Sparse Time-Frequency decomposition by adaptive basis pursuit

Thomas Y. Hou, Zuoqiang Shi

Abstract

In our recent paper, we proposed a data-driven time-frequency analysis method. We also presented convincing numerical evidence to demonstrate the effectiveness of our method. Convergence analysis has been also carried out for periodic signals that satisfy certain scale separation property. In this paper, we propose an improved time-frequency analysis method that can be used to decompose signals that do not have good scale separation property. Our method is formulated as a nonlinear optimization problem using nonlinear basis pursuit. Unlike the classical basis pursuit where the basis is known a priori, the basis in our nonlinear basis pursuit is not known a priori. Instead, it is adapted to the signal and is determined as part of the nonlinear optimization problem. This nonlinear optimization problem is solved by using the Augmented Lagrangian Multiplier method (ALM) iteratively. We further accelerate the convergence of the ALM method in each iteration by using the fast wavelet transform. We apply our method to decompose a number of multiscale data without scale separation, including signals with noise and signals with outliers. Our results show that our method can give accurate recovery of both the instantaneous frequencies and the intrinsic mode functions even for signals that have poor or no scale separation.

Index Terms

Sparse Time-Frequency decomposition, Adaptive basis pursuit, Instantaneous frequency

I. INTRODUCTION

Nowadays we must process a massive amount of data in our daily life and the scientific research. Data analysis methods have played an important role in processing and analyzing these data. Most data
analysis methods use pre-determined basis, including the most commonly used Fourier transform and wavelet transform. While these data analysis methods are very efficient in processing data, each component of the decomposition in general does not reveal the intrinsic physical information of these data due to the presence of harmonics in the decomposition. For example, application of these traditional data analysis methods to a modulated oscillatory chirp signal would produce many components. Thus, it is essential to develop a truly adaptive data analysis method that can extract hidden physical information such as trend and time varying cycles from these data and preserve the integrity of the physically meaningful components. To achieve this, we need to use a data-driven basis that is adapted to the signal instead of being determined \textit{a priori}.

Inspired by the EMD method \cite{14}, \cite{17} and the recently developed compressed (compressive) sensing theory, we have recently proposed a data-driven time-frequency analysis method \cite{11}, \cite{12}. This data-driven time-frequency analysis method is based on looking for the sparsest representation of a multiscale signal over a largest possible dictionary consisting of intrinsic mode functions. The adoption of this largest possible time-frequency basis and the search for the sparsest decomposition over a highly redundant basis make the time-frequency analysis method fully adaptive to the signal. In this method, the signal is decomposed by solving a nonlinear optimization problem. Further, we proposed an efficient algorithm based on matching pursuit \cite{12}, \cite{15} and Fast Fourier transform to solve this nonlinear optimization problem. In a subsequent paper \cite{13}, we proved the convergence of our algorithm for periodic data that satisfy certain scale separation property.

Although we have presented extensive numerical evidence to demonstrate the effectiveness of this method for data with good scale separation property \cite{12}, \cite{16}, this method does not work well for data with poor scale separation property \cite{12}, \cite{13}. In many real world applications, the physical data may not satisfy the scale separation condition. Thus, it is desirable to design an improved algorithm that can handle data without scale separation. In this paper, we attempt to alleviate this limitation by proposing an improved algorithm based on a new formulation.

In this paper, we consider those data that have the following representation:

\[
f(t) = \sum_{j=1}^{M} a_j(t) \cos \theta_j(t), \quad t \in \mathbb{R},
\]

where \(a_j(t), \theta_j(t)\) are smooth functions, \(\theta_j'(t) > 0, j = 1, \cdots, M\) and \(M\) is an integer that is given \textit{a priori}. We assume that \(a_j(t)\) and \(\theta_j'\) are less oscillatory than \(\cos \theta_j(t)\). The objective of our data-driven time-frequency analysis is to extract the Intrinsic Mode Functions (IMFs) defined as \(a_j(t) \cos \theta_j(t)\) and their Instantaneous Frequencies defined as \(\theta_j', j = 1, \cdots, M\). In many physical applications, the phase
functions $\theta_j$’s are nonlinear as a function of time. Thus the instantaneous frequencies are not constant in time. Traditional data analysis methods such as the Fourier Transform or Wavelet Transform would not be able to preserve the structure of the IMFs and tend to produce many harmonic modes. We do not assume that the instantaneous frequencies are well separated in the sense that $\theta_j(t)$ and $\theta_{j+1}(t)$ could be close to each other or even cross each other in some localized region. This poses a great challenge in decomposing the original signal into a sum of IMFs and recovering the $M$ instantaneous frequencies hidden in the original signal.

To decompose these multiscale data without scale separation, we formulate the above problem as the following nonlinear optimization problem:

$$\min_{x, \theta_1, \cdots, \theta_M} \|x\|_1, \quad \text{subject to } \Phi_{\theta_1, \cdots, \theta_M} x = f;$$

(2)

where

$$\Phi_{\theta_1, \cdots, \theta_M} = [\Phi_{\theta_1}, \cdots, \Phi_{\theta_M}],$$

(3)

and $\Phi_{\theta_j}$, $j = 1, \cdots, M$ is the basis to decompose the signal $f$. The specific form of the basis as a function of $\theta_j$ will be given in (7).

In the above problem, unlike the $l^1$ minimization problem studied in the compressive sensing, the basis $\Phi_{\theta_1, \cdots, \theta_M}$ is not known a priori. It is determined by the phase functions $\theta_j$, $j = 1, \cdots, M$ and the phase functions are adaptive to the data. In the above optimization problem, we need to solve not only the optimal coefficients $x$ but also the optimal basis $\Phi_{\theta_1, \cdots, \theta_M}$. In this sense, this problem can be seen as a nonlinear $l^1$ optimization problem.

This nonlinear optimization problem is non-convex and is challenging to solve. We propose to use the Augmented Lagrangian Multiplier method (ALM) to solve this nonlinear optimization problem iteratively. We further accelerate the convergence of the ALM method in each iteration by using the fast wavelet transform. This method can be also generalized to decompose signals that contain outliers by enlarging the dictionary to include impulses.

We will demonstrate the effectiveness of our method by applying it to decompose several multiscale data without scale separation. For those signals that do not have scale separation but with no noise pollution, we recover both the instantaneous frequencies and the intrinsic mode functions very accurately. Even for those signals that are polluted by noise, we still can approximate the instantaneous frequencies and the IMFs with reasonable accuracy comparable to the noise level. Application to signals with outliers gives very similar results.
We remark that a synchrosqueezed wavelet transform method has been developed by Daubechies, Lu and Wu [8] with a purpose similar to ours. This method also requires that the data have good scale separation property.

The remaining of the paper is organized as follows. In Section 2, we give the formulation of our problem. In Section 3, an iterative algorithm is introduced to solve the nonlinear optimization problem. An accelerated procedure is introduced based on the fast wavelet transform in Section 4. In Section 5, we generalize this method to deal with signals that have outliers. In Section 6, several numerical results are presented to demonstrate the effectiveness of our method. Finally, some concluding remarks are made in Section 7.

II. SPARSE TIME-FREQUENCY DECOMPOSITION BASED ON ADAPTIVE BASIS PURSUIT

In this section, we will set up the framework of the sparse time-frequency decomposition. Let \( \{V_l\}_{l \in \mathbb{Z}} \) be a multi-resolution approximation of \( L^2(\mathbb{R}) \) and \( \varphi \) the associated scaling function, \( \psi \) the corresponding wavelet function. Assume that \( \varphi \) is real and \( \hat{\varphi} \) has compact support, \( \text{supp}(\hat{\varphi}) = (-s_\varphi, s_\varphi) \), where \( \hat{\varphi} \) is the Fourier transform of \( \varphi \) defined below,

\[
\hat{\varphi}(k) = \frac{1}{2\pi} \int_{\mathbb{R}} \varphi(t) e^{-ikt} dt.
\]

For each \( 1 \leq j \leq M \), we assume that \( \theta'_j > 0 \). Since \( \theta_j \) is a strictly monotonously increasing function, we can use \( \theta_j \) as a coordinate. Then we can define the following wavelet basis in the \( \theta_j \)-coordinate,

\[
\Pi_{\theta_j} = \left\{ \psi_{l,n}(\theta_j)_{l,n \in \mathbb{Z}}, \varphi_{l_0,n}(\theta_j)_{n \in \mathbb{Z}} \right\},
\]

where \( l_0 \) is a positive integer associated with the lowest frequency of the envelope. In practice, \( l_0 \) is usually determined by the time span of the signal. And

\[
\psi_{l,n}(\theta_j) = \frac{1}{\sqrt{2^{l}s_\varphi}} \psi \left( \frac{\theta_j}{2^l s_\varphi} - n \right),
\]

\[
\varphi_{l,n}(\theta_j) = \frac{1}{\sqrt{2^{l}s_\varphi}} \varphi \left( \frac{\theta_j}{2^l s_\varphi} - n \right).
\]

In this paper, we use the Meyer wavelet to construct the basis.

In order to simplify the notations, we denote

We assume that the envelope is sufficiently smooth, \( a_j \in C^\infty \). Thus, \( a_j \) has a sparse representation over the wavelet basis \( \Pi_{\theta_j} \), and the corresponding IMF, defined as \( a_j \cos \theta_j \), has a sparse representation.
over the following basis:

\[ \Phi_{\theta_j} = (\cos \theta_j)^{\Pi_{\theta_j}} = \left[ (\psi_{l,n}(\theta_j) \cos \theta_j)_{i,n \in \mathbb{Z}, 0 < l \leq l_0} (\varphi_{l,n}(\theta_j) \cos \theta_j)_{n \in \mathbb{Z}} \right] \]  

(7)

This in turn implies that the signal \( f(t) \) given in (1) would have a sparse representation over the combination of all \( \Phi_{\theta_j}, j = 1, \cdots, M \):

\[ \Phi_{\theta_1, \cdots, \theta_M} = [\Phi_{\theta_1}, \cdots, \Phi_{\theta_M}] \]  

(8)

Based on this observation, we propose to decompose the signal \( f(t) \) in (1) by looking for the sparsest representation over \( \Phi_{\theta_1, \cdots, \theta_M} \):

\[ \min_{x, \theta_1, \cdots, \theta_M} \|x\|_0, \quad \text{subject to } \Phi_{\theta_1, \cdots, \theta_M} x = f, \]  

(9)

where \( x = [x_1, \cdots, x_M]^T \) and \( x_j, j = 1, \cdots, M \) are the coefficients of \( a_j \cos \theta_j \) over \( \Phi_{\theta_j} \) respectively.

Thanks to the development of the compressive sensing, we know that the minimization using the \( l^0 \) norm can be relaxed to the convex minimization using the \( l^1 \) norm in many cases. We also apply this relaxation to simplify the optimization problem. This gives rise to the following nonlinear \( l^1 \) optimization problem:

\[ \min_{x, \theta_1, \cdots, \theta_M} \|x\|_1, \quad \text{subject to } \Phi_{\theta_1, \cdots, \theta_M} x = f. \]  

(10)

Our nonlinear optimization problem is different from the compressive sensing problem in that not only the coefficients are unknown, but the basis over which the signal has a sparse representation is also unknown \( a \text{ priori} \). The basis is determined by the phase functions \( \theta_1, \cdots, \theta_M \). Both the coefficients \( x \) and the phase functions need to be determined in the optimization process. In this sense, our decomposition is completely data driven.

**Remark II.1.** In this paper, we construct the basis by utilizing the wavelet basis instead of the overcomplete Fourier basis which was used in our previous paper [12]. We choose the wavelet basis over the overcomplete Fourier basis because there are fast decomposition and reconstruction algorithms using the wavelet basis. This feature makes our algorithm very efficient. For periodic signals, we can use the overcomplete Fourier basis, which can be made very efficient by using the Fast Fourier transform. But the Fourier basis works only for periodic signals. For general nonperiodic signals, the wavelet basis seems to be more robust although it still suffers from the "end effect" near the boundary of the signal.
III. ALGORITHM BASED ON THE AUGMENTED LAGRANGIAN MULTIPLIER METHOD

Inspired by the algorithm in [12], we propose the following iterative algorithm to solve the nonlinear $l^1$ optimization problem (10).

**Algorithm 1** (Gauss-Newton type iteration)

**Input:** Initial guess of phase functions $\theta^0_j, j = 1, \cdots, M, \eta = l_0$, where $l_0$ is same as that in (4).

**Output:** Phase function and corresponding envelope $\theta_j, a_j, b_j, j = 1, \cdots, M$.

1: while $\eta \geq 1$ do
2:   while $\sum_{j=1}^M \| \theta_j^{n+1} - \theta_j^n \|_2 > \epsilon_0$ do
3:     Solve the following $l^1$ optimization problem:

$$
\left( \tilde{a}^{n+1}, \tilde{b}^{n+1} \right) = \arg \min_{x, y} (\| x \|_1 + \| y \|_1),
$$

subject to

$$
\Phi_{\theta_1^n, \cdots, \theta_M^n} \cdot x + \Psi_{\theta_1^n, \cdots, \theta_M^n} \cdot y = f,
$$

where $\Psi_{\theta_1^n, \cdots, \theta_M^n} = [\Psi_{\theta_1^n}, \cdots, \Psi_{\theta_M^n}]$ and

$$
\Psi_{\theta_j^n} = (\sin \theta_j) \Pi_{\theta_j}, \quad j = 1, \cdots, M. \tag{12}
$$

4: Calculate the envelopes $a_j, b_j, j = 1, \cdots, M$:

$$
a_j = \Pi_{\theta_j^n} \cdot \tilde{a}_j^{n+1}, \quad b_j = \Pi_{\theta_j^n} \cdot \tilde{b}_j^{n+1}, \tag{13}
$$

5: Update $\theta_j^n, j = 1, \cdots, M$:

$$
\Delta \theta_j' = PV_{n}(\theta_j^n) \left( \frac{d}{dt} \left( \arctan \left( \frac{b_j}{a_j} \right) \right) \right),
$$

$$
\Delta \theta_j = \int_{0}^{t} \Delta \theta_j'(s) ds, \quad \theta_j^{n+1} = \theta_j^n - \beta_j \Delta \theta_j,
$$

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

$$
\beta_j = \max \left\{ \alpha \in [0, 1] : \frac{d}{dt} (\theta_j^n - \alpha \Delta \theta_j) \geq 0 \right\}.
$$

Here $PV_{n}(\theta_j^n)$ is the projection operator to the space $V_{\eta}(\theta_j^n)$ and

$$
V_{\eta}(\theta) = \text{span} \left[ \psi_{l,n} (\theta)_{l \leq l_0, n \in \mathbb{Z}}, \varphi_{l,n} (\theta)_{n \in \mathbb{Z}} \right].
$$

6: end while
7: $\eta = \eta - 1.$
8: end while
In the above algorithm, the most expensive part is to solve the $l^1$ optimization problem (11). In this paper, we use an Augmented Lagrange Multiplier (ALM) algorithm [1] to solve (11). In order to simplify the notations, we denote

$$\Theta_{\theta_1, \cdots, \theta_M} = [\Theta_{\theta_1}, \cdots, \Theta_{\theta_M}],$$

where

$$\Theta_{\theta_j} = [\Phi_{\theta_j}, \Psi_{\theta_j}], \quad j = 1, \cdots, M.$$

The ALM method operates on the augmented Lagrangian

$$L(p, q) = \|p\|_1 + \langle q, f - \Theta_{\theta_1, \cdots, \theta_M} p \rangle + \frac{\mu}{2} \|f - \Theta_{\theta_1, \cdots, \theta_M} p\|^2.$$

A generic Lagrange multiplier algorithm [1] would solve (11) by repeatedly setting $p^{k+1} = \arg\min_p L(p, q^k)$, and then updating the Lagrange multiplier via $q^{k+1} = q^k + \mu(f - \Theta_{\theta_1, \cdots, \theta_M} p^k)$.

In this iteration, to solve $\min_p L(p, q^k)$ is also very time consuming. Note that the matrix $\Theta_{\theta_1, \cdots, \theta_M} = [\Theta_{\theta_1}, \cdots, \Theta_{\theta_M}]$ is the combination of $M$ matrices with smaller size. It is natural to use the following sweeping algorithm to solve $\min_p L(p, q^k)$ iteratively:

**Algorithm 2 (Sweeping)**

**Input:** $p_j^0 = 0$, $j = 1, \cdots, M$, $\mu > 0$.

1. while not converge do
2. for $j=1:M$ do
3. Compute $r_j^m = f - \sum_{l=1}^{j-1} \Theta_{\theta_l} p_l^{m+1} - \sum_{l=j+1}^{M} \Theta_{\theta_l} p_l^m$.
4. Compute $p_j^{m+1} = \arg\min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r_j^m + q^k / \mu - \Theta_{\theta_j} p_j\|^2.$
5. end for
6. end while

Theoretically, we need to run the above sweeping process several times until the solution converges, but in practical computations, in order to save the computational cost, we only run the sweeping process once. Combining this idea with the augmented Lagrange multiplier method, we obtain the following algorithm to solve the $l^1$ optimization problem (11):
**Algorithm 3 (Sweeping ALM)**

**Input:** $p_j^0 = 0$, $j = 1, \cdots, M$, $q_j^0 = 0$, $\mu > 0$

1: while not converge do
2:  for $j=1:M$ do
3:  Compute $r_j^k = f - \sum_{l=1}^{j-1} \Theta_{\theta_l} p_l^{k+1} - \sum_{l=j+1}^{M} \Theta_{\theta_l} p_l^k$.
4:  Compute $p_j^{k+1} = \arg \min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r_j^k + q_j^k / \mu - \Theta_{\theta_j} p_j\|_2^2$.
5:  end for
6:  $q^{k+1} = q^k + \mu \left( f - \sum_{j=1}^{M} \Theta_{\theta_j} p_j^{k+1} \right)$.
7: end while

---

**IV. A FAST ALGORITHM BASED ON THE DISCRETE WAVELET TRANSFORM**

In this section, we propose an approximate solver to accelerate the computation of $p_j^{k+1} = \arg \min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r_j^k + q_j^k / \mu - \Theta_{\theta_j} p_j\|_2^2$ which is the most expensive step in Algorithm 1.

In this paper, we only consider the signal which is well resolved by the samples and the samples are uniformly distributed in time and the total number of the samples is $N$. Based on these assumptions, we can switch between the continuous integral and the discrete summation and the integration error is omitted. This would give us lots of convenience in the calculation.

Now, we turn to simplify the optimization problem. First, we replace the standard $L^2$ norm by a weighted $L^2$ norm which gives the following approximation:

$$p_j^{k+1} = \arg \min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r_j^k + q_j^k / \mu - \Theta_{\theta_j} p_j\|_{2,\theta_j}^2,$$

where $\|g\|_{2,\theta_j}^2 = \sum_i g_i^2 \theta_j^i(t_i)$.

Using the fact that $\text{supp}(\hat{\varphi}) = (-s_\varphi, s_\varphi)$, it is easy to check that the columns of the matrix $\Theta_{\theta}$ are orthonormal under the weighted discrete inner product

$$\langle g, h \rangle_{\theta} = \sum_{i=1}^{N} g_i h_i \theta'(t_i) = g \cdot \theta'(h),$$

where $g = [g_i], h = [h_i], \theta'(h) = [\theta'(t_i) h_i], i = 1, \cdots, N$.

Based on this property, it is easy to derive the following equality:

$$\|\Theta_{\theta} \cdot x\|_{2,\theta} = \|x\|_2.$$

We can also define the projection operator $P_{V(\theta)}$ to $V(\theta)$ space. Here $V(\theta)$ is the linear space spanned by the columns of the matrix $\Theta_{\theta}$ and $P_{V(\theta)}$ is the projection operator to $V(\theta)$. Since the columns of the
matrix $\Theta_\theta$ are orthonormal under the weighted inner product (17), projection $P_{V(\theta)}$ can be calculated as follows:

$$P_{V(\theta)}(r) = \Theta_\theta \cdot \tilde{r}, \quad \tilde{r} = \Theta_\theta^T \cdot [\theta' r].$$

(19)

Now, we are ready to show that the optimization problem (16) can be solved explicitly by the shrinkage operator. To simplify the notation, we denote $w = r^k + q^k / \mu$,

$$\arg\min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r^k + q^k / \mu - \Theta_\theta p_j\|_{2,\theta_j}^2 = \arg\min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|\Theta_\theta p_j - P_{V(\theta_j)}(w)\|_{2,\theta_j}^2$$

$$= \arg\min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|\Theta_\theta p_j - \Theta_\theta^T \cdot [\theta' w]\|_{2,\theta_j}^2$$

$$= \arg\min_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|p_j - \Theta_\theta^T \cdot [\theta' w]\|_2^2$$

$$= S_{\mu^{-1}}(\Theta_\theta^T \cdot [\theta' (r^k + q^k / \mu)]) ,$$

(20)

where $S_\tau$ is the shrinkage operator defined below:

$$S_\tau(x) = \text{sgn}(x) \max(|x| - \tau, 0).$$

(21)

Notice that the matrix vector product in (20) has the following structure by the definition of $\Theta_\theta_j$ in (14)

$$\Theta_{\theta_j}^T \cdot (\theta'_j r) = \left[ \Pi_{\theta_j}^T \cdot (\cos \theta_j r \, \theta'_j), \Pi_{\theta_j}^T \cdot (\sin \theta_j r \, \theta'_j) \right]^T.$$

This is nothing but the wavelet transform of $\sin \theta_j r$ and $\cos \theta_j r$ in the $\theta_j$-coordinate, since the columns of $\Pi_{\theta_j}$ are standard wavelet basis in the $\theta_j$-coordinate. Then this product can be computed efficiently by interpolating $r \cos \theta_j$ and $r \sin \theta_j$ to the uniform grid in the $\theta_j$ coordinate and employing the fast wavelet transform.

Summarizing the above discussion, we get the following fast algorithm based on fast wavelet transform to solve the optimization problem (11):
Algorithm 4 (Sweeping ALM accelerated by the fast wavelet transform)

Input: $a_{\theta_j}^0 = b_{\theta_j}^0 = 0$, $j = 1, \cdots, M$, $q^0 = 0$.

1: while not converge do
2: for $j=1:M$ do
3: Compute
   \[ r_j^n = f - \sum_{l=1}^{j-1} (a_{\theta_l}^{n+1} \cos \theta_l + b_{\theta_l}^{n+1} \sin \theta_l) - \sum_{l=j+1}^{M} (a_{\theta_l}^{n} \cos \theta_l + b_{\theta_l}^{n} \sin \theta_l). \]
4: Interpolate $R = r_j^n + q^n/\mu$ from $\{t_i\}_{i=1}^N$ in the physical space to a uniform mesh in the $\theta_j$-coordinate to get $R_{\theta_j}$ and compute the wavelet representations:
   \[ R_{\theta_j,k} = \text{Interpolate} (\theta_j(t_i), R, \theta_j, k), \]
   where $\theta_{j,k}$, $j = 0, \cdots, N - 1$ are uniformly distributed in the $\theta_j$-coordinate, i.e. $\theta_{j,k} = 2\pi L_{\theta_j} k/N$. And compute
   \[ \tilde{a} = \sum_{k=1}^{N} \Pi_{\theta_j}^T(\theta_{j,k}) \cdot (\cos \theta_{j,k}) R_{\theta_j,k}, \]
   \[ \tilde{b} = \sum_{k=1}^{N} \Pi_{\theta_j}^T(\theta_{j,k}) \cdot (\sin \theta_{j,k}) R_{\theta_j,k}. \]
   This computation can be accelerated by the fast wavelet transform.
5: Apply the shrinkage operator compute $a_{\theta_j}$ and $b_{\theta_j}$ in $\theta_j$-coordinate:
   \[ a_{\theta_j}^{n+1}(\theta_{j,k}) = \Pi_{\theta_j}(\theta_{j,k}) \cdot S_{\mu^{-1}}(\tilde{a}), \]
   \[ b_{\theta_j}^{n+1}(\theta_{j,k}) = \Pi_{\theta_j}(\theta_{j,k}) \cdot S_{\mu^{-1}}(\tilde{b}). \]
   This step can also be accelerated by the wavelet reconstruction algorithm.
6: Interpolate $a_{\theta_j}$ and $b_{\theta_j}$ back to the physical grid points $\{t_i\}_{i=1}^N$:
   \[ a_{\theta_j}^{n+1}(t_i) = \text{Interpolate} (\theta_{j,k}, a_{\theta_j}(\theta_{j,k}), \theta_j(t_i)), \]
   \[ b_{\theta_j}^{n+1}(t_i) = \text{Interpolate} (\theta_{j,k}, b_{\theta_j}(\theta_{j,k}), \theta_j(t_i)). \]
7: end for
8: Compute
   \[ q^{n+1} = q^n + \mu \left( f - \sum_{j=1}^{M} (a_{\theta_j}^{n+1} \cos \theta_j + b_{\theta_j}^{n+1} \sin \theta_j) \right). \]
Remark IV.1. The final algorithm we described above is based on employing the soft shrinkage operator iteratively. At the first glance, this may look like the split Bregman iteration developed by Goldstein and Osher [10]. Actually, our method is quite different from the split Bregman iteration. The split Bregman iteration is a general method that can be used to solve the $l^1$ minimization problem with any matrix. In the derivation of our method, we utilize the special structure of the matrix $\Theta_{\theta_j}$, i.e. the orthonormality of the columns in the $\theta_j$ coordinate. For this specific problem, our method is more efficient than the split Bregman iteration.

V. GENERALIZATION FOR SIGNALS WITH OUTLIERS

One advantage of the formulation (10) is that we generalize it to deal with more complicated data with some minor modifications. In this section, we will give one generalization for signals with outliers.

In order to deal with this kind of signals, we have to enlarge the dictionary since the outliers are not sparse over the time-frequency dictionary. Fortunately, we know that the outliers are sparse over the basis consisting of the impulses $\delta[n-i], i = 1, \cdots, N$, where $N$ is the number of samples and
\[
\delta[n] = \begin{cases} 
1, & n = 0, \\
0, & n \neq 0. 
\end{cases}
\] (22)

If we enlarge the dictionary to include all the impulses, then the generalized formulation can be used to decompose signals with outliers.

More specifically, in this case, the optimization problem is formulated in the following way,
\[
\min_{x,\theta_1, \cdots, \theta_M} \|x\|_1 + \|z\|_1, 
\] (23)
subject to: $\Phi_{\theta_1, \cdots, \theta_M} \cdot x + z = f,$

where $\Phi_{\theta_1, \cdots, \theta_M}$ is given in (8).

Using the idea similar to that of Algorithm 1, we obtain the following Gauss-Newton type method for the above optimization problem (23):
Algorithm 5 (Gauss-Newton type iteration with outliers)

Input: Initial guess of phase functions $\theta_j^0$, $j = 1, \cdots, M$, $\eta = l_0$, where $l_0$ is same as that in (4).

Output: Phase function and corresponding envelope $\theta_j$, $a_j$, $j = 1, \cdots, M$.

1: while $\eta \geq 1$ do
2: \hspace{1em} while $\sum_{j=1}^{M} \|\theta_j^{n+1} - \theta_j^n\|_2 > \epsilon_0$ do
3: \hspace{2em} Solve the following $l^1$ optimization problem:
4: \hspace{3em} $\begin{aligned} (\tilde{a}^{n+1}, \tilde{b}^{n+1}, z^{n+1}) &= \text{argmin}_{x, y, z} (\|x\|_1 + \|y\|_1 + \|z\|_1), \\
\text{subject to: } & \Phi_{\theta_1^n, \cdots, \theta_M^n} \cdot x + \Psi_{\theta_1^n, \cdots, \theta_M^n} \cdot y + z = f. \end{aligned}$
5: \hspace{1em} Calculate the envelopes in the same way as we did in Algorithm 1.
6: \hspace{1em} Update $\theta_j^n$, $j = 1, \cdots, M$ in the same way as we did in Algorithm 1.
7: \hspace{1em} end while
8: $\eta = \eta - 1$.
9: end while

Moreover, the $l^1$ optimization problem in the above iterative algorithm can be solved by the following sweeping ALM method:

Algorithm 6 (Sweeping ALM with outliers)

Input: $p_j^0 = 0$, $j = 1, \cdots, M$, $q_j^0 = 0$, $\mu > 0$.

Output: Phase function and corresponding envelope $\theta_j$, $a_j$, $j = 1, \cdots, M$.

1: while not converge do
2: \hspace{1em} for $j = 1 : M$ do
3: \hspace{2em} $r_j^k = f - \sum_{l=1}^{j-1} \Theta_{\theta_l} p_l^{k+1} - \sum_{l=j+1}^{M} \Theta_{\theta_l} p_l^k - z^k.$
4: \hspace{2em} $p_j^{k+1} = \text{arg min}_{p_j} \|p_j\|_1 + \frac{\mu}{2} \|r_j^k + q_j^k / \mu - \Theta_{\theta_j} p_j\|^2.$
5: \hspace{1em} end for
6: $r_0^k = f - \sum_{l=1}^{M} \Theta_{\theta_l} p_l^k.$
7: $z^{k+1} = S_{\mu^{-1}}(r_0^k + q_j^k / \mu).$
8: $q_j^{k+1} = q_j^k + \mu \left( f - \sum_{j=1}^{M} \Theta_{\theta_j} p_j^{k+1} - z^{k+1} \right).$
9: end while
The computation of $P_j^{k+1} = \arg\min_{P_j} \|P_j\|_1 + \frac{\mu}{2} \|r_j^k + q_j^k - \Theta_{\theta_j} P_j\|_2^2$ can be accelerated by Algorithm 4 in the previous section.

VI. NUMERICAL RESULTS

In this section, we present several numerical results to demonstrate the effectiveness of our time-frequency analysis methods.

Example 1: In the first example, the signal contains two different components whose instantaneous frequencies intersect with each other. More specifically, the signal is generated by the formula below:

$$f = \cos \theta_1(t) + \cos \theta_2(t), \quad t \in [0, 1],$$

where the phase function $\theta_1, \theta_2$ are given as following:

$$\theta_1(t) = 39.2 \pi t - 12 \sin 2 \pi t,$$

$$\theta_2(t) = 85.4 \pi t + 12 \sin 2 \pi t.$$

The signal is sampled over 1024 grid points which are uniformly distributed over the interval $[0, 1]$. The original signal is shown in Fig. 1.

For this signal, the classical time-frequency analysis methods, such as the windowed Fourier transform, the wavelet transform give a poor result near the intersection of the two instantaneous frequencies. The EMD method and the data-driven time-frequency analysis method introduced in our previous paper [12] also have problem near the intersection.

The result given by the data-driven time-frequency analysis method proposed in this paper is shown in Fig. 1. In the computation, the initial guesses of the instantaneous frequencies are chosen to be $128 \pi t$ and $32 \pi t$ respectively, which are far from the ground truth. As we can see, even with these rough initial guesses, our iterative algorithm still can accurately recover the instantaneous frequencies and the corresponding IMFs.

Example 2: In this example, we add white noise to the signal in Example 1 to test the robustness of our method with respect to noise perturbation. The signal is given by

$$f = \cos \theta_1(t) + \cos \theta_2(t) + 0.5X(t), \quad t \in [0, 1],$$

where $X(t)$ is white noise with zero mean and variance $\sigma^2 = 1$. $\theta_1, \theta_2$ are given in (25) and (26). The signal is also sampled over 1024 uniform grid points and the samples are shown in Fig. 2.
Fig. 1. Left: Original signal in Example 1; Middle: instantaneous frequencies, red: exact, blue: numerical; Right: corresponding IMFs, red: exact, blue: numerical.

Fig. 2. Left: Original signal in Example 2, red curve is the clean signal without noise; Middle: instantaneous frequencies, red: exact, blue: numerical; Right: corresponding IMFs, red: exact, blue: numerical.

In this case, we still use the same initial guesses as those in the previous example. As shown in Fig. 2, our method can still recover the instantaneous frequencies and the corresponding IMFs with reasonable accuracy, although the signal is polluted by noise. We note that the end-effect is more pronounced in this case due to the noise pollution.

**Example 3:** In the third example, we consider a signal consisting of three components whose instantaneous frequencies all intersect. The signal is generated by the following formula:

\[ f = \cos \theta_1(t) + \cos \theta_2(t) + \cos 60\pi t, \quad t \in [0, 1]. \]  

(28)

\( \theta_1, \theta_2 \) are given in (25) and (26).
We also sample the signal over 1024 uniform grid points. The initial guesses are chosen to be $128\pi t$, $64\pi t$ and $32\pi t$ respectively. As shown in Fig. 3, our method gives very accurate approximations to the exact instantaneous frequencies and the corresponding IMFs. The recovered instantaneous frequencies and the IMFs are almost indistinguishable from the exact ones.

**Example 4:** The last signal that we consider is polluted by outliers. We generate the signal by using the following formula:

$$f = \cos \theta_1(t) + \cos \theta_2(t) + \sigma(t).$$

The signal is sampled over 1024 uniform grid points. Among these samples, there are 32 samples that are outliers. The locations of these outliers are selected randomly and the strengths satisfy the normal distribution. In Fig. 4, we present the results that we obtain using the algorithm given in Section 5. As we can see, both the IMFs and the outliers are captured very accurately.

We also test the signal with outliers and noise. The results is shown in Fig. 5. In this example, the noise and the outliers are added to the original signal together. The signal is sampled over 1024 uniform grid points. Among these samples, there are 32 samples that are outliers. The locations of these outliers are selected randomly and the strengths satisfy the normal distribution whose standard deviation is 1. The noise is Gaussian noise and the standard deviation is 0.1.

In this case, the instantaneous frequencies and the IMFs are still quite accurate. But the outliers are not captured as well as in the case with no noise. We would like to emphasize that it would be hard to distinguish the outliers from the noise when the amplitude of the outliers are small. For the outliers
whose amplitude is large, they can be separated from the noise by a proper shrinkage operator. However, the shrinkage operator also kills the outliers whose amplitude is comparable to or smaller than the noise level.

**VII. CONCLUDING REMARKS**

In our previous paper [12], a data-driven time-frequency analysis method has been developed by looking for the sparsest decomposition of the signal over a largest possible dictionary. A numerical algorithm based on nonlinear matching pursuit was proposed to solve the nonlinear $\ell^0$ optimization problem to obtain the decomposition. In a subsequent paper [13], we proved that this method is convergent and
stable for periodic data that have certain scale separation property. However, the method proposed in [12] does not work very well for data with poor scale separation.

In this paper, we introduced an improved numerical method to decompose data with poor scale separation. We again formulated this problem as a nonlinear optimization problem by looking for the sparsest decomposition of the signal over a largest possible dictionary. The main difference between the method presented in this method and our original method presented in [12] is that we decompose IMFs simultaneously instead of decomposing them one at a time as in our original method.

The algorithm that we proposed in this paper to obtain the sparsest decomposition is based on the Augmented Lagrangian Multiplier method (ALM). By designing the dictionary carefully to make them orthogonal in the coordinate of the phase function, we can accelerate the algorithm by using the fast wavelet transform. This makes our algorithm very efficient.

Another advantage of this method is that it can be easily generalized to deal with more complicated data that are not sparse over the time-frequency dictionary, such as data with outliers. For this kind of signals, we just need to enlarge the dictionary and follow the similar procedure to look for the sparsest decomposition over this enlarged dictionary. We presented several numerical examples to demonstrate the effectiveness of our method, including data that do not have scale separation and data that are polluted by noise or outliers. The results that we obtained seem to suggest that our method can offer an effective way to decompose multiscale data with poor scale separation property.

There are some remaining issues to be studied in the future. Our method requires several initial guesses of the phase functions. How to get proper initial guesses is not so easy especially for some real world data. Another important problem is to decompose data with intra-wave frequency modulation. This type of data is known to be very challenging. Naive application of traditional data analysis methods tends to introduce artificial harmonics. These issues will be addressed in our future work.

Acknowledgments. This research was supported in part by a DOE Grant DE-FG02-06ER25727, an AFOSR MURI Grant FA9550-09-1-0613 and an NSF FRG Grant DMS-1159138. The research of Dr. Z. Shi was supported by a NSFC Grant 11201257.

REFERENCES


