise $X(t)$ was added to the original signal, corresponding SNR is 0.68 dB.

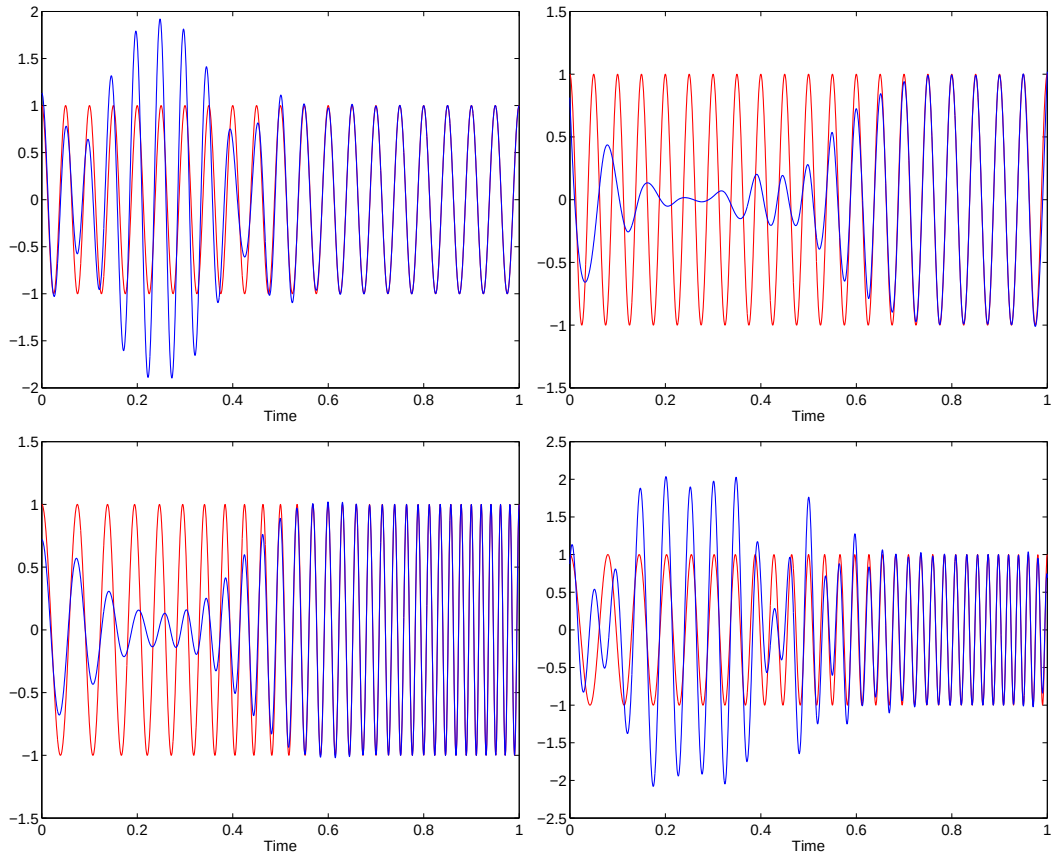

Figure 10: The IMFs extracted by our method and EMD method from the signal $f(t)$ defined in (48). Left column: IMFs extracted by our method; Right column: IMFs obtained by EMD method. Top row: IMF with low frequency; Lower row: IMF with high frequency.

uniquely. We observe that the instantaneous frequency has some bias toward the 'ideal' one. But in the area far from the region of intersection, the recovery is quite reasonable. This shows that our method has a temporal locality property, which is important in many physical applications.

The IMFs decomposed by our method and the EMD method from the data without noise are shown in Fig. 10. In the area away from the intersection, both our method and EMD work very well. In the region close to the intersection, the phase captured by our method seems to be better than that obtained by EMD. After the noise is added, the EEMD method seems to fail to give a good approximation to the IMF, even in the region far away from the intersection. In comparison, our method still can extract the IMF with acceptable accuracy as shown in Fig. 11.

Real Data: Length-of-Day Data

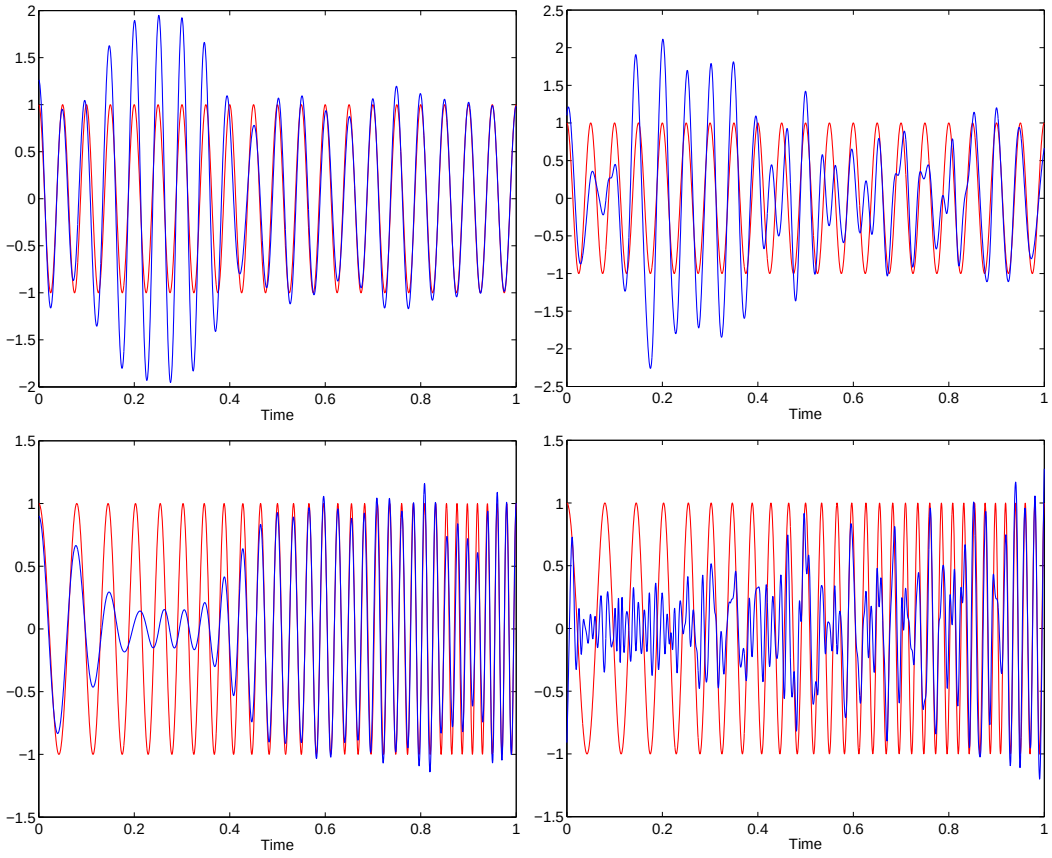

Figure 11: The IMFs extracted by our method and EEMD method from the signal $f(t) + X(t)$, where $f(t)$ is defined in (48). Left column: IMFs extracted by our method; Right column: IMFs obtained by EMD method. Top row: IMF with low frequency; Lower row: IMF with high frequency.

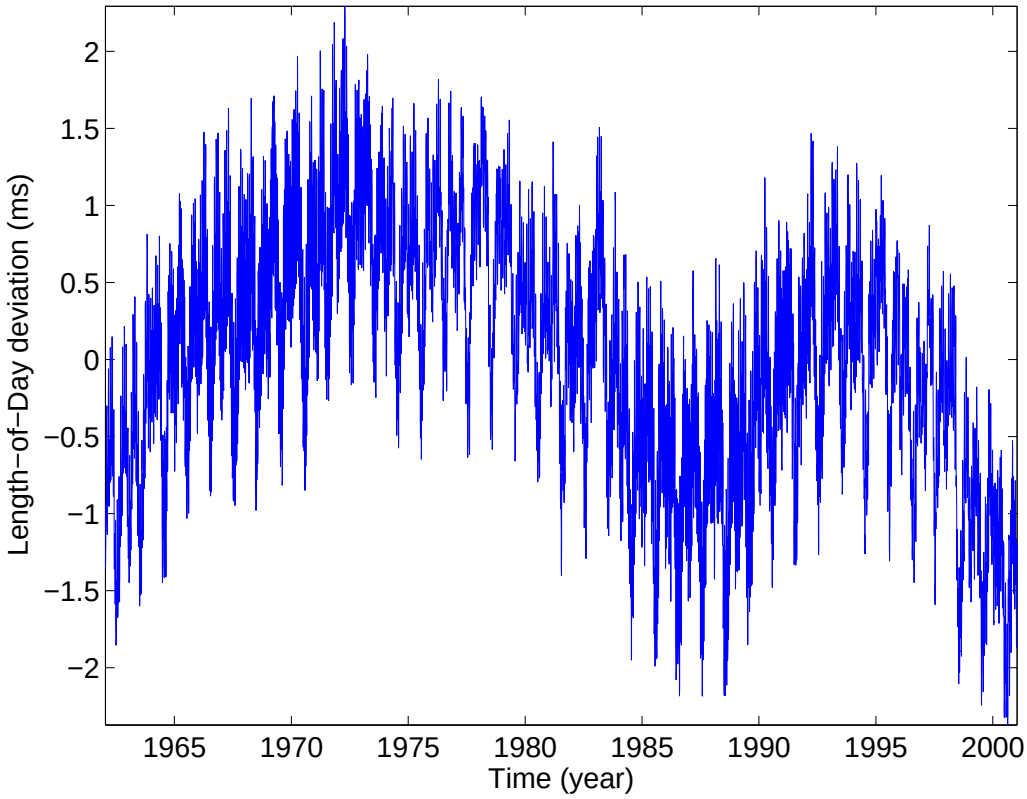

Figure 12: The daily Length-of-Day data from Jan 20, 1962 to Jan 6, 2001.

Next, we apply our method to the Length-of-Day data, see Fig. 12. The data we adopt here was produced by Gross [18], covering the period from 20 January 1962 to 6 January 2001, for a total of 14,232 days (approximate 39 years). In our previous paper [22], we also studied this data set. Due to the high computational cost associated with the l^1 minimization, we can not decompose the enture data set. Instead, we decompose a segment of the data that contains 700 consecutive days. Thanks to the low computational cost of the FFT-based nonlinear matching pursuit method, we can now study the entire data set without any compromise.

Fig. 13 displays the first 9 IMFs extracted by our method. In Fig. 13, these IMFs are sorted by their energy. The results we obtain are qualitatively consistent with those obtained by EMD/EEMD [19, 44]. Comparing with the results obtained by the EMD method, the results obtained by our method do not suffer from the mode mixing phenomenon that is present in the EMD decomposition. Moreover, each component is enforced to be an IMF by the construction of our dictionary. Thus, there is no need to do post-processing as was done in the EEMD method. The lowest panel in Fig. 13 shows the residual after we substract these 9 IMFs. The residual still has some oscillations, but the amplitude is very small. One can decompose the residual further. After extracting the 19 IMFs, the residual become

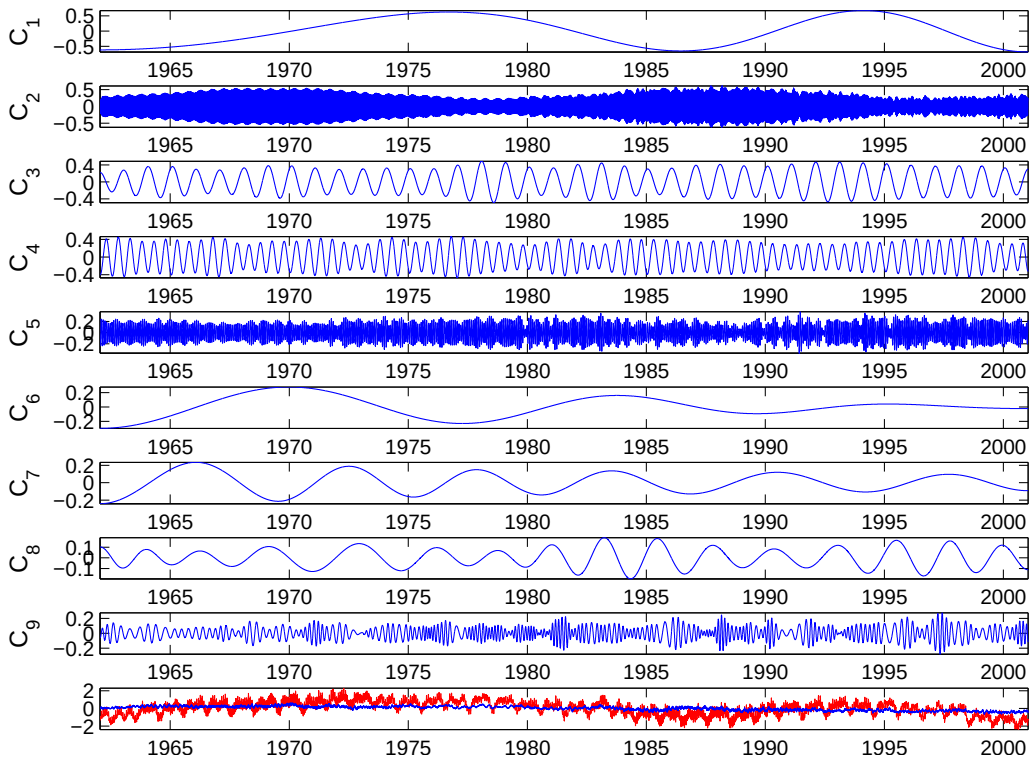

Figure 13: The first 9 IMFs sorted by energy and the residual. In the lowest panel, the red line is the original data and blue line is the residual after substracting the above 9 IMFs.

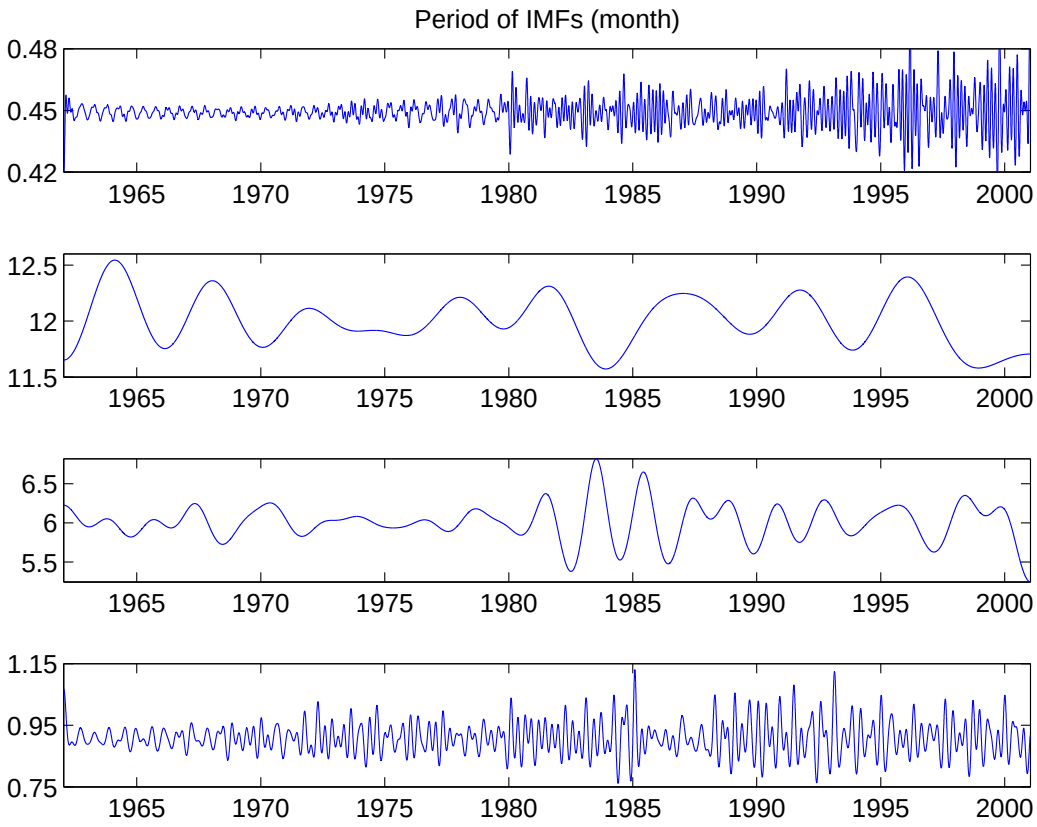

Figure 14: The period of C_2, C_3, C_4, C_5.

pretty smooth. Comparing with the first 9 IMFs, the energy level of the remaining 10 IMFs is relatively small. The first 9 IMFs have 67% of the energy of the original data, and the first 5 IMFs contain 55%.

It is interesting to note that the each IMF that we obtain has a clear physical interpretation. To demonstrate this, we plot the periods of the four IMFs, C_2, C_3, C_4, C_5, in Fig. 14 in the unit of month. As we can see, the period of C_2 is around 14 days, corresponding to the semi-monthly tides. Similarly, the period of C_3 is about one year, corresponding to the annual cycle and C_4 corresponds to the semi-annual cycle. The period of C_5 is about 28 days, corresponding to the monthly tides.

5 Algorithms for non-periodic data

The nonlinear matching pursuit algorithm based on the Fourier Transform assumes that the signal is periodic. For non-periodic data, the results given by the algorithm based on FFT tend to produce some oscillations near the boundary. This so-called "end effect" is also present in the EMD method and other data analysis methods. In our method, the "end

effect" comes from the use of the Fourier Transform in the algorithm. Clearly, the Fourier Transform is only a convenient tool for us and is not essential in our approach. In fact, non-periodic data can be handled by using the following modified algorithm:

- $\theta_k^0 = \theta_0$.

Step 1: Interpolate $r(t)$ to a uniform mesh of the θ^n coordinate to get $r_{\theta^n}(\theta_j^n)$:

$$r_{\theta^n}(\theta_j^n) = \text{Interpolate } \left(\theta^n(t_i), r, \theta_j^n\right), \tag{49}$$

where $\theta_j^n,\ j = 1, \cdots, N$ are uniformly distributed in the θ^n coordinate.

Step 2: Project $r_{\theta^n}\cos\theta^n$ and $r_{\theta^n}\sin\theta^n$ into the $V(\theta^n)$ space to get a^{n+1} and b^{n+1} as follows:

$$a^{n+1}(\theta^n) = 2P_{\theta^n}\left(r_{\theta^n}\cos\theta^n\right), \tag{50}$$

$$b^{n+1}(\theta^n) = 2P_{\theta^n}\left(r_{\theta^n}\sin\theta^n\right), \tag{51}$$

where P_{θ^n} is a projection operator to the $V(\theta^n)$ space. Note that the operator P_{θ^n} does not have be a L^2 projection. The choice of P_{θ^n} plays key role in this algorithm.

Step 3: Interpolate $a^{n+1}(\theta^n)$ and $b^{n+1}(\theta^n)$ back to the uniform mesh of t:

$$a^{n+1}(t_i) = \text{Interpolate } \left(\theta_j^n, a^{n+1}(\theta_j^n), t_i\right), \tag{52}$$

$$b^{n+1}(t_i) = \text{Interpolate } \left(\theta_j^n, b^{n+1}(\theta_j^n), t_i\right). \tag{53}$$

Step 4: Update θ^n in the t coordinate:

$$\theta_k^{n+1} = \theta_k^n - \lambda \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right), \tag{54}$$

where $\lambda \in [0,1]$ is chosen to ensure that θ_k^{n+1} is monotonically increasing,

$$\lambda = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta_k^n - \alpha \arctan\left(\frac{b_k^{n+1}}{a_k^{n+1}}\right)\right) \geq 0\right\}. \tag{55}$$

Step 5: If $\|\theta_k^{n+1} - \theta_k^n\|_2 < \epsilon_0$, stop. Otherwise, go to Step 1.

In the above algorithm, the key is Step 2 which aims to extract the components of $r_{\theta^n}\cos\theta^n$ and $r_{\theta^n}\sin\theta^n$ that are smoother than $\cos\theta^n$. Since the signal has been transferred to the θ coordinate, this problem is equivalent to extracting the component smoother than $\cos t$ for any given signal $f(t)$ in the physical space. In the remaining of this section, we will discuss what are the possible choices for us to define such projection operator P in the physical space t.

Projection via Fourier Transform. In the algorithm given in the previous section, we use the Fourier Transform to extract the components of $r_{\theta^n}\cos\theta^n$ and $r_{\theta^n}\sin\theta^n$ that are smoother than $\cos\theta^n$. In this case, the operator P is defined as follows:

$$\widehat{P(f)}(k) = \widehat{f}(k)\widehat{\phi}(k), \tag{56}$$

where ϕ is a low-pass filter. One example of such low-pass filter is given by (35).

This operator is very efficient, but the end effect (oscillation) emerges when applying this operator to non-periodic data due to the Gibbs phenomenon.

Projection via Fourier Transform with mirror reflection. In order to reduce the end effect, one commonly used technique is to extend the data first by the mirror reflection, then apply the Fourier Transform, i.e.

$$\widehat{P(f)}(k) = \widehat{f_e}(k)\widehat{\phi}(k), \tag{57}$$

where f_e is the extended data obtained by the mirror reflection which is defined as follows:

$$f_e(t) = \begin{cases} f(t), & t \geq 0, \\ f(-t), & t < 0. \end{cases} \tag{58}$$

Projection via the least square. As we mentioned earlier, the end effect is primarily caused by the use of the periodic Fourier basis to approximate a non-perodic function. In principle, one should use a basis that can represent non-periodic data accurately in order to alleviate the end effect. Based on this observation, we modify the definition of $V(\theta)$ to make it more suitable to represent non-periodic data. Specifically, we propose to use the following overcomplete Fourier basis:

$$V(\theta) = \text{span}\left\{1, \cos\left(\frac{k\theta}{2L_\theta}\right), \sin\left(\frac{k\theta}{2L_\theta}\right),\ k = 1, \cdots, 2\lambda L_\theta\right\}, \tag{59}$$

where $L_\theta = \frac{\theta(1)-\theta(0)}{2\pi}$ and $\lambda \leq 1/2$ is a parameter to control the smoothness of $V(\theta)$. To simplify the notation, we denote by $\mathbf{\Phi}_\theta$ the matrix whose columns are the overcomplete Fourier basis given by (59).

For this choice of $V(\theta)$, we can define the projection operator P through the least sqaure, i.e. $P_\theta(f) = \mathbf{\Phi}_\theta \mathbf{x}^*$ and

$$\mathbf{x}^* = \arg\min \|\mathbf{\Phi}_\theta \mathbf{x} - f\|_2 = \left(\mathbf{\Phi}_\theta^T \mathbf{\Phi}_\theta\right)^{-1} f. \tag{60}$$

A difficulty associated with this simple least square approach is that the matrix $\mathbf{\Phi}_\theta^T \mathbf{\Phi}_\theta$ becomes very ill-conditioned as the number of bases increases. In Fig. 15, we plot the condition number versus the number of bases. We can see that the conditional number increases very rapidly as the number of bases grows. It is about 10^{15} when the number of bases is 21. Thus the projection operator defined by the least square problem is not very useful in real applications.

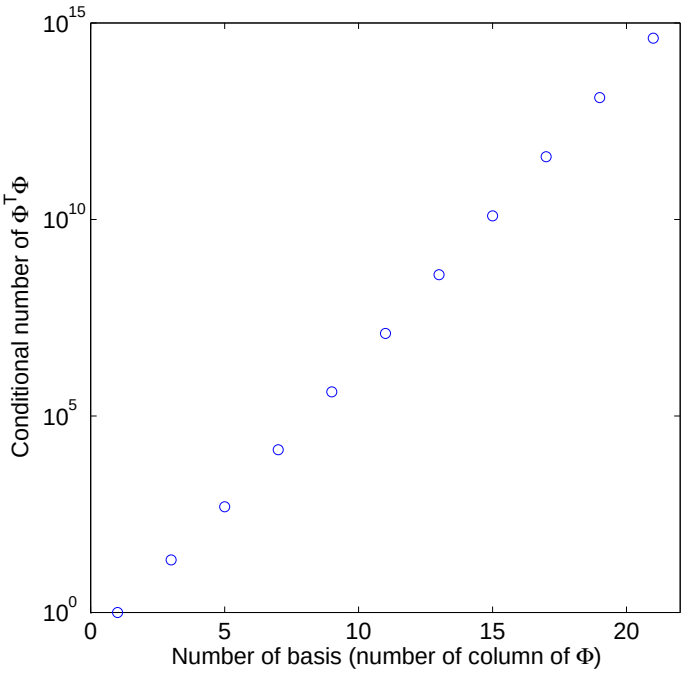

Figure 15: The condition number of $\mathbf{\Phi}^T\mathbf{\Phi}$ versus the number of bases.

Projection via a l^1 regularized least sqaure It is well known that the least square with ill-conditioned matrix can be stablized by adding a regularizing term in the objective functional. We can also use this idea to obtain a stable projection operator. Since we favor a sparse representation, a l^1 regularization is preferred. This gives rise to the following l^1 regularized least square problem:

$$\mathbf{x}^* = \arg\min \|\mathbf{\Phi}_\theta \mathbf{x} - f\|_{l^2} + \gamma \|\mathbf{x}\|_{l^1}, \tag{61}$$

where γ is a parameter which is given *a priori* and $\mathbf{\Phi}_\theta$ is a matrix consisting of the over-complete Fourier basis. The projection operator is defined as $P_\theta(f) = \mathbf{\Phi}_\theta \mathbf{x}^*$.

Next, we perform a numerical experiment to test the effectiveness of his projection operator. We use the following data in our experiment

$$\begin{aligned} &\theta_1 = 20\pi(t+1)^2 + 1, \quad \theta_2 = 161.4\pi t + 4(1-t)^2 \sin(16\pi t), \\ &f(t) = \frac{1}{1.5 + \sin(1.5\pi t)} + (2t+1)\cos\theta_1 + (2-t)^2 \cos\theta_2. \end{aligned} \tag{62}$$

In this numerical example, the parameter γ is chosen to be 1.

From Fig. 16, we observe that the projection operator defined by the l^1 regularized least square method seems to produce considerably smaller error near the boundary for this non-periodic signal.

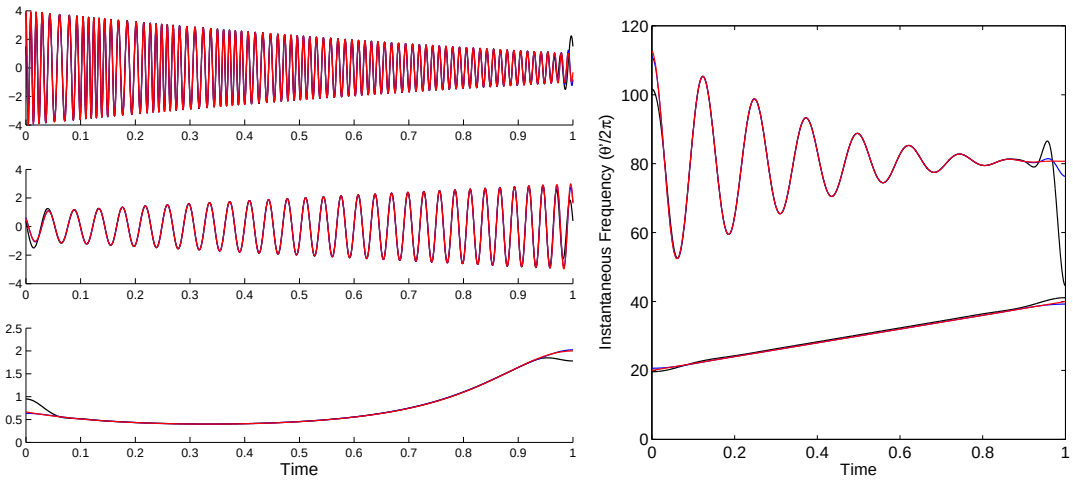

Figure 16: IMF (left) and Instantaneous frequency (right) of the signal in (62) obtained from different methods. Red: exact; Blue: l^1 regularized least square; Black: Fourier transform with mirror reflection.

The l^1 regularized least square problem has been extensively studied recently especially in the compressive sensing community. Although many efficient algorithms to solve the l^1 regularized least square were developed in recent years, the computational cost is still relatively high, especially for data with high frequency oscillations.

In order to reduce the computational cost, we combine this method with the approach based on the Fourier transform. This leads to the following hybrid method.

Projection based on a hybrid approach. First of all, we observe that the end effect is relatively localized. Thus, even for non-periodic data, we may still use the Fourier Transform to obtain an accurate decomposition in the interior region away from the boundary, see Fig. 17. This suggests that instead of solving the optimization problem (61) over the whole domain, we just need to solve the l^1 regularized least square problem over a local region near the boundary to correct the result given by the Fourier transform. We can then combine the two decompositions by using a partition of unit function to produce an accurate decomposition over the entire domain.

An essnetial ingredient of this hybrid method is to determine the width of the boundary region in the θ space. Let us denote the width of this boundary region by $2L\pi$. Typically, L is a small number determined by the width of the support of the low-pass filter used in the approach based on the Fourier Transform. We pick $L = 4$ in the following numerical example. Once L is chosen, the whole domain is divided to three parts: the left boundar area, $\Omega_L = [\theta(0), \theta(0) + 2L\pi)$; the right boundary area, $\Omega_R = (\theta(1) - 2L\pi, \theta(1)]$ and the interior part, $\Omega_I = [\theta(0) + 2L\pi, \theta(1) - 2L\pi]$.

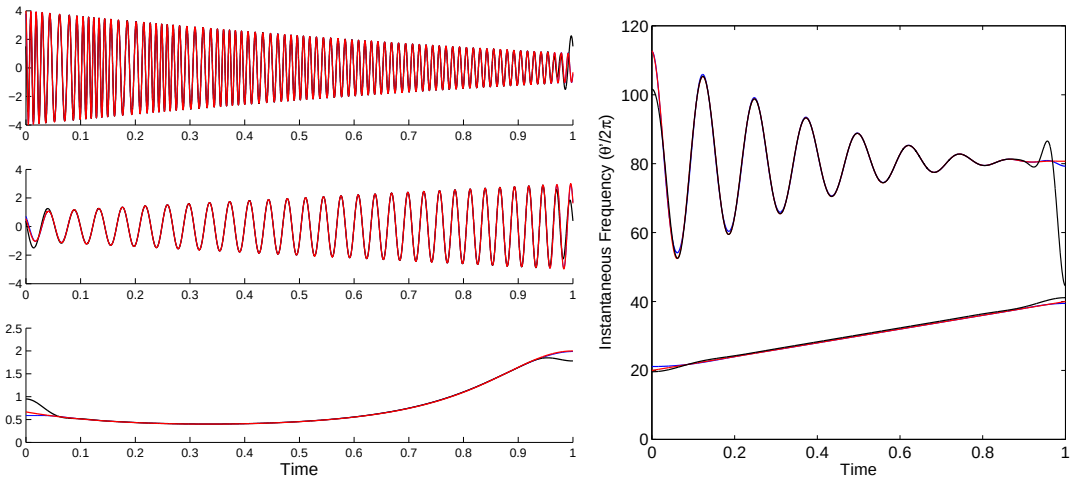

Figure 17: IMF (left) and Instantaneous frequency (right) of the signal in (62) obtained from different methods. Red: exact; Blue: FFT-based algorithm with the local correction; Black: FFT-based algorithm with mirror reflection.

In Ω_I, the projection operator is defined by the Fourier Transform, which can be computed very efficiently. In Ω_L and Ω_R, we use the l^1 regularized least sqaure to define the projection operator. In each of these areas, the span of θ is $2L\pi$. Thus, the number of bases is $2L+1$, which is a small number. Then in each of the boundary regions, the computational cost to solve the l^1 regularized least square problem is relatively small.

In Fig. 17, we apply the above hybrid method to the data given by (62). After applying the local l^1 regularized least sqaure correction, the large error near the boundary induced by the Fourier Transform projection has been significantly reduced.

Projection via boundary wavelets. We may also apply wavelets to deal with non-periodic data. The space $V(\theta)$ can be constructed by the scaling function and the corresponding wavelet ψ in the following way:

$$V(\theta) = \text{span}\left\{\phi\left(\frac{\theta(t)}{\omega} - n\right),\ n \in \mathbb{Z}\right\}, \tag{63}$$

where ω is the central frequency of the wavelet ψ, which is a constant.

We borrow some ideas of constructing boundary wavelets in a finite domain. In the boundary wavelet approach, one needs to construct special wavelet basis near the boundary so that these boundary wavelets and the interior wavelets together form an orthogonal basis and give an accurate approximation to the signal over the whole interval [8].

We remark that how to control the error due to the end effect is still an on-going research. Here we just give a brief introduction to some potential ideas. Further study of this topic will be reported in our subsequent paper.

6 Some preliminary error analysis

In this section, we perform some preliminary error analysis for our nonlinear matching pursuit method. To guarantee uniquness of the deocmposition, we need to impose certain scale separation assumptions for the data that we try to decompose. Before we state our result, we first define what we mean by scale separation for a given signal.

Definition 6.1 (Scale-seperation) *One function $f(t) = a(t)\cos\theta(t)$ is said to satisfy a scale-seperation property with a separation factor $\epsilon > 0$, if $a(t)$ and $\theta(t)$ satisfy the following conditions:*

$$\begin{aligned}
&a(t) \in C^1(\mathbb{R}),\ \theta \in C^2(\mathbb{R}),\\
&\inf_{t\in\mathbb{R}} \theta'(t) > 0, \quad M = \sup_{t\in\mathbb{R}} |\theta''(t)| < \infty\\
&\left|\frac{a'(t)}{\theta'(t)}\right|,\ \left|\frac{\theta''(t)}{(\theta'(t))^2}\right| \le \epsilon, \quad \forall t \in \mathbb{R}.
\end{aligned}$$

Definition 6.2 (Well-seperated signal) *A signal $f : \mathbb{R} \to \mathbb{R}$ is said to be well-seperated with separation factor ϵ and frequency ratio $d > 1$ if it can be written as*

$$f(t) = \sum_{k=1}^{K} a_k(t)\cos\theta_k(t)$$

where all $f_k(t) = a_k(t)\cos\theta_k(t)$ satisfies the scale-seperation property with separation factor ϵ, and their phase function θ_k satisfies

$$\theta_k'(t) \ge d\theta_{k-1}'(t), \quad \forall t \in \mathbb{R}. \tag{64}$$

Theorem 6.1 *Let $f(t)$ be a function satisfying the scale-seperation property with separation factor ϵ and frequency ratio d as defined in Definition 6.2. Choose a low-pass filter ϕ such that its Fourier Transform $\widehat{\phi}$ has support in $[-\Delta, \Delta]$ with $\Delta < \frac{d-1}{d+1/2}$ and $\widehat{\phi}(k) = 1,\ \forall k \in [-\Delta/2, \Delta/2]$. If in the nth step, the approximate phase function $\theta_{k_0}^n$ satisfies the following condition:*

$$\left|\frac{\left(\theta_{k_0}^n\right)'(t)}{\theta_{k_0}'(t)} - 1\right| < \frac{\Delta}{2}, \quad \left|\frac{\left(\theta_{k_0}^n\right)''(t)}{\left(\left(\theta_{k_0}^n\right)'(t)\right)^2}\right| \le \epsilon, \tag{65}$$

then the accuracy of the phase function in the next step is order ϵ, i.e.

$$\left|\theta_{k_0}^{n+1}(t) - \theta_{k_0}(t)\right| = O(\epsilon). \tag{66}$$

In order to prove the above theorem, we need the following lemma:

Lemma 6.1 *For any $a(t) \in C^1(\mathbb{R})$, $\theta \in C^2(\mathbb{R})$, we have*

$$\left| \int a(\tau) e^{-i\theta(\tau)} \, \phi(\tau - t) d\tau - a(t) e^{-i\theta(t)} \widehat{\phi}\left(\theta'(t)\right) \right| \leq \sup |a'(t)| I_1 + \frac{1}{2} |a(t)| \sup |\theta''(t)| I_2, \quad (67)$$

where $I_n = \int |t^n \phi(t)| dt$.

Proof The proof follows from the following direct calculations:

$$\begin{aligned}
& \left| \int a(\tau) e^{-i\theta(\tau)} \, \phi(\tau - t) d\tau - a(t) e^{-i\theta(t)} \widehat{\phi}\left(\theta'(t)\right) \right| \\
= & \left| \int (a(\tau) - a(t)) \, e^{i\theta(\tau)} \, \phi(\tau - t) d\tau + a(t) \int \left(e^{-i\theta(\tau)} - e^{-i(\theta(t) - \theta'(t)(\tau - t))} \right) \, \phi(\tau - t) d\tau \right| \\
= & \left| \int (a(\tau) - a(t)) \, e^{i\theta(\tau)} \, \phi(\tau - t) d\tau + a(t) \int \left(e^{-i(\theta(\tau) - \theta(t) - \theta'(t)(\tau - t))} - 1 \right) e^{-i(\theta(t) - \theta'(t)(\tau - t))} \, \phi(\tau - t) d\tau \right| \\
\leq & \sup |a'(t)| \int |\tau \phi(\tau)| d\tau + |a(t)| \left| \int \left(e^{-\frac{1}{2} i \theta''(s(\tau))(\tau - t)^2} - 1 \right) e^{-i(\theta(t) + \theta'(t)(\tau - t))} \phi(\tau - t) d\tau \right| \\
\leq & \sup |a'(t)| \int |\tau \phi(\tau)| d\tau + |a(t)| \int \left| \frac{1}{2} \theta''(s(\tau))(\tau - t)^2 \phi(\tau - t) \right| d\tau \\
\leq & \sup |a'(t)| \int |\tau \phi(\tau)| d\tau + \frac{1}{2} |a(t)| \sup |\theta''(t)| \int \left| \tau^2 \phi(\tau) \right| d\tau \\
= & \sup |a'(t)| I_1 + \frac{1}{2} |a(t)| \sup |\theta''(t)| I_2 \qquad (68)
\end{aligned}$$

□

Remark 6.1 *We remark that since we typically deal with data of finite support and extend them periodically to the whole domain, the estimates for I_1 and I_2 in the above lemma are effectively taken only in the finite support of the data.*

Corollary 6.1 *If the Fourier Transform of the low-pass filter ϕ is symmetic, i.e. $\widehat{\phi}(k) = \widehat{\phi}(-k)$, then we have*

$$\left| \int a(\tau) \cos\left(\theta(\tau)\right) \phi(\tau - t) d\tau - a(t) \cos\theta(t) \widehat{\phi}\left(\theta'(t)\right) \right| \leq \sup |a'(t)| I_1 + \frac{1}{2} |a(t)| \sup |\theta''(t)| I_2 \quad (69)$$

$$\left| \int a(\tau) \sin\left(\theta(\tau)\right) \phi(\tau - t) d\tau - a(t) \sin\theta(t) \widehat{\phi}\left(\theta'(t)\right) \right| \leq \sup |a'(t)| I_1 + \frac{1}{2} |a(t)| \sup |\theta''(t)| I_2. \quad (70)$$

Now we can prove Theorem 6.1. Here we only give a sketch of the proof, the detail of the proof is deferred to the Appendix.

Proof of Theorem 6.1: In order to simplify the notation, we denote $\overline{\theta} = \theta_{k_0}^n$ and and define $\overline{(\cdot)}$ as the mapping from t to $\overline{\theta}$, i.e. $\overline{f}(\overline{\theta}) = f(t), \quad \forall f$. In our algorithm, we update $\theta_{k_0}^{n+1}$ by the following step

$$\theta_{k_0}^{n+1} = \overline{\theta} - \arctan\left(\frac{b(t)}{a(t)}\right), \quad a(t) = A(\overline{\theta}(t)), \quad b(t) = B(\overline{\theta}(t)), \quad (71)$$

where

$$A(\gamma) = 2\int \overline{f}(\overline{\theta})\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}, \quad B(\gamma) = 2\int \overline{f}(\overline{\theta})\sin(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}. \tag{72}$$

We will prove that $a(t)$ and $b(t)$ satisfy the following estimates:

$$\begin{aligned} a(t) &= A\left(\overline{\theta}(t)\right) = a_{k_0}(t)\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). & (73)\\ b(t) &= B\left(\overline{\theta}(t)\right) = a_{k_0}(t)\sin\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). & (74)\end{aligned}$$

Then, we can get

$$\Delta\theta = \arctan\left(\frac{B(t)}{A(t)}\right) = \theta_{k_0}(t)-\overline{\theta}+O(\epsilon), \tag{75}$$

which implies that

$$\left|\theta_{k_0}^{n+1}(t)-\theta_{k_0}(t)\right| = O(\epsilon). \tag{76}$$

First, let us estimate $A(\gamma)$ as follows:

$$\begin{aligned} A(\gamma) &= 2\int \overline{f}(\overline{\theta})\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta} = 2\sum_{k=1}^{n}\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}\\
&= \sum_{k=1}^{n}\int \overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)+\overline{\theta}\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}+\sum_{k=1}^{n}\int \overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)-\overline{\theta}\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}\\
&= \sum_{k=1}^{n}\int \overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)+\overline{\theta}\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}+\sum_{k\neq k_0}\int \overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)-\overline{\theta}\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}\\
&\quad +\int \overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}\\
&= I+II+III. \end{aligned} \tag{77}$$

For I and II, the scale seperation assumption of the data implies that $\overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)+\overline{\theta}\right)$ and $\overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)-\overline{\theta}\right)$ $(k\neq k_0)$ are more oscillatory than the kernel $\phi(\overline{\theta})$. Thus we expect that these two terms are small after convoluting with a smooth kernel. In fact, we can prove that $I, II = O(\epsilon)$ by Corollary 6.1.

The estimate for III is more difficult. By our assumption, we have $\left|\frac{\left(\theta_{k_0}^n\right)'(t)}{\theta_{k_0}'(t)}-1\right| < \frac{\Delta}{2}$. This implies that $\overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)$ is smoother than the kernel $\phi(\overline{\theta})$. Using Corollary 6.1, we can prove that $III = \overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon)$. By combining these results, we get

$$a(t) = A\left(\overline{\theta}(t)\right) = a_{k_0}(t)\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). \tag{78}$$

Similarly, we can prove the estimate for $b(t)$

$$b(t) = B\left(\overline{\theta}(t)\right) = a_{k_0}(t)\sin\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). \tag{79}$$

This completes the proof. □

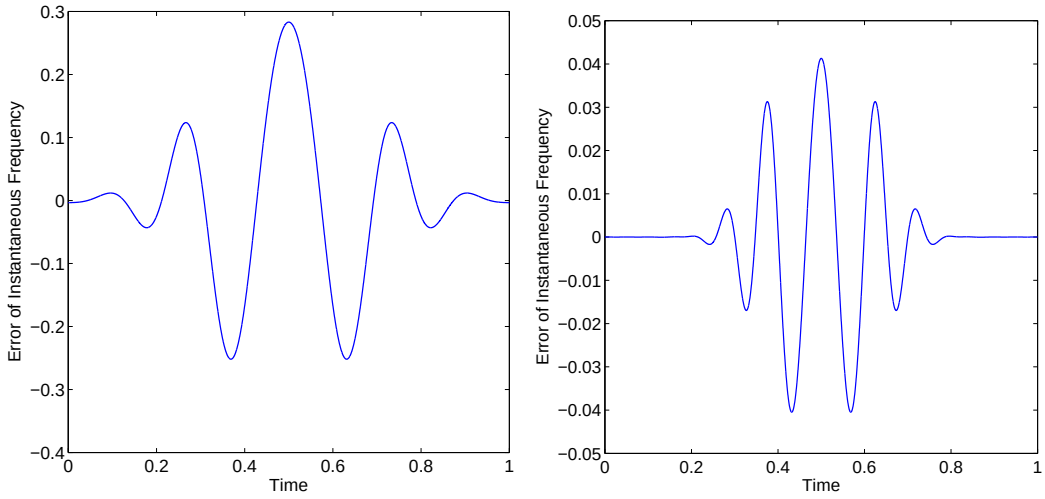

Figure 18: Error of the Instantaneous frequency with different factors of scale seperation. Left: poor scale seperation; Right: good scale seperation.

Remark 6.2 *Under the same assumption, we can prove that the error of the frequency is also order ϵ, i.e.*

$$\left| \left(\theta_{k_0}^{n+1}\right)'(t) - \theta_{k_0}'(t) \right| = O(\epsilon). \tag{80}$$

The argument is almost the same as the above proof, except that the calculation is a little more involved.

From the above theorem, we can see that the accuracy of our method depends on the factor of scale seperation. This is also consistent with our numerical results. In the following numerical example, we compare the error of the instantaneous frequency obtained by our method for two different signals. One has poor scale seperation, the other one has better scale seperation. The signals are given by (81). The signal f_2 has a better scale seperation property than f_1 since its instantaneous frequency is twice of that of f_1. As shown in Fig. 18, the error we obtainn for f_2 is considerably smaller than that of f_1.

$$\begin{aligned} a_0(t) = a_1(t) = \frac{1}{1.1 + \cos(2\pi t)}, \quad \theta = 10\sin(2\pi t) + 40\pi t. \\ f_1(t) = a_0(t) + a_1(t)\cos\theta(t), \quad f_2(t) = a_0(t) + a_1(t)\cos(2\theta(t)). \end{aligned} \tag{81}$$

We would like to point out that the error estimate given by Theorem 6.1 is highly over-estimated. In Fig. 18, we can see that even for the signal with a poor scale seperation property, the instantaneous frequency we obtain is still reasonably accurate, although the correspondingi scale separation factor $\epsilon \approx 1.4$ is quite big according to Definition 6.1).

In the estimate of Lemma 6.1, instead of taking the supremem over $\mathbb{R}$, we can take supremem over a finite interval, since we can choose a low-pass filter ϕ that decays exponentially fast. Then, we can get a more local estimate:

$$\left| \int a(\tau) e^{i\theta(\tau)} \, \phi(\tau - t) d\tau - a(t) e^{i\theta(t)} \widehat{\phi} \left(\theta'(t) \right) \right| \leq \sup_{t \in S_\phi} |a'(t)| I_1 + \frac{1}{2} |a(t)| \sup_{t \in S_\phi} |\theta''(t)| I_2, \quad (82)$$

where $S_\phi = \{t \in \mathbb{R} : |\phi(t)| > \epsilon\}$.

This seems to suggest that for the signal that does not have a good scale separation property in some region, its influence on the accuracy of the decomposition is limited to that region. This can be also seen in Fig. 18. In this example, the scales are not well seperated in the center of the interval. However, the scales are better separated near the two ends. We can see that the error near the boundary is much smaller than that in the center.

From this analysis, we can also see that the low-pass filter with smooth Fourier spectrum (such as the cosine function given by Fig. 2) would perform better than that with discontinueous spectrum (such as the funtion given by Fig. 1) in terms of maintaining the temporal locality property of the decomposition. The low-pass filter with discontinueous spectrum decays much slower in the time domain due to the Gibbs phenomena. This is also the reason we use the cosine low-pass filter instead of the stair one.

Theorem 6.1 tells us that if we have a good initial guess, then there is no need to do iteration. But in most cases, we have onlya rough initial guess, and the condition in Theorem 6.1 may not be satisfied. In this case, the iterative procedure in our algorithm improves the result gradually as the number of iterations increases. It will provide a good approximation after a number of iterations. In Fig. 19, we show how the iteration can improve the approximation of the instantaneous frequency for a simple chirp signal: $f(t) = \cos(10\pi(3t+1)^2)$.

In this example, the initial guess for the instantaneous frequency is a constant, $\theta_0 = 80\pi t$. Near the intersection of $80\pi t$ and the exact instantaneous frequency, the initial guess is relatively good. After one step, we can get a better approximation in this local region. This new estimate gives us a better guess for the instantaneous frequency in a slightly larger interval that contains the good interval given by the initial guess. Then in the next step, we can get an estimate which gives an good approximation in a even larger interval. Grandually, we can get an accurate approximation of the instantaneous frequency in the whole interval. Fig. 19 plots the approximate instantaneous frequency in different steps. As the number of iteration increases, the region in which we have a good approximation becomes larger and larger. Finally, the iterative algorithm produces an accurate instantaneous frequency in the entire domain.

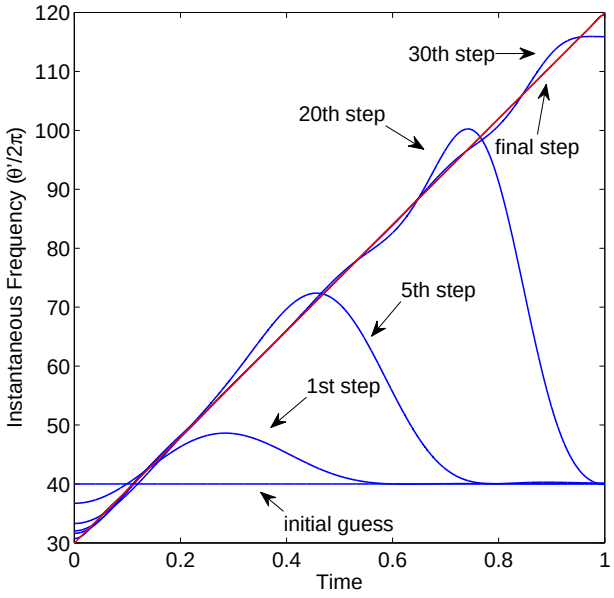

Figure 19: Instantaneous frequency at different step.

7 Nonlinear matching pursuit using other norms

Typically, the matching pursuit is based on minimizing the l^2 norm of the residual, but we can also use other norms.

Corresponding to the method based on $TV^{(3)}$ minimization proposed in our previous paper [22], if we change the l^2 norm in the matching pursuit to the $TV^{(3)}$ norm, we would get the following method:

Initialization: $\theta^0 = \theta_0$.

Main Iteration:

Step 1: Update $a_0^n,\ a_1^n,\ b_1^n$ by solving the following linear optimization problem:

$$\begin{aligned} \text{Minimize} \quad & TV^3(a_0^n), && (83)\\ \text{Subject to :} \quad & a_0^n + a_1^n \cos\theta^{n-1}(t) + b_1^n \sin\theta^{n-1}(t) = f(t) \\ & a_1^n,\ b_1^n \in V(\theta^{n-1}). && (84) \end{aligned}$$

Step 2: Update the phase function θ:

$$\theta^n = \theta^{n-1} - \mu \arctan\left(\frac{b_1^n}{a_1^n}\right), \tag{85}$$

where $\mu \in [0,1]$ is chosen to enforce that θ^n is an increasing function:

$$\mu = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta_k^{n-1} - \alpha \arctan\left(\frac{b_1^n}{a_1^n}\right)\right) \geq 0\right\}. \tag{86}$$

Step 3: If $\|\theta^n - \theta^{n-1}\|_2 \leq \epsilon_0$, stop. Otherwise, go to Step 1.

In this method, the smoothness of the envelope $a_1(t)$ is enforced by the space $V(\theta)$ instead of the Lagrangian multiplier which was used in our previous paper [22]. This is the only difference between these two methods. The performance of these two methods turns out to be very similar.

We can also change $TV^{(3)}$ norm to any other norm in this framework based on our knowledge of the signal that we try to decompose. One advantage of the framework that we propose in this paper is the flexibility that it offers. If you have any *a priori* information of the data, it is easy to produce a method specially taylored to this kind of data. For example, if we know that the local median a_0, envelope a_1, and the phase function θ of the data have a sparse representation in some given basis $\Phi_0, \Phi_1, \Phi_\theta$, then we can exploit this kind of sparsity by minimizing the l^1 norm of the coefficients of a_0 and a_1 when projected into this basis. In this case, the corresponding algorithm becomes:

Initialization: $\theta^0 = \theta_0$.

Main Iteration:

Step 1: Get $a_0^n,\ a_1^n,\ b_1^n$ by solving following linear optimization problem:

$$\text{Minimize} \qquad \|\widehat{a}_0^n\|_{l^1} + \|\widehat{a}_1^n\|_{l^1} + \|\widehat{b}_1^n\|_{l^1}, \tag{87}$$

$$\text{Subject to :} \qquad a_0^n + a_1^n \cos\theta^{n-1}(t) + b_1^n a_1^{n-1} \sin\theta^{n-1}(t) = f(t). \tag{88}$$

where $\widehat{a}_0^n$, $\widehat{a}_1^n$ and $\widehat{b}_1^n$ are coefficients of the representations of $a_0^n,\ a_1^n,\ b_1^n$,

$$a_0^n = \Phi_0 \widehat{a}_0^n, \quad a_1^n = \Phi_1 \widehat{a}_1^n, \quad b_1^n = \Phi_\theta \widehat{b}_1^n \tag{89}$$

Step 2: Update the phase function θ:

$$\theta^n = \theta^{n-1} - \mu b_1^n, \tag{90}$$

where $\mu \in [0, 1]$ is chosen to make θ^n increasing

$$\mu = \max\left\{\alpha \in [0,1] : \frac{d}{dt}\left(\theta^{n-1} - \alpha b_1^n\right) \geq 0\right\}. \tag{91}$$

Step 3: If $\|\theta^n - \theta^{n-1}\|_{l^2} \leq \epsilon_0$, stop. Otherwise, go to Step 1.

We have analyzed the convergence property of the above algorithm under some sparsity assumptions that are similar in spirit to the compressed sensing method. We will report this result in a subsequent paper.

8 Conclusion

In this paper, we introduce a new adaptive data analysis method based on the nonlinear matching pursuit. The adaptivity of our decomposition is obtained by looking for the sparsest representation of signals in the time-frequency domain from a largest possible dictionary that consists of all possible candidates for intrinsic mode functions (IMFs). Solving this nonlinear optimization problem is in general very difficult. We propose a nonlinear matching pursuit method to solve this nonlinear optimization problem by generalizing the classical matching pursuit for the l^0 optimization problem. One important advantage of this nonlinear matching pursuit method is it can be implemented by FFT with computational complexity of order $O(N \log N)$, where N is the number of data sample points. Further, this approach is very stable to noise. For non-periodic data, we propose two methods to overcome the difficulty associated with the end effect. For datat with good scale separation property, these methods seem to offer accurate decompositions up to the boundary.

We have also carried out some preliminary theoretical study for the nonlinear optimization method proposed in this paper. In the case when the signal satisfies certain scale separation conditions, we show that our iterative algorithm converges to an approximate decomposition with the accuracy determined by the scale separation factor of the signal.

There are some remaining issues to be studied in the future. How to control the error resulting from the end effect is still a big challenge, especially for non-periodic data. We plan to further develop our method to reduce the error due to the end effect for non-periodic data that do not have a good scale separation property. Another direction is to generalize this adaptive data analysis method to high dimensional data. In some physical applications such as propagation of nonlinear ocean waves, each wave form has a dominating propagation direction. In this case, our method has a natural generalization by adopting a multi-dimensional phase function.

Appendix

Proof of Theorem 6.1: In order to simplify the notation, we denote $\overline{\theta} = \theta_{k_0}^n$ and use $\overline{(\cdot)}$ to represente the mapping from t to $\overline{\theta}$, i.e. $\overline{f}(\overline{\theta}) = f(t), \quad \forall f$.

According to our algorithm, we update $\theta_{k_0}^{n+1}$ as follows:

$$\theta_{k_0}^{n+1} = \overline{\theta} - \arctan\left(\frac{b(t)}{a(t)}\right), \quad a(t) = A(\overline{\theta}(t)), \quad b(t) = B(\overline{\theta}(t)), \tag{92}$$

where

$$A(\gamma) = 2\int \overline{f}(\overline{\theta})\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}, \quad B(\gamma) = 2\int \overline{f}(\overline{\theta})\sin(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}. \tag{93}$$

We first estimate $A(\gamma)$ as follows:

$$A(\gamma) = 2\int \overline{f}(\overline{\theta})\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta} = 2\sum_{k=1}^{n}\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}.$$

For $k \neq k_0$, we have

$$\begin{aligned} & 2\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta} \\ = & \int \overline{a}_k(\overline{\theta})\left(\cos\left(\theta_k(t)+\overline{\theta}\right)+\cos\left(\theta_k(t)-\overline{\theta}\right)\right)\phi\left(\overline{\theta}-\gamma\right)d\overline{\theta}. \end{aligned} \tag{94}$$

Since

$$\left|\frac{d\overline{a}_k(\overline{\theta})}{d\overline{\theta}}\right| = \left|\frac{a_k'(t)}{\overline{\theta}'(t)}\right| \leq \epsilon\left|\frac{\theta_k'(t)}{\overline{\theta}'(t)}\right|, \tag{95}$$

we obtain

$$\left|\frac{d^2}{d\overline{\theta}^2}\left(\theta_k(t)\pm\overline{\theta}\right)\right| = \left|\frac{\theta_k''(t)\overline{\theta}'(t)-\theta_k'(t)\overline{\theta}''(t)}{\left(\overline{\theta}'(t)\right)^3}\right| \leq \epsilon\left|\frac{\left(\theta_k'(t)\right)^2-\theta_k'(t)\overline{\theta}'(t)}{\left(\overline{\theta}'(t)\right)^2}\right|, \quad \text{for } k\neq k_0. \tag{96}$$

Now we apply Corollary 6.1 for $k \neq k_0$ to obtain

$$\begin{aligned} & 2\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\phi\right)d\overline{\theta} \\ = & \overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)+\overline{\theta}\right)\widehat{\phi}\left(\frac{\theta_k'(t)}{\overline{\theta}'(t)}+1\right)+\overline{a}_k(\overline{\theta})\cos\left(\theta_k(t)-\overline{\theta}\right)\widehat{\phi}\left(\frac{\theta_k'(t)}{\overline{\theta}'(t)}-1\right)+O(\epsilon). \end{aligned} \tag{97}$$

Using the condition $\Delta < \frac{d-1}{d+1/2}$, $\theta_k'(t) > d\theta_{k-1}'(t)$ and $\left|\frac{\overline{\theta}'(t)}{\theta_{k_0}'(t)}-1\right| < \frac{\Delta}{2}$, we get

$$\frac{\theta_k'(t)}{\overline{\theta}'(t)}-1 = \frac{\theta_k'(t)}{\theta_{k_0}(t)}\frac{\theta_{k_0}'(t)}{\overline{\theta}'(t)}-1 > d(1-\Delta/2)-1 > \Delta, \quad \text{if } k>k_0 \tag{98}$$

$$\frac{\theta_k'(t)}{\overline{\theta}'(t)}-1 = \frac{\theta_k'(t)}{\theta_{k_0}(t)}\frac{\theta_{k_0}'(t)}{\overline{\theta}'(t)}-1 < (1+\Delta/2)/d-1 < -\Delta, \quad \text{if } k<k_0 \tag{99}$$

$$\frac{\theta_k'(t)}{\overline{\theta}'(t)}+1 > 1 > \Delta. \tag{100}$$

Since the support of $\widehat{\phi}$ is within $[-\Delta,\Delta]$, we have

$$2\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\phi\right)d\overline{\theta} = O(\epsilon). \tag{101}$$

For $k = k_0$, we proceed as follows

$$\begin{aligned} & 2\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\phi\right)d\overline{\theta} \\ = & \int \overline{a}_{k_0}(\overline{\theta})\left(\cos\left(\theta_{k_0}(t)+\overline{\theta}\right)+\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)\right)\phi\left(\overline{\theta}-\phi\right)d\overline{\theta}. \end{aligned} \tag{102}$$

Similarly, by using the assumption

$$\left|\frac{d\overline{a}_{k_0}(\overline{\theta})}{d\overline{\theta}}\right| = \left|\frac{a'_{k_0}(t)}{\overline{\theta}'(t)}\right| \leq \epsilon \left|\frac{\theta'_{k_0}(t)}{\overline{\theta}'(t)}\right| \tag{103}$$

we obtain the following estimates:

$$\left|\frac{d}{d\overline{\theta}}\left(\theta_{k_0}(t)+\overline{\theta}\right)\right| = \left|\frac{\theta'_{k_0}(t)}{\overline{\theta}'(t)}+1\right| > 1 > \Delta, \tag{104}$$

$$\left|\frac{d}{d\overline{\theta}}\left(\theta_{k_0}(t)-\overline{\theta}\right)\right| = \left|\frac{\theta'_{k_0}(t)}{\overline{\theta}'(t)}-1\right| < \frac{\Delta}{2}, \tag{105}$$

$$\left|\frac{d^2}{d\overline{\theta}^2}\left(\theta_{k_0}(t)\pm\overline{\theta}\right)\right| = \left|\frac{\theta''_{k_0}(t)\overline{\theta}'(t)-\theta'_{k_0}(t)\overline{\theta}''(t)}{\left(\overline{\theta}'(t)\right)^3}\right| \leq \epsilon \left|\frac{\left(\theta'_{k_0}(t)\right)^2-\theta'_{k_0}(t)\overline{\theta}'(t)}{\left(\overline{\theta}'(t)\right)^2}.\right|. \tag{106}$$

By applying Corollary 6.1 again, we get

$$\begin{aligned} & 2\int \overline{a}_k(\overline{\theta})\cos\theta_k(t)\cos(\overline{\theta})\phi\left(\overline{\theta}-\phi\right)d\overline{\theta} \\ = \;& \overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)+\overline{\theta}\right)\widehat{\phi}\left(\frac{\theta'_{k_0}(t)}{\overline{\theta}'(t)}+1\right)+\overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)\widehat{\phi}\left(\frac{\theta'_{k_0}(t)}{\overline{\theta}'(t)}-1\right)+O(\epsilon) \\ = \;& \overline{a}_{k_0}(\overline{\theta})\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). \end{aligned} \tag{107}$$

Finally, we get the following estimate for $a(t)$,

$$a(t) = A\left(\overline{\theta}(t)\right) = a_{k_0}(t)\cos\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). \tag{108}$$

For $b(t)$, we can obtain a similar estimate

$$b(t) = B\left(\overline{\theta}(t)\right) = a_{k_0}(t)\sin\left(\theta_{k_0}(t)-\overline{\theta}\right)+O(\epsilon). \tag{109}$$

Thus, we have

$$\Delta\theta = \arctan\left(\frac{B(t)}{A(t)}\right) = \theta_{k_0}(t)-\overline{\theta}+O(\epsilon), \tag{110}$$

which implies that

$$\left|\theta^{n+1}_{k_0}(t)-\theta_{k_0}(t)\right| = O(\epsilon). \tag{111}$$

This completes the proof. □

Acknowledgments.

This work was in part supported by the AFOSR MURI grant FA9550-09-1-0613. We would like to thank Professors Norden E. Huang and Zhaohua Wu for many stimulating discussions on EMD/EEMD and topics related to the research presented here. We would also like to thank Professors Ingrid Daubechies, Stanley Osher, and Zuowei Shen for their interest in this work and for a number of valuable discussions. Prof. Hou would like to express his gratitude to the National Central University (NCU) for their support and hospitality during his visits to NCU in the past two years.

References

[1] E. Bedrosian, *A product theorem for Hilbert transforms,* Proc. IEEE., **51** (1963), pp. 868-869.

[2] B. Boashash, *Time-Frequency Signal Analysis: Methods and Applications.* Longman-Cheshire, Melbourne and John Wiley Halsted Press, New York, 1992.

[3] A. M. Bruckstein, D. L. Donoho, M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images,* SIAM Review, **51** (2009), pp. 34-81.

[4] E. Candès and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Trans. on Information Theory*, **52(12)** (2006), pp. 5406-5425.

[5] E. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal recovery from highly incomplete frequency information,* IEEE Trans. Inform. Theory, **52** (2006), pp. 489-509.

[6] E. Candes, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements,* Comm. Pure and Appl. Math., **59** (2006), pp. 1207-1223.

[7] L. Cohen, Generalized phase-space distribution functions, *J. Math. Phys.*, **7**, pp. 781-786, 1966.

[8] A. Cohen, I. Daubechies and P. Vial, Wavelet bases on the interval and fast algorithms, *J. Appl. Comput. Harmon. Anal.*, **1** (1993), pp. 54-81.

[9] S. Chen, D. Donoho and M. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.*, **20**, pp. 33-61, 1998.

[10] R. R. Coifman, Y. Meyer and M. V. Wickerhauser, Wavelet analysis and signal processing, *Wavelets and Theire Applications*, M. B. Ruskai et al., Eds., Jones and Bartlett, pp. 163-178, 1992.

[11] R. R. Coifman and M. V. Wickerhuaser, Entropy-based algorithms for best-basis selection, *IEEE Trans. on Information Theory*, **38**, pp. 713-718, 1992.

[12] L. Cohen, *Time-Frequency Analysis.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[13] I. Daubechies, Time-frequency localization operators: a geometric phase space approach, *IEEE Trans. on Information Theory*, **34**, pp. 605-612, 1988.

[14] I. Daubechies, *Ten Lectures on Wavelets,* CBMS-NSF Regional Conference Series on Applied Mathematics, Vol. 61, SIAM Publications, 1992.

[15] I. Daubechies, J. Lu and H. Wu, Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool, Appl. Comp. Harmonic Anal. **30** (2011), pp. 243-261.

[16] D. L. Donoho, *Compressed sensing,* IEEE Trans. Inform. Theory, **52** (2006), pp. 1289-1306.

[17] P. Flandrin, *Time-Frequency/Time-Scale Analysis.* Academic Press, San Diego, CA, 1999.

[18] R. S. Gross, 2001 Combinations of Earth orientation measurements: SPACE2000, COMB2000, and POLE2000. JPL Publication 01-2. Jet Propulsion Laboratory, Pasadena, CA.

[19] N. E. Huang et al., A confidence limit for the empirical mode decomposition and the Hilbert spectral analysis, *Proc. of Roy. Soc. London*, **459A**, pp. 2317-2345, 2003.

[20] D. Gabor, *Theory of communication*, J. IEE. **93** (1946), pp. 426-457.

[21] Tom Goldstein and Stanley Osher, The Split Bregman Method for L_1-Regularized Problems, *SIAM J. Imaging Sci.*, **2(2)** (2009), pp. 323-343.

[22] T. Y. Hou and Z. Shi, *Adaptive Data Analysis via Sparse Time-Frequency Representation. Advances in Adaptive Data Analysis*, submitted.

[23] N. E. Huang et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. R. Soc. Lond. A*, **454** (1998), pp. 903-995.

[24] N. E. Huang, Z. Shen, and S. R. Long, *A New View of Nonlinear Water Waves - The Hilbert Spectrum,* Ann. Rev. Fluid Mech., **31** (1999), pp. 417-457.

[25] N. E. Huang, Z. Wu, S. R. Long, K.C. Arnold, X. Chen, and K. Blank, *On Instantaneous Frequency.* Advances in Adaptive Data Analysis. **1(2)** (2009), pp. 177-229.

[26] D. L. Jomes and T. W. Parks, A high resolution data-adaptive time-frequency representation, *IEEE Trans. Acoust. Speech Signal Process*, **38** (1990), pp. 2127-2135.

[27] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale L^1-regularized least squares,* IEEE J. of Selected Topics in Signal Processing, **1 (4)** (2007), pp. 606-617.

[28] P. J. Loughlin and B. Tracer, *On the amplitude - and frequency-modulation decomposition of signals,* J. Acoust. Soc. Am., **100** (1996), pp. 1594-1601.

[29] B. C. Lovell, R. C. Williamson and B. Boashash, *The relationship between instantaneous frequency and timefrequency representations*, IEEE Trans. Signal Process. **41(3)** (1993), pp. 1458-1461.

[30] S. Mallat and Z. Zhang, Matching pursuit with time-frequency dictionaries, *IEEE Trans. Signal Process*, **41**, pp. 3397-3415, 1993.

[31] S. Mallat, A wavelet tour of signal processing: the Sparse way, Academic Press, 2009.

[32] W. K. Meville, *Wave modulation and breakdown*, J. Fluid Mech., **128** (1983), pp. 489-506.

[33] A. H. Nuttall, *On the quadrature approximation to the Hilbert transform of modulated signals,* Proc. IEEE. **54** (1966), pp. 1458-1459.

[34] S. Olhede and A. T. Walden, *The Hilbert spectrum via wavelet projections,* Proc. Roy. Soc. London A, **460** (2004), pp. 955-975.

[35] B. Picinbono, *On instantaneous amplitude and phase signals,* IEEE Trans. Signal Process., **45** (1997), pp. 552-560.

[36] S. Qian and D. Chen, Joint Time-Frequency Analysis: Methods and Applications, Prentice Hall, 1996.

[37] S. Qian and D. Chen, Signal representation using adaptive normalized Gaussian functions, *Signal Processing*, **36** pp. 1-11, 1994.

[38] S. O. Rice, *Mathematical analysis of random noise*, Bell Syst. Tech. J., **23** (1944), pp. 282-310.

[39] J. Shekel, *Instantaneous frequency.* Proc. IRE, **41** (1953), pp. 548-548.

[40] J. Tropp and A. Gilbert, Signal recovery from random measurements via Orthogonal Matching Pursuit, *IEEE Trans. Inform. Theory*, **53:12**, pp. 4655-4666, 2007.

[41] B. Van der Pol, *The fundamental principles of frequency modulation,* Proc. IEE, **93** (1946), pp. 153-158.

[42] Z. Wu and N. E. Huang, *Ensemble Empirical Mode Decomposition: a noise-assisted data analysis method,* COLA Technical Report 193 (2005). ftp://grads.iges.org/pub/ctr/ctr_193.pdf.

[43] Z. Wu, N. E. Huang, S. R. Long, C. K. Peng, *Trend, detrend, and the variability of nonlinear and non-stationary time series.* Proc. Natl. Acad. Sci. USA., **104** (2007), 14889-14894. doi: 10.1073/pnas.0701020104.

[44] Z. Wu and N. E. Huang, *Ensemble Empirical Mode Decomposition: a noise-assisted data analysis method.* Advances in Adaptive Data Analysis. **1(1)** (2009), pp. 1-41.