Mechanical Systems and Signal Processing 116 (2019) 566–584

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Adaptive chirp mode pursuit: Algorithm and applications

Shiqian Chen [a], Yang Yang [b], Zhike Peng [a,*], Xingjian Dong [a], Wenming Zhang [a], Guang Meng [a]

[a] State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Science and Technology on vehicle transmission laboratory, China North Vehicle Institute, Beijing 100072, China

ARTICLE INFO

Article history:
Received 20 March 2018
Received in revised form 31 May 2018
Accepted 25 June 2018

Keywords:
Multi-component signal
Chirp
Empirical mode decomposition
Variational mode decomposition
Synchrosqueezing
Time-frequency analysis

ABSTRACT

Signal decomposition has drawn growing interest in various applications these days. Some recent decomposition methods, like the variational mode decomposition (VMD) and the variational nonlinear chirp mode decomposition (VNCMD), employ a joint-optimization scheme to accurately estimate all the signal modes underlying a signal. Some existing issues for these methods are: requiring prior knowledge of the number of the signal modes, empirically setting the bandwidth parameter to a fixed value, and lacking of an effective initialization scheme for the optimization algorithm. To address these issues, this paper presents a new decomposition approach called adaptive chirp mode pursuit (ACMP). Similar to the matching pursuit method, the ACMP captures signal modes one by one in a recursive framework. In addition, an adaptive bandwidth parameter updating rule and an instantaneous frequency initialization method based on Hilbert transform are incorporated into the ACMP. Several examples including simulated signals as well as real-life applications are provided to show the effectiveness and advantages of the ACMP.

1. Introduction

Signal or data processing has drawn increasingly attention in various fields. Signals generated by real-life physical systems usually consist of multiple superimposed oscillations which are also called signal modes [1]. These signal modes contain valuable information of the system. Therefore, extracting the constituent modes underlying the signal becomes very important to study a physical system. In some applications such as fault diagnosis of varying-speed rotating machineries [2–5], the systems will generate non-stationary signals, i.e., signals whose frequency contents change with time, also known as chirp signals [6,7]. Decomposing a chirp signal or finding its modes (referred to as chirp modes) is quite challenging for existing methods [8].

During the past decades, researchers have proposed many adaptive signal decomposition methods among which the empirical mode decomposition (EMD) method is the most popular one [9,10]. The EMD employs a recursive sifting algorithm to find each signal mode. The problem is that the sifting algorithm is empirical and therefore it is difficult to model the algorithm with mathematical theory. In addition, the sifting procedure involves detecting and interpolating local extreme points of the signal, which is sensitive to noise and suffers from end effects. Despite these limitations, the EMD is still widely used in various applications such as speech processing [11] and condition monitoring of mechanical systems [12]. Although some

[*] Corresponding author.
E-mail addresses: chenshiqian@sjtu.edu.cn (S. Chen), yangyang913@163.com (Y. Yang), z.peng@sjtu.edu.cn (Z. Peng), donxij@sjtu.edu.cn (X. Dong), wenmingz@sjtu.edu.cn (W. Zhang), gmeng@sjtu.edu.cn (G. Meng).

https://doi.org/10.1016/j.ymssp.2018.06.052
0888-3270/© 2018 Elsevier Ltd. All rights reserved.

improved versions of EMD such as ensemble EMD (EEMD) [13] and the recent time-varying filter based EMD (TVF-EMD) [14] were developed, these methods still cannot achieve desired performance.

To overcome the limitations of the EMD, many advanced alternative methods have been used. For example, some methods retrieve each signal mode from proper time-frequency (TF) distributions because TF analysis methods can better represent the time-varying frequency contents of non-stationary signals [1,15]. Decomposition methods of this type usually use a post-processing procedure to find TF coefficients (obtained by certain TF transforms) associated with each mode and then reconstruct the modes using inverse TF transforms. The synchrosqueezing transform (SST) [16] is one of the most popular methods belonging to this type because it can significantly improve the readability and the resolution of the TF distribution (TFD) by using a TF reassignment technique. Researchers also developed some improved versions of the SST such as the high-order SST [17] or the matching SST [18] for signals with fast varying frequencies, and the synchroextracting transform [19] allowing for greatly enhancing the energy concentration of the TFD. Some different methods use advanced filtering techniques to extract signal modes. For example, the variational mode decomposition (VMD) method employs the Wiener filter bank [20,21], the empirical wavelet transform (EWT) [22,23] makes use of the wavelet filter bank, and the method in [24] utilizes the adaptive local iterative filtering technique. Other authors use a different way for signal decomposition by Hilbert transform [25,26]. The basic idea of the methods of this class is that the Hilbert transform can obtain the information of signal frequencies and amplitudes which can be further used for signal reconstruction. Other more widely used schemes decompose signals by optimization. For example, the operator based methods [21,27,28] and the sparse TF methods [29–31] explicitly model each signal mode as a local narrow-band signal or an AM-FM signal, and then find the modes by optimizing a function which incorporates the regularity assumptions about the defined model. The convergence of the resulting optimization algorithms usually depends on the initialization. Therefore, how to obtain a good initialization becomes quite important for this class of methods. Other recent work reports that the subspace decomposition algorithm based on SVD can also be used to extract desired signal modes [32,33].

It is worth noting that some decomposition methods (e.g., the VMD and EWT above) work in the frequency domain and thus cannot deal with chirp modes whose frequency ranges overlap. To address this issue, McNeill proposed the short-time narrow-banded mode decomposition (STNBMD) method [34] which extracts signal modes by optimizing an object function imposing constraints on the mode reconstruction error and the smooth degrees of the mode amplitudes and frequencies. The STNBMD is a successful and systematic extension of the VMD to signals with time-varying frequencies. Then, more recently, Chen introduced the variational nonlinear chirp mode decomposition (VNCMD) method [35] which exploits a complete variational framework to generalize the VMD. The VNCMD can decompose chirp signals with very close or even crossed modes. However, several issues still remain to be addressed for both VMD and VNCMD. Firstly, the two methods (as well as the STNBMD) find all the signal modes concurrently via a joint-optimization technique and therefore the number of the signal modes should be specified in advance. In fact, it is difficult to know the exactly number of the modes for real-life data. Secondly, the bandwidth parameter for both methods is fixed during the optimization. Nevertheless, to facilitate the convergence of the algorithm, it is desired to adjust the bandwidth parameter according to different stages of the optimization. Last but not least, how to obtain good initializations for the frequencies of the modes is unsolved for these methods.

In this paper, a signal decomposition method called adaptive chirp mode pursuit (ACMP) is proposed to address the issues listed above. Inspired by the matching pursuit method [30,36], the ACMP adopts a recursive decomposition scheme to extract the signal modes one by one. The algorithm no longer requires the input of the number of the modes. We also show that the bandwidth parameter can be adaptively updated with the iterations of the algorithm. In addition, we show the initial frequencies for the iterative algorithm of ACMP can be obtained by using Hilbert transform.

The rest of the paper is organized as follows. In Section 2, we introduce the model of the chirp mode and review the VNCMD method. The proposed ACMP method is introduced in Section 3 where a simulated example is also considered to demonstrate our method. In Section 4, various examples including simulated and real-life applications are presented to test the performance of the ACMP. The conclusion is summarized in Section 5.

2. Theoretical background

2.1. Signal model

In reality, non-stationary signals often contain several sub-signals which are termed as chirp modes in this paper. Each chirp mode can be modeled as an AM-FM signal. More precisely, the signal model is expressed as

$$x(t) = \sum_{m=1}^{M} x_m(t) = \sum_{m=1}^{M} a_m(t)\cos(2\pi \int_0^t f_m(s)\mathrm{d}s + \varphi_m) \tag{1}$$

where the signal $x(t)$ is a superposition of M chirp modes $x_m(t)$ for $m = 1,\ldots,M$; $a_m(t) > 0$ is the non-negative envelope (or amplitude) of the m-th mode, $f_m(t) > 0$ is the instantaneous frequency (IF), and φ_m denotes the initial phase. Generally, the envelope and the IF are slowly varying functions compared to the phase function, i.e., $|a'_m(t)|, |f'_m(t)| \ll |f_m(t)|$. In this paper, we assume that these signal modes are well separated in the TF domain as the SST does. Namely, the IFs of the signal modes should satisfy the separation condition [16]: $f_m(t) > f_{m-1}(t)$ and $|f_m(t) - f_{m-1}(t)| \geqslant \gamma |f_m(t) + f_{m-1}(t)|$ with the separation degree $\gamma > 0$. The task of a signal decomposition method is to find each mode $x_m(t)$ and even its envelope $a_m(t)$ and IF $f_m(t)$ (for $m = 1, \ldots, M$) for a given signal $x(t)$.

In some applications, Eq. (1) can be rewritten as the de-chirped form as

$$x(t) = \sum_{m=1}^{M} \alpha_m(t)\cos(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) + \beta_m(t)\sin(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) \tag{2}$$

with

$$\alpha_m(t) = a_m(t)\cos(2\pi\int_0^t (f_m(s) - \tilde{f}_m(s))\mathrm{d}s + \varphi_m) \tag{3}$$

$$\beta_m(t) = -a_m(t)\sin(2\pi\int_0^t (f_m(s) - \tilde{f}_m(s))\mathrm{d}s + \varphi_m) \tag{4}$$

where $\alpha_m(t)$ and $\beta_m(t)$ are two de-chirped signals, $\tilde{f}_m(t)$ is the frequency function of the de-chirping factors $\cos(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s)$ and $\sin(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s)$. Note that $\alpha_m(t)$ and $\beta_m(t)$ will become slowly varying baseband signals when $\tilde{f}_m(t) = f_m(t)$.

2.2. Variational nonlinear chirp mode decomposition

The VNCMD in [35] is a generalization of the VMD [20] to analyzing non-stationary signals with spectrum-overlapped modes. The basic idea of the VNCMD is that the de-chirped signals $\alpha_m(t)$ and $\beta_m(t)$ (see (3) and (4)) will have the narrowest frequency band when $\tilde{f}_m(s) = f_m(s)$. Therefore, the VNCMD estimates the signal modes and their IFs by solving the following constrained optimization problem [35]

$$\begin{aligned} &\min_{\{\alpha_m\},\{\beta_m\},\{\tilde{f}_m\}} \left\{ \sum_{m=1}^{M} (\|\alpha''_m(t)\|_2^2 + \|\beta''_m(t)\|_2^2) \right\} \\ &\mathrm{s.}t.x(t) = \sum_{m=1}^{M} \alpha_m(t)\cos(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) + \beta_m(t)\sin(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) \end{aligned} \tag{5}$$

where the square of the l_2 norm (i.e., $\|\cdot\|_2$) of the second derivative is employed to measure the bandwidth of the signal. Then, the augmented Lagrangian multiplier is employed to solve the optimization problem (5) as

$$L_\tau(\{\alpha_m\},\{\beta_m\},\{\tilde{f}_m\},\lambda) = \sum_{m=1}^{M} (\|\alpha''_m(t)\|_2^2 + \|\beta''_m(t)\|_2^2) + \langle \lambda(t), e(t) \rangle + \tau\|e(t)\|_2^2 \tag{6}$$

with

$$e(t) = x(t) - \sum_{m=1}^{M} \{\alpha_m(t)\cos(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) + \beta_m(t)\sin(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s)\} \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $e(t)$ is the error term, $\lambda(t)$ is the Lagrangian multiplier, $\tau > 0$ is a weighting factor controlling the relative contribution of the bandwidth term and the error term to the function (6). The solution to the problem (5) is found as the saddle point of (6) by alternately solving a series of sub-problems. The details of the optimization algorithm are available in [35]. In fact, the VNCMD behaves as a TF filter bank. The parameter τ controls the bandwidth of the filter. A smaller τ leads to a narrower bandwidth for the filter (i.e., the output signals will be smoother).

It shows that the VNCMD uses the joint-optimization scheme (6) to find all the M signal modes simultaneously. The problem is that the number of the modes (i.e., M) is not usually available and such a joint optimization may be unstable in some cases as indicated in [35]. In addition, the VNCMD employs a fixed bandwidth (i.e., a fixed τ), which cannot best adapt to different stages of the optimization algorithm. How to obtain good initial guesses of the IFs is also unsolved for the VNCMD. All these issues prohibit practical applications of the VNCMD.

3. Adaptive chirp mode pursuit

To address the issues of the VNCMD, the ACMP algorithm is presented in this section. The main ingredient of our algorithm is a recursive mode extraction framework together with an adaptive bandwidth updating rule and an IF initialization scheme based on Hilbert transform. A simulated example is also provided to demonstrate the effectiveness of the ACMP in this section.

3.1. Algorithmic framework

As mentioned above, simultaneously estimating all the signal modes is difficult and may be subjected to stability issues. Motivated by the matching pursuit-like methods [27,28,30], the ACMP employs a recursive framework to extract signal modes one by one. Specifically, to extract the m-th signal mode (see (1) and (2)), the ACMP solves the following problem

$$\min_{\alpha_m,\beta_m,\tilde{f}_m} \{\|\alpha_m''(t)\|_2^2 + \|\beta_m''(t)\|_2^2 + \tau\|x(t) - x_m(t)\|_2^2\} \tag{8}$$

with

$$x_m(t) = \alpha_m(t)\cos(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) + \beta_m(t)\sin(2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s) \tag{9}$$

where the first two terms impose smoothness constraints on the two de-chirped signals $\alpha_m(t)$ and $\beta_m(t)$ (similar to that in (5)), the third term represents the energy of the residual signal, $\tau > 0$ is the weighting factor. In essence, formula (8) indicates a greedy algorithm similar to the matching pursuit which finds the desired mode by minimizing the energy of the residue.

We assume that the signal is observed in discrete time $t = t_0, \cdots, t_{N-1}$ where N is the number of samples. Then, substituting (9) into (8) leads to the following objective function in the matrix form as

$$J_\tau(\boldsymbol{y}_m,\boldsymbol{f}_m) = \|\boldsymbol{\Phi}\boldsymbol{y}_m\|_2^2 + \tau\|\boldsymbol{x} - \mathbf{K}_m\boldsymbol{y}_m\|_2^2 \tag{10}$$

where $\boldsymbol{x} = [x(t_0)\cdots x(t_{N-1})]^T$, $\boldsymbol{f}_m = [\tilde{f}_m(t_0)\cdots\tilde{f}_m(t_{N-1})]^T$; $\boldsymbol{y}_m = [\boldsymbol{\alpha}_m^T\ \boldsymbol{\beta}_m^T]^T$ with $\boldsymbol{\alpha}_m = [\alpha_m(t_0)\cdots\alpha_m(t_{N-1})]^T$, $\boldsymbol{\beta}_m = [\beta_m(t_0)\cdots\beta_m(t_{N-1})]^T$;

$$\mathbf{K}_m = [\mathbf{C}_m\ \mathbf{S}_m] \tag{11}$$

with

$$\mathbf{C}_m = diag[\cos(\theta_m(t_0))\cdots\cos(\theta_m(t_{N-1}))] \tag{12}$$

$$\mathbf{S}_m = diag[\sin(\theta_m(t_0))\cdots\sin(\theta_m(t_{N-1}))] \tag{13}$$

where $\theta_m(t) = 2\pi\int_0^t \tilde{f}_m(s)\mathrm{d}s$; $\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}$ where $\mathbf{0}$ is a zero matrix, $\mathbf{D}$ is a second-order difference matrix of size $(N-2)\times N$ as

$$\mathbf{D} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -2 & 1 \end{bmatrix}.$$

To minimize the objective function (10), an iterative algorithm alternately updating the de-chirped signal (i.e., $\boldsymbol{y}_m$) and the IF (i.e., $\boldsymbol{f}_m$) is developed. Firstly, the vector $\boldsymbol{y}_m$ is updated by setting the gradient to zero, i.e., $\partial J_\tau(\boldsymbol{y}_m,\boldsymbol{f}_m)/\partial\boldsymbol{y}_m = \mathbf{0}$ as

$$\boldsymbol{y}_m^n = \begin{bmatrix} \boldsymbol{\alpha}_m^n \\ \boldsymbol{\beta}_m^n \end{bmatrix} = \left(\frac{1}{\tau}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + (\mathbf{K}_m^n)^T\mathbf{K}_m^n\right)^{-1}(\mathbf{K}_m^n)^T\boldsymbol{x} \tag{14}$$

where the superscript $\cdot^n$ stands for the iteration counter. Then, the target signal mode can be recovered as

$$\boldsymbol{x}_m^n = \mathbf{K}_m^n\boldsymbol{y}_m^n. \tag{15}$$

Since the objective function $J_\tau(\boldsymbol{y}_m,\boldsymbol{f}_m)$ is nonlinearly dependent on the IF $\boldsymbol{f}_m$, it is difficult to directly update the IF using the gradient of $J_\tau(\boldsymbol{y}_m,\boldsymbol{f}_m)$. Fortunately, the phase function of the two de-chirped signals in (3) and (4) is related to the IF error which can be used to update the IF. Therefore, using the result in (14), the IF increment can be obtained by arctangent demodulation [37] as

$$\Delta\tilde{f}_m^n(t) = -\frac{1}{2\pi}\frac{\mathrm{d}}{\mathrm{d}t}\left(\arctan\left(\frac{\beta_m^n(t)}{\alpha_m^n(t)}\right)\right) = \frac{\beta_m^n(t)\cdot(\alpha_m^n(t))' - \alpha_m^n(t)\cdot(\beta_m^n(t))'}{2\pi((\alpha_m^n(t))^2 + (\beta_m^n(t))^2)}. \tag{16}$$

In general, the IF is a smooth (or low-pass) function [34]. To alleviate the influence of noise, the IF increment is also assumed to satisfy the low-pass property [38] as

$$\min_{\Delta\boldsymbol{f}_m^n}\left\{\|\mathbf{D}\Delta\boldsymbol{f}_m^n\|_2^2 + \mu\|\Delta\boldsymbol{f}_m^n - \Delta\tilde{\mathbf{f}}_m^n\|_2^2\right\} \tag{17}$$

where $\Delta\tilde{\mathbf{f}}_m^n = [\Delta\tilde{f}_m^n(t_0),\ldots,\Delta\tilde{f}_m^n(t_{N-1})]^T$, $\mu > 0$ is a weighting factor, $\Delta\boldsymbol{f}_m^n$ is the estimated IF increment as

$$\Delta \boldsymbol{f}_m^n = \left(\mathbf{I} + \frac{1}{\mu}\mathbf{D}^T\mathbf{D}\right)^{-1} \Delta \tilde{\mathbf{f}}_m^n \tag{18}$$

where $\mathbf{I}$ is an identity matrix. In fact, formula (17) acts as a low-pass filter whose cutoff frequency is directly proportional to the parameter μ. Then, the IF vector can be updated as

$$\boldsymbol{f}_m^{n+1} = \boldsymbol{f}_m^n + \Delta \boldsymbol{f}_m^n. \tag{19}$$

The obtained IF $\boldsymbol{f}_m^{n+1}$ can in turn be used to update the kernel matrix $\mathbf{K}_m^{n+1}$ according to Eqs. (11)–(13). The above process should be repeated until there is little difference in the obtained signal modes (see (15)) in two adjacent iterations. Then, the finally obtained signal mode can be subtracted (or removed) from the original signal $x(t)$ and the same iterative procedure can be applied to the residual signal to estimate other signal modes. It is worth noting that the computational cost of the ACMP is mainly attributed to the matrix manipulations (e.g., (14)). Fortunately, the ACMP uses many sparse matrixes (e.g., the difference matrix and the diagonal matrix) which can greatly save the storage space and reduce the computational complexity.

The algorithmic framework of ACMP is summarized in Algorithm 1. The algorithm includes two loops where the inner loop is designed to estimate a certain signal mode while the outer loop is to recursively extract each signal mode. The recursive mode extraction scheme is more stable and thus is widely adopted by decomposition methods like the EMD and matching pursuit. In the algorithm, we do not need to input the number of the signal modes but it can be learnt by assessing the energy of the residual signal. The stopping threshold ε (for inner loop) is usually set to 1e−9–1e−7 (depending on the desired accuracy) while δ (for outer loop) is often set to 0.05–0.15 (considering different noise levels). As for the parameter μ, the method can obtain smoother IFs with a smaller μ but it may be unable to capture details of some complicated IFs if μ is too small. Generally, it is suggested to set μ to 1e−9–1e−8. In next subsection, the setting of the bandwidth parameter τ will be discussed.

Algorithm 1: Recursive mode extraction framework for ACMP

```
Input signal $\boldsymbol{x}$; parameters $\mu > 0$, $\tau > 0$; stopping thresholds $\delta$, $\varepsilon$
Set $m = 1$, $\boldsymbol{r}_1 = \boldsymbol{x}$
while $\|\boldsymbol{r}_m\|_2^2/\|\boldsymbol{x}\|_2^2 > \delta$ do
  Set $n = 0$, obtain the initial IF $f_m^1(t)$, and create matrix $\mathbf{K}_m^1$ with $f_m^1(t)$
  while $\|\boldsymbol{x}_m^n - \boldsymbol{x}_m^{n-1}\|_2^2/\|\boldsymbol{x}_m^{n-1}\|_2^2 > \varepsilon$ do
    $n = n + 1$
    $\boldsymbol{y}_m^n = \begin{bmatrix} \boldsymbol{\alpha}_m^n \\ \boldsymbol{\beta}_m^n \end{bmatrix} = \left(\frac{1}{\tau}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + (\mathbf{K}_m^n)^T\mathbf{K}_m^n\right)^{-1}(\mathbf{K}_m^n)^T\boldsymbol{r}_m$
    $\boldsymbol{x}_m^n = \mathbf{K}_m^n\boldsymbol{y}_m^n$
    $\Delta\tilde{f}_m^n(t) = \frac{\beta_m^n(t)\cdot(\alpha_m^n(t))' - \alpha_m^n(t)\cdot(\beta_m^n(t))'}{2\pi((\alpha_m^n(t))^2 + (\beta_m^n(t))^2)}$
     $\boldsymbol{f}_m^{n+1} = \boldsymbol{f}_m^n + \left(\mathbf{I} + \frac{1}{\mu}\mathbf{D}^T\mathbf{D}\right)^{-1}\Delta\tilde{\mathbf{f}}_m^n$
     Update kernel matrix $\mathbf{K}_m^{n+1}$ using $f_m^{n+1}(t)$ based on (11)–(13)
  end while
  Set $\boldsymbol{x}_m = \boldsymbol{x}_m^n, \boldsymbol{f}_m = \boldsymbol{f}_m^{n+1}$, and update the residue $\boldsymbol{r}_{m+1} = \boldsymbol{x} - \boldsymbol{x}_m$
  $m = m + 1$
end while
Output signal modes $\{\boldsymbol{x}_m\}_{m=1,2\cdots}$ and IFs $\{\boldsymbol{f}_m\}_{m=1,2\cdots}$
```

3.2. Bandwidth adaption rule

In essence, the ACMP method can also be regarded as an adaptive TF filter. As shown in (14), the center frequency of the filter is the current estimated IF (i.e., $f_m^n(t)$ included in matrix $\mathbf{K}_m^n$), and the bandwidth is controlled by parameter τ. Since the IF is updating in the iterative algorithm, the pass band of the filter is always changing until it finds the correct signal mode. If using a smaller τ, the output signal mode will be smoother, i.e., the bandwidth of the filter is smaller. Many filter-like decomposition methods such as the VMD, VNCMD, and sparse TF approaches [20,29,31,35] use a fixed bandwidth parameter in the whole optimization process. According to our experience, in the initial stage of the optimization algorithm, the estimated IF is often far from the true IF, i.e., there exist large estimation errors. In this stage, it is desired to choose a large bandwidth to include the valuable information of the target signal mode as much as possible. For example, as shown in Fig. 1(a), a large bandwidth BW_1 is required so that the pass band of the filter can cover the signal mode (i.e., the purple solid line). Conversely, in the final stage of the algorithm, the estimated IF is close to the true one and therefore a small bandwidth is enough for recovering the signal mode, as shown in Fig. 1(b). In this stage, a small bandwidth is beneficial to exclude much unwanted noise, i.e., increase signal-to-noise ratio (SNR) of the mode.

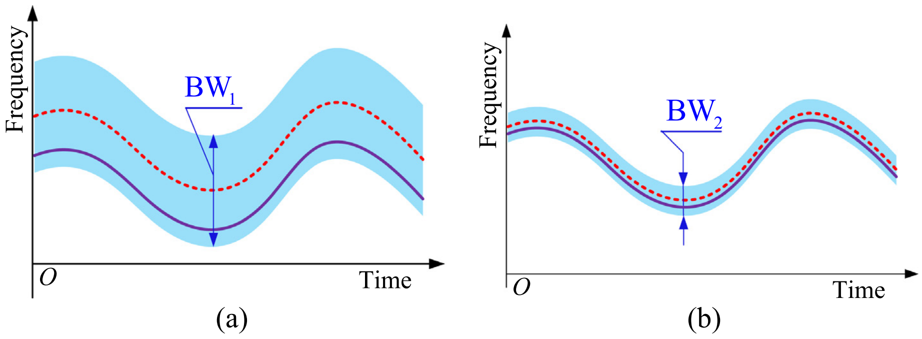


Fig. 1. Illustration of the bandwidth adaption rule at different optimization stages (the red dash line denotes the current estimated IF, i.e., the center frequency of the TF filter; the purple solid line denotes the true IF, i.e., the location of the signal mode; the wathet blue area represents the pass band of the filter; BW$_1$, BW$_2$ are the bandwidth values and BW$_1$ > BW$_2$). (a) Initial stage. (b) Final stage. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As discussed above, an intelligent scheme is to gradually decrease the bandwidth of the filter from a large value to a relatively small value with the increasing of iterations (i.e., gradually decrease the parameter τ). Note that, in [34], the STNBMD firstly suggests to set the bandwidth parameter to several different values at different optimization stages. In this paper, a more systematic algorithm is introduced to adaptively update the bandwidth parameter. To achieve this goal, the orthogonal property of the matching pursuit method is employed [28,36]. Firstly, we consider a signal model as

$$\boldsymbol{x} = \hat{\boldsymbol{x}}_m + \hat{\boldsymbol{r}}_m \tag{20}$$

where $\boldsymbol{x}$ is the input signal in the vector form, $\hat{\boldsymbol{x}}_m$ denotes the optimal target mode, $\hat{\boldsymbol{r}}_m$ is the optimal residual signal. According to the principle of the matching pursuit, $\hat{\boldsymbol{x}}_m$ is obtained as the orthogonal projection of $\boldsymbol{x}$ on a certain basis of the signal space spanned by all the chirp modes (i.e., model (1)). Therefore, $\hat{\boldsymbol{x}}_m$ is orthogonal to $\hat{\boldsymbol{r}}_m$. Namely,

$$\hat{\boldsymbol{x}}_m^T \hat{\boldsymbol{r}}_m = 0. \tag{21}$$

Eq. (21) indicates that there is no interference between the signal mode and the residue. Combining (20) with (21) leads to

$$\frac{\hat{\boldsymbol{x}}_m^T \hat{\boldsymbol{x}}_m}{\hat{\boldsymbol{x}}_m^T \boldsymbol{x}} = 1. \tag{22}$$

Considering (14) and (15), we have

$$\begin{aligned}\hat{\boldsymbol{x}}_m &= \hat{\mathbf{K}}_m \hat{\boldsymbol{y}}_m = \hat{\mathbf{K}}_m \left(\frac{1}{\hat{\tau}} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + (\hat{\mathbf{K}}_m)^T \hat{\mathbf{K}}_m \right)^{-1} (\hat{\mathbf{K}}_m)^T \boldsymbol{x} \\ &= \frac{1}{\hat{\tau}} \hat{\mathbf{K}}_m \left(\frac{1}{\hat{\tau}^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{1}{\hat{\tau}} (\hat{\mathbf{K}}_m)^T \hat{\mathbf{K}}_m \right)^{-1} (\hat{\mathbf{K}}_m)^T \boldsymbol{x} \\ &= \frac{1}{\hat{\tau}} \mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m) \boldsymbol{x}\end{aligned} \tag{23}$$

where $\hat{\mathbf{K}}_m$ denotes the optimal kernel matrix created with the true IF $f_m(t)$ according to (11)–(13), $\hat{\boldsymbol{y}}_m$ is the optimal dechirped signal, $\hat{\tau}$ is the optimal bandwidth parameter, the matrix $\mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m)$ is related to $\hat{\tau}$ and $\hat{\mathbf{K}}_m$ and is expressed as

$$\mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m) = \hat{\mathbf{K}}_m \left(\frac{1}{\hat{\tau}^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{1}{\hat{\tau}} (\hat{\mathbf{K}}_m)^T \hat{\mathbf{K}}_m \right)^{-1} (\hat{\mathbf{K}}_m)^T. \tag{24}$$

By inserting (23) into (22), the following equation is obtained

$$\hat{\tau} = \frac{\boldsymbol{x}^T (\mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m))^T \mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m) \boldsymbol{x}}{\boldsymbol{x}^T (\mathbf{R}(\hat{\tau}; \hat{\mathbf{K}}_m))^T \boldsymbol{x}}. \tag{25}$$

The optimal bandwidth parameter $\hat{\tau}$ is found as the fixed point [28] of (25) using the following iterative scheme as

$$\tau^{n+1} = \frac{\boldsymbol{x}^T (\mathbf{R}(\tau^n; \mathbf{K}_m^n))^T \mathbf{R}(\tau^n; \mathbf{K}_m^n) \boldsymbol{x}}{\boldsymbol{x}^T (\mathbf{R}(\tau^n; \mathbf{K}_m^n))^T \boldsymbol{x}}. \tag{26}$$

where the superscript $\cdot^n$ or $\cdot^{n+1}$ also stands for the iteration counter. In fact, using the relation in (22), equation (26) can be simplified as (or equivalent to)

$$\tau^{n+1} = \frac{\tau^n (\boldsymbol{x}_m^n)^T \boldsymbol{x}_m^n}{(\boldsymbol{x}_m^n)^T \boldsymbol{x}}. \tag{27}$$

Note that we usually have $(\boldsymbol{x}_m^n)^T \boldsymbol{x}_m^n / (\boldsymbol{x}_m^n)^T \boldsymbol{x} < 1$ when $\boldsymbol{x}_m^n \neq \hat{\boldsymbol{x}}_m$ (the case $\boldsymbol{x}_m^n = \hat{\boldsymbol{x}}_m$ is given in Eq. (22)). Namely, Eq. (27) indicates that the bandwidth parameter τ will be gradually decreased in the iterative algorithm until a certain termination condition is satisfied. Herein $(\boldsymbol{x}_m^n)^T \boldsymbol{x}_m^n / (\boldsymbol{x}_m^n)^T \boldsymbol{x}$ can be regarded as an index of orthogonality (IO) which measures the independence of the estimated signal mode (at n-th iteration; i.e., $\boldsymbol{x}_m^n$) from the residual signal (i.e., $\boldsymbol{x} - \boldsymbol{x}_m^n$; see (21)). A larger IO (the optimal case: IO = 1) means that there is less interference between the signal mode and the residue. In general, a smaller bandwidth parameter τ leads to a larger IO since the ACMP can exclude out more irrelevant components (including noise). Therefore, with the decrease of the bandwidth parameter in the iterations (see (27)), the IO is gradually increased. The IO controls the rate of decrease of the bandwidth in turn (see (27)). Only when the IO is close to 1, the updating of the bandwidth parameter can be stopped.

The complete ACMP algorithm with the bandwidth adaption rule is outlined in Algorithm 2. One should input the initial bandwidth parameter τ^1 for the algorithm. As discussed above, the initial bandwidth should be relatively large since the initial obtained IF $f_m^1(t)$ is far from the true IF. It is suggested to set the initial parameter τ^1 to 1e−5–1e−2 according to our simulations. Different initial values of τ^1 may require different iterations to obtain a good final bandwidth value. In addition, the convergence of the ACMP often depends on the initialization of the IF (step 4 in Algorithm 2). With a too bad initial IF (i.e., far from the true IF), the ACMP can extract nothing but the noise component near the initial IF. In this case, the IO value is much smaller than 1 and the bandwidth parameter of the ACMP tends to shrink fast to a very small value (to discard the noise). In practical applications, one can easily identify this situation and then try to reinitialize the IF or choose a larger initial bandwidth parameter τ^1. An effective IF initialization method will be introduced in next subsection.

Algorithm 2: ACMP with bandwidth adaption scheme

```
1: **Input** signal $\boldsymbol{x}$; parameters $\mu > 0$, $\tau^1 > 0$ (initial); stopping thresholds $\delta$, $\varepsilon$
2: Set $m = 1$, $\boldsymbol{r}_1 = \boldsymbol{x}$
3: **while** $\|\boldsymbol{r}_m\|_2^2 / \|\boldsymbol{x}\|_2^2 > \delta$ **do**
4:     Set $n = 0$, obtain the initial IF $f_m^1(t)$, and create matrix $\mathbf{K}_m^1$ with $f_m^1(t)$
5:     **while** $\|\boldsymbol{x}_m^n - \boldsymbol{x}_m^{n-1}\|_2^2 / \|\boldsymbol{x}_m^{n-1}\|_2^2 > \varepsilon$ **do**
6:         $n = n + 1$
7:         $\boldsymbol{y}_m^n = \begin{bmatrix} \boldsymbol{\alpha}_m^n \\ \boldsymbol{\beta}_m^n \end{bmatrix} = \left( \frac{1}{\tau^n} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + (\mathbf{K}_m^n)^T \mathbf{K}_m^n \right)^{-1} (\mathbf{K}_m^n)^T \boldsymbol{r}_m$
8:         $\boldsymbol{x}_m^n = \mathbf{K}_m^n \boldsymbol{y}_m^n$
9:         $\Delta \tilde{f}_m^n(t) = \frac{\beta_m^n(t) \cdot (\alpha_m^n(t))' - \alpha_m^n(t) \cdot (\beta_m^n(t))'}{2\pi((\alpha_m^n(t))^2 + (\beta_m^n(t))^2)}$
10:        $\boldsymbol{f}_m^{n+1} = \boldsymbol{f}_m^n + \left( \mathbf{I} + \frac{1}{\mu} \mathbf{D}^T \mathbf{D} \right)^{-1} \Delta \tilde{\mathbf{f}}_m^n$
11:        Update kernel matrix $\mathbf{K}_m^{n+1}$ using $f_m^{n+1}(t)$ based on (11)–(13)
12:        $\tau^{n+1} = \frac{\tau^n (\boldsymbol{x}_m^n)^T \boldsymbol{x}_m^n}{(\boldsymbol{x}_m^n)^T \boldsymbol{r}_m}$
13:    **end while**
14:    Set $\boldsymbol{x}_m = \boldsymbol{x}_m^n$, $\boldsymbol{f}_m = \boldsymbol{f}_m^{n+1}$, and update the residue $\boldsymbol{r}_{m+1} = \boldsymbol{x} - \boldsymbol{x}_m$
15:    $m = m + 1$
16: **end while**
17: **Output:** signal modes $\{\boldsymbol{x}_m\}_{m=1,2\cdots}$ and IFs $\{\boldsymbol{f}_m\}_{m=1,2\cdots}$
```

3.3. IF initialization

The last issue is how to obtain the initial IF $f_m^1(t)$ for the algorithm (see Step 4 in Algorithm 2). An effective and reliable method is to estimate initial frequencies by detecting ridge curves of a TFD generated by a certain TF transform. However, calculating the TFD and further extracting ridges of the TFD can be tedious and time-consuming. Since Hilbert transform can provide frequency information of a signal [25], an alternative IF initialization method based on Hilbert transform is introduced for the ACMP.

Firstly, the signal model (1) is rewritten into a compact form as

$$x(t) = \sum_{m=1}^{M} a_m(t) \cos(\phi_m(t)) \tag{28}$$

where $\phi_m(t) = 2\pi \int_0^t f_m(s)ds + \varphi_m$ is the instantaneous phase function of the m-th signal mode. If the amplitude $a_m(t)$ and the phase $\phi_m(t)$ satisfy certain properties [39], the analytic signal of (28) can be created as

$$z(t) = x(t) + j\mathscr{H}\{x(t)\} \approx \sum_{m=1}^{M} a_m(t)\exp(j\phi_m(t)) = A(t)\exp(j\phi(t)) \tag{29}$$

where $\sqrt{-1} = j$, $\mathscr{H}\{\cdot\}$ denotes the Hilbert transform, $A(t)$ and $\phi(t)$ are the amplitude and phase of the whole analytic signal (i.e., the multimode signal), respectively. The analytic expression of the IF of (29) is obtained in [40] as

$$f(t) = \frac{\phi'(t)}{2\pi} = \sum_{m=1}^{M} \frac{w_m(t)}{A^2(t)} \frac{\phi'_m(t)}{2\pi} + G(t) \tag{30}$$

with

$$A^2(t) = \sum_{m=1}^{M} w_m(t) \tag{31}$$

$$w_m(t) = \sum_{n=1}^{M} a_m(t)a_n(t)\cos(\phi_m(t) - \phi_n(t)) \tag{32}$$

$$G(t) = \sum_{m=1}^{M}\sum_{n=1}^{M} \frac{a'_m(t)a_n(t)}{2\pi A^2(t)} \sin(\phi_m(t) - \phi_n(t)). \tag{33}$$

It can be seen that the IF (30) is composed of two terms: the first term is the weighted average of the IFs (i.e., $\phi'_m(t)/2\pi$) of the modes, the second term is a oscillating function related to the amplitudes. If $a_i(t) \gg a_m(t)$ for $i \neq m$, then $f(t) \approx \phi'_i(t)/2\pi$ [40]. Namely, the IF of the whole multimode signal is approximately identical to that of the dominant signal mode. In addition, the desired initial IF should be smooth enough and therefore the contribution of the oscillating parts (e.g., $G(t)$ in (30)) to the IF should be reduced. Consequently, similar to (18), we obtain the initial IF by low-pass filtering (i.e., smoothing) the IF (i.e., $f(t)$) of the multimode signal. In summary, for the IF initialization step of Algorithm 2 (see step 4), we first calculate the IF of the whole residual signal $\boldsymbol{r}_m$ (see step 13) by Hilbert transform, and then low-pass filter the obtained IF.

As discussed above, the initial IF is essentially obtained as the weighted average of the IFs of the residual modes. The obtained initial IF is often closer to that of the stronger mode. Therefore, the proposed method will first extract the strongest signal mode. Some similar IF calculation and initialization strategies based on Hilbert transform can be found in [14,25,41]. It is worth noting that, the signal may be stationary (i.e., the IF is a constant instead of a function) in many cases. For stationary signals, the initial frequency can be easily obtained by detecting the peak frequency of the Fourier spectrum, and step 10 of Algorithm 2 can be simplified as

$$f_m^{n+1} = f_m^n + \frac{1}{T}\int_0^T \Delta\tilde{f}_m^n(t)\mathrm{d}t \tag{34}$$

where f_m^n or f_m^{n+1} is a constant frequency value, T is the time duration of the signal. The second term on the right hand side of (34) stands for the average value of the function $\Delta\tilde{f}_m^n(t)$. In this way, the proposed method can be tailored to analyze stationary signals effectively.

3.4. Demonstration and comparison with VNCMD

To demonstrate the effectiveness and advantages of the proposed ACMP, a polynomial phase signal contaminated by noise is considered as

$$x_n(t) = \cos(2\pi(2.6 + 30t + 9t^2 - 2t^3 + 0.18t^4)) + n(t) \tag{35}$$

where $n(t) \sim \mathcal{N}(0, 1.26)$ denotes a white Gaussian noise with the mean of zero and the standard deviation of 1.26, the IF of the signal is $f(t) = 30 + 18t - 6t^2 + 0.72t^3$. To measure the noise level or the accuracy of the mode reconstruction, the SNR (unit: dB) is computed as

$$\mathrm{SNR} = 10\log_{10}\frac{\|x(t)\|_2^2}{\|\tilde{x}(t) - x(t)\|_2^2} \tag{36}$$

where $x(t)$ denotes the theoretical noise-free signal, $\tilde{x}(t)$ is the estimated or measured one. Therefore, the SNR of the noisy signal (35) is obtained as -5 dB. The waveform and the TFD (calculated by short-time Fourier transform (STFT)) of the noisy signal are illustrated in Fig. 2. It shows that the noise is so strong that clear TF pattern of the signal is not available in the TFD (see Fig. 2(b)).

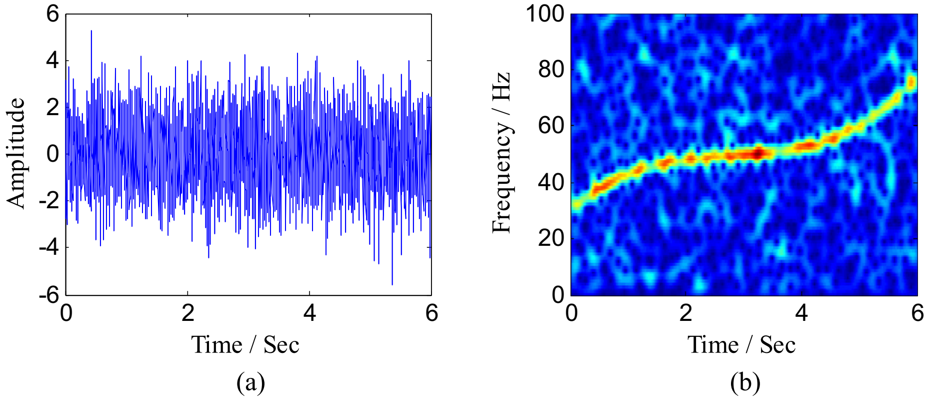


Fig. 2. Test signal in (35). (a) Waveform. (b) TFD by STFT.

The proposed ACMP algorithm is used to estimate the IF and reconstruct the signal. Herein the initial bandwidth parameter τ^1 is set to 1e−2; since the noise is very strong, the IF smoothing parameter μ is set to a relatively small value, i.e., 1e−10. For comparison, the VNCMD method is also used to analyze the signal. The VNCMD employs a fixed bandwidth parameter τ and herein we consider two cases, i.e., a smaller bandwidth parameter 1e−4 and a larger one 1e−2 (other parameter settings of the VNCMD are the same as that of the ACMP). The method introduced in Section 3.3 is adopted to obtain the initial IF for the VNCMD and ACMP. The analysis results for the signal are shown in Fig. 3. It can be seen that although the obtained initial IF deviates from the true one due to the so strong noise, it still follows the variation tendency of the true IF. For the VNCMD, if using a smaller τ (i.e., 1e−4), the algorithm cannot converge to correct results where the initial IF is far from the true IF (see 0 s– 2.5 s in Fig. 3(a) and (d); note that the algorithm can get good results during 2.5 s–6 s because the initial IF is closer to the true one in that time segment). On the other hand, if using a larger bandwidth parameter (i.e., 1e−2), the VNCMD will include the signal as well as more unwanted noise. Therefore, the estimated IF is unsmooth (or distorted) and the reconstructed signal is too noisy (see Fig. 3(b) and (e)). As discussed previously, the proposed ACMP employs an adaptive bandwidth rule: using a larger bandwidth in the initial stage to find the correct signal

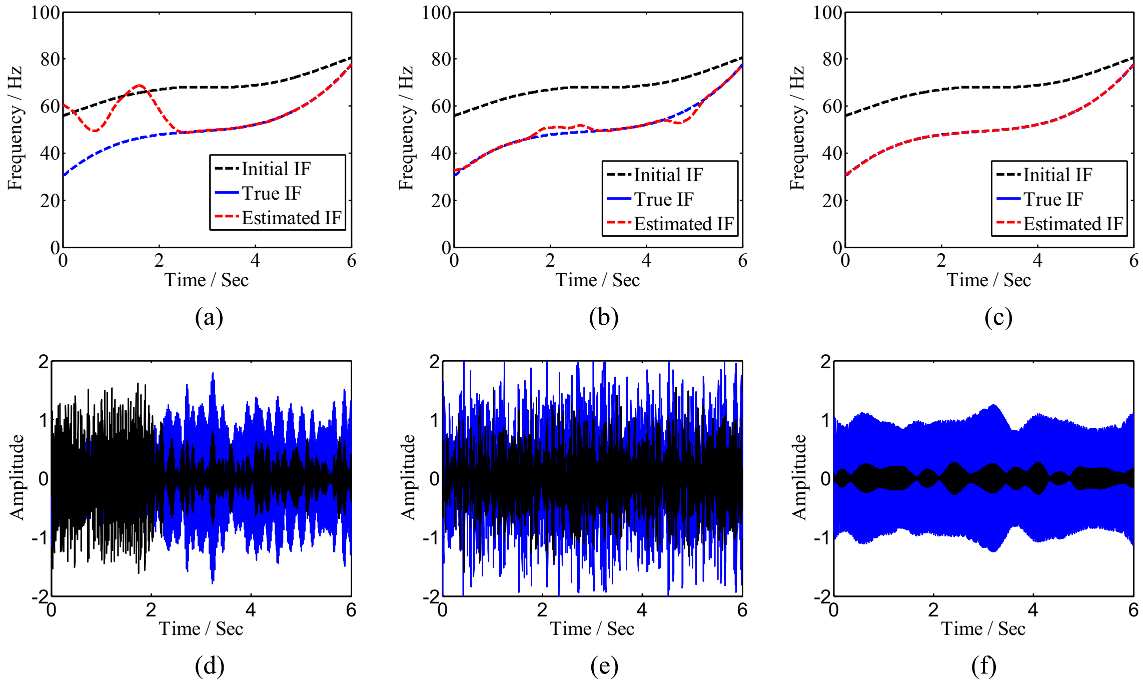


Fig. 3. Estimated IFs and signals for the example in (35) by ACMP and VNCMD. (a) and (d) show the estimated IF and the reconstructed signal by the VNCMD with $\tau = 1e{-}4$, respectively; (b) and (e) are the results of the VNCMD with $\tau = 1e{-}2$; (c) and (f) show the estimation results of the proposed ACMP. For (d)–(f), the blue and black waveforms denote the estimated signals and the estimation errors, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

mode and then gradually reducing the bandwidth to alleviate the influence of noise. As a result, the method obtains an accurate estimation for the IF (see Fig. 3(c)) and significantly removes the noise for the reconstructed signal (see Fig. 3(f)). The SNRs of the reconstructed signals in Fig. 3(d)–(f) are 3.29, 3.19, and 15.84 dB, respectively. It shows that the SNR of the signal can be surprisingly improved by the ACMP (compared to the original SNR −5 dB). This example indicates that the ACMP has better convergence and filtering properties than the VNCMD in a so noisy environment.

4. Simulated examples and real applications

In the section, more examples including both simulated signals as well as real-life applications will be provided to show the effectiveness of the proposed method. In this paper, all the numerical experiments are carried out with MATLAB (R2010b) on a personal computer with a 2.6-GHz CPU.

4.1. Tones separation

As a signal decomposition algorithm, the separation capability for close signal modes is a key factor for consideration. To test the separation performance of the ACMP, we consider a two-tone signal model introduced in [42] as

$$x(t;a,f_r)=\cos(2\pi t)+a\cos(2\pi f_r t) \tag{37}$$

where $0\leqslant f_r\leqslant 1$, $0.01\leqslant a\leqslant 100$ denotes the frequency ratio and the amplitude ratio of the two signal modes, respectively. f_r essentially controls the close degree of the two modes. Obviously, if f_r is larger, the two modes will be closer (i.e., it will be more difficult to separate them; the two modes will overlap when $f_r=1$). Herein we define a separation index (SI) to quantify the separation performance as

$$\mathrm{SI}(a,f_r)=\min\left\{\frac{\|\tilde{x}_1(t;a,f_r)-\cos(2\pi t)\|_2}{\|a\cos(2\pi f_r t)\|_2},\frac{\|\tilde{x}_1(t;a,f_r)-a\cos(2\pi f_r t)\|_2}{\|\cos(2\pi t)\|_2}\right\} \tag{38}$$

where $\tilde{x}_1(t;a,f_r)$ stands for the first extracted mode from the signal $x(t;a,f_r)$ by the algorithm. Since it is not sure which signal mode is extracted first by the ACMP algorithm, the SI in (38) computes two indexes related to both signal modes and chooses the smaller one. A zero value of (38) indicates that the two modes are perfectly separated while the SI value approaching to 1 means that the modes can hardly be separated.

To separate two close signal modes, the maximum bandwidth of the ACMP should not be too large. Therefore, we choose to use a relatively small initial bandwidth parameter τ^1 for the ACMP. In addition, to show the effect of τ^1 on the separation performance, three different values of τ^1, i.e., 1e−5, 1e−6, and 1e−8 are considered. For comparison, the separation performance of the EMD, the recently introduced TVF-EMD [14], and the EWT [22] is also evaluated. Fig. 4 illustrates the SI results by different methods. It shows that the SI results depend on both the amplitude and frequency ratios for the EMD, TVF-EMD and EWT. However, it seems that the amplitude ratio (i.e., a) has less effect on the results of the ACMP. The EMD and TVF-EMD can separate two modes only when f_r is smaller than a certain cut-off value. The value of EMD is around 0.67 while that of the TVF-EMD is increased to 0.8 (see Fig. 4(a) and (b)). The EWT shows better separation properties than EMD and TVF-EMD but the separation errors of the EWT are also significantly increased when $f_r\geqslant 0.8$ (see Fig. 4(c)). For the ACMP, the cut-off f_r is as high as 0.96 when letting $\tau^1=1\mathrm{e}{-5}$ (see Fig. 4(d)). If using a smaller τ^1, the separation performance of the ACMP can be further improved. For example, as shown in Fig. 4(f), the algorithm can nearly separate the two modes of any $f_r\neq 1$ when letting $\tau^1=1\mathrm{e}{-8}$.

4.2. Stationary example

As mentioned in Section 3.3, the proposed method can adapt to stationary signals with constant frequencies (see (34)). Herein we consider a stationary signal composed of four signal modes and the Gaussian noise as

$$\begin{aligned}&x_n(t)=x_1(t)+x_2(t)+x_3(t)+x_4(t)+n(t)\\&x_1(t)=6t+1,x_2(t)=\cos(8\pi t),\\&x_3(t)=0.5\cos(40\pi t),x_4(t)=0.5\cos(50\pi t)\end{aligned} \tag{39}$$

where the first mode is a linear trend, $n(t)\sim\mathcal{N}(0,0.2)$ is a white noise. The noisy signal and its Fourier spectrum are shown in Fig. 5. It can be seen that the waveform of the synthesized signal is very complicated (see Fig. 5(a)). The proposed ACMP is used to decompose the signal, and the extracted four signal modes are denoted by m1–m4, respectively. The decomposition results by EEMD (we use the EEMD instead of EMD in a noisy environment) and EWT are also provided, as shown in Fig. 6. It can be observed that the ACMP successfully extracts the four modes, and these estimated signal modes are almost coincident with the true modes (see Fig. 6(a)). The decomposition result of the EEMD is very similar to that of the EWT as shown in Fig. 6 (b) and (c). For the two methods, it can be found that: the estimated m1 contains the first two modes (i.e., $x_1(t), x_2(t)$) of (39) while the m2 includes the last two signal modes (i.e., $x_3(t), x_4(t)$); m3 and m4 are mainly composed of noise. Namely, m1 and m2 obtained by the two methods require for further decomposition. It indicates that in a multiple-mode and noisy environ-

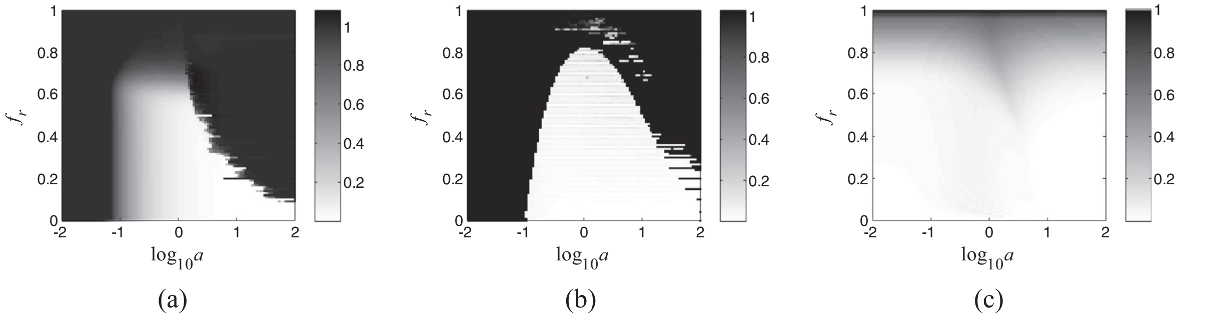


Fig. 4. SI(a,f_r) for the signal (37) by different methods. (a) EMD. (b) TVF-EMD. (c) EWT. (d)–(f) show the results of ACMP with τ^1 being 1e−5, 1e−6, and 1e−8, respectively.

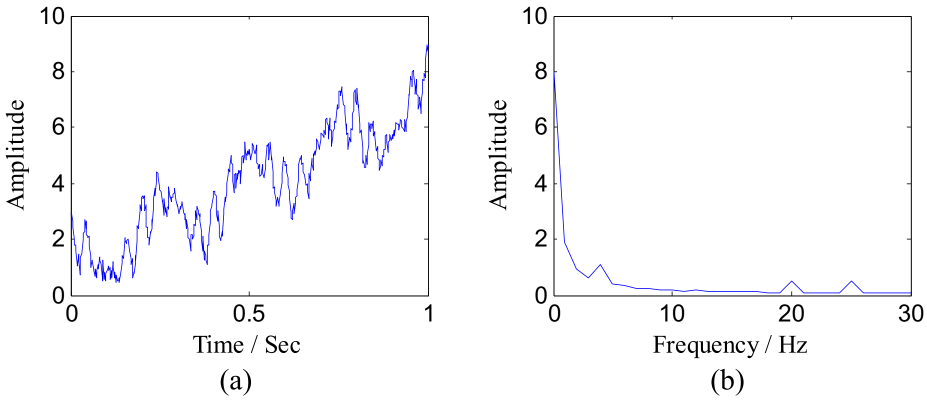


Fig. 5. Stationary signal in (39). (a) Waveform. (b) Fourier spectrum.

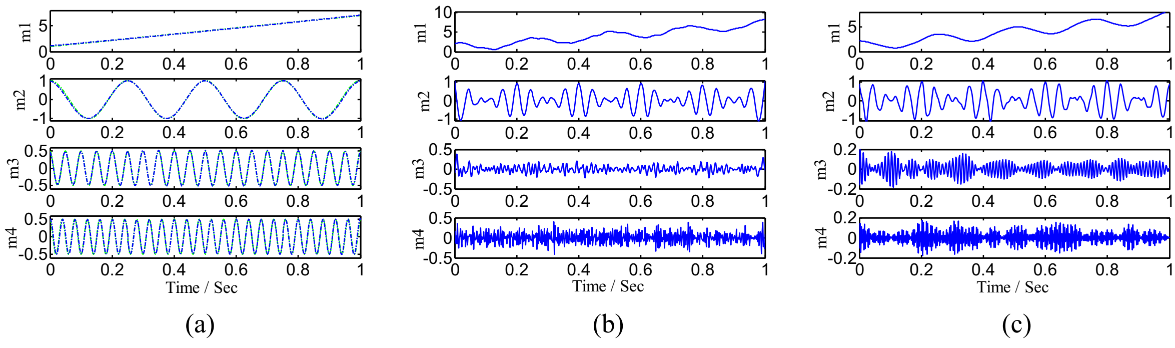


Fig. 6. Decomposition results for signal (39) by different methods. (a) ACMP (blue: extracted; green: true). (b) EEMD. (c) EWT. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ment, the EEMD and EWT cannot identify and separate close signal modes ($x_1(t), x_2(t)$ in (39) are close to each other, so are $x_3(t), x_4(t)$). Herein the run times of the ACMP, EEMD and EWT are 0.7 s, 0.07 s and 0.06 s, respectively. The cost times of the EEMD and EWT are smaller since they do not involve a complex optimization process.

4.3. Non-Stationary example

Herein we study the application of the ACMP to decompose non-stationary signals (or chirp signals) with multiple signal modes. We consider a noisy signal composed of three signal modes as

$$\begin{aligned} x_n(t) &= x_1(t) + x_2(t) + x_3(t) + n(t) \\ x_1(t) &= \exp(-0.03t) \times \cos(2\pi(1.3 + 25t + 4t^2 - 0.8t^3 + 0.07t^4)) \\ x_2(t) &= \exp(-0.06t) \times \cos(2\pi(2.6 + 40t + 8t^2 - 1.6t^3 + 0.14t^4)) \\ x_3(t) &= \exp(-0.09t) \times \cos(2\pi(3.9 + 60t + 12t^2 - 2.4t^3 + 0.21t^4)) \end{aligned} \tag{40}$$

where $n(t) \sim \mathcal{N}(0, 0.3)$ denotes the noise; each signal mode has an exponential amplitude and a polynomial phase; the IFs of the three modes are $f_1(t) = 25 + 8t - 2.4t^2 + 0.28t^3$, $f_2(t) = 40 + 16t - 4.8t^2 + 0.56t^3$, and $f_3(t) = 60 + 24t - 7.2t^2 + 0.84t^3$, respectively. The SNR with respect to each signal mode is 6.69, 5.98, and 5.31 dB, respectively. The noisy signal is illustrated in Fig. 7. It can be seen that the frequency ranges of the three signal modes are overlapping.

The ACMP is employed to analyze the non-stationary signal. The nonlinear IF curves of the modes are accurately estimated as shown in Fig. 8(a). The three signal modes are successfully extracted as shown in Fig. 8(b). The SNRs of the extracted modes are 20.60, 20.47, and 20.05 dB, respectively, which have been effectively improved by comparison to the original SNRs (i.e., 6.69, 5.98, and 5.31 dB). Note that the three signal modes are overlapping in the frequency domain and some existing methods like EWT cannot separate them. Therefore, we only consider the EEMD and TVF-EMD for comparison. Due to the limited space, the waveforms of the extracted modes by EEMD and TVF-EMD are not given. Instead, the TFDs of these modes are provided to clearly show their TF patterns, as shown in Fig. 9. It shows that the ACMP accurately separates the three signal modes and effectively removes the noise (see Fig. 9(a)–(c)). For the EEMD, serious mode mixing problem occurs (especially the second mode in Fig. 9(e)) for the non-stationary example, and the extracted signal modes by EEMD contain much noise. The TVF-EMD recovers the first signal mode well (see Fig. 9(g)) but it still cannot fully separate the last two modes (see Fig. 9(h) and (i)). The SNRs of the three reconstructed modes by TVF-EMD are 14.51, 6.76, and 5.08 dB, respectively, which are much lower than those of ACMP. In addition, different from the EEMD, TVF-EMD and EWT, the ACMP can not only separate the non-stationary signal modes but also estimate their IFs (e.g., see Fig. 8(a)), which shows important applications in various fields. For this example, the run times of the ACMP, EEMD and TVF-EMD are 2.7 s, 0.3 s and 27.1 s, respectively.

4.4. Application to length-of-day data

To show the effectiveness of the proposed method in practical applications, the ACMP is used to analyze the deviation of the length-of-day (LOD) data in 1096 consecutive days from the year 1980 to 1982 [43]. Analyzing the LOD data is very important in geophysics research fields. The LOD data considered here is shown in Fig. 10(a). To show the hidden physical information in the data, the ACMP is employed to decompose the signal and the first extracted 5 signal modes (denoted by m1–m5) are provided in Fig. 10(b). Note that there is a clear physical interpretation for each extracted signal in Fig. 10(b). For example, m1 reflects the trend of the LOD data; the period of m2 is around a year (i.e., 365 days) and therefore m2 relates to

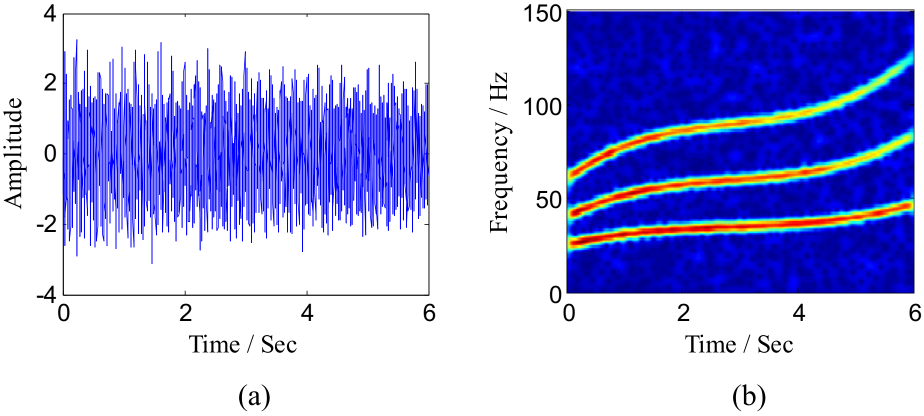


Fig. 7. Non-stationary signal in (40). (a) Waveform. (b) TFD by STFT.

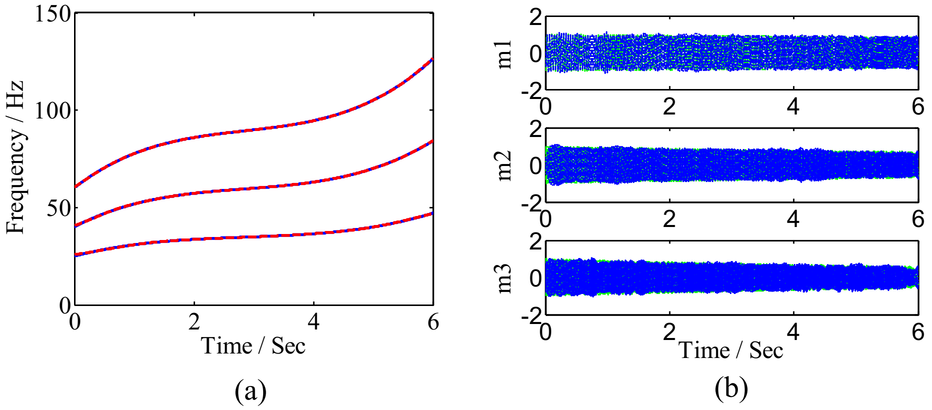


Fig. 8. Analysis results for signal (40) by ACMP. (a) Estimated IFs (red: estimated; blue: true). (b) Extracted signal modes (blue: extracted; green: true). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

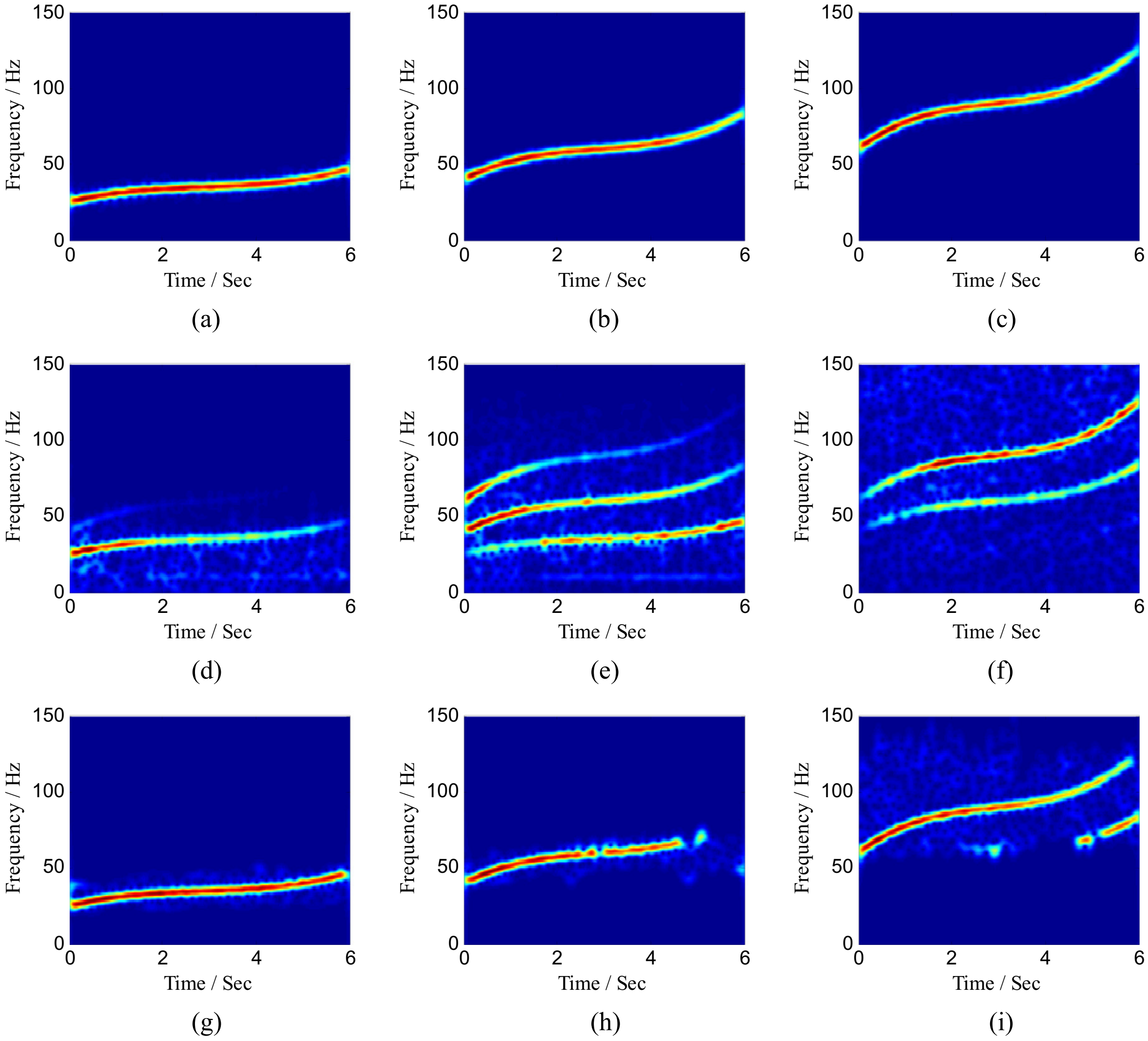


Fig. 9. TFDs of the extracted modes for signal (40) by different methods. (a)–(c) show the TFDs of the three extracted modes by ACMP. (d)–(f) show the results of EEMD. (g)–(i) provide the results of TVF-EMD.

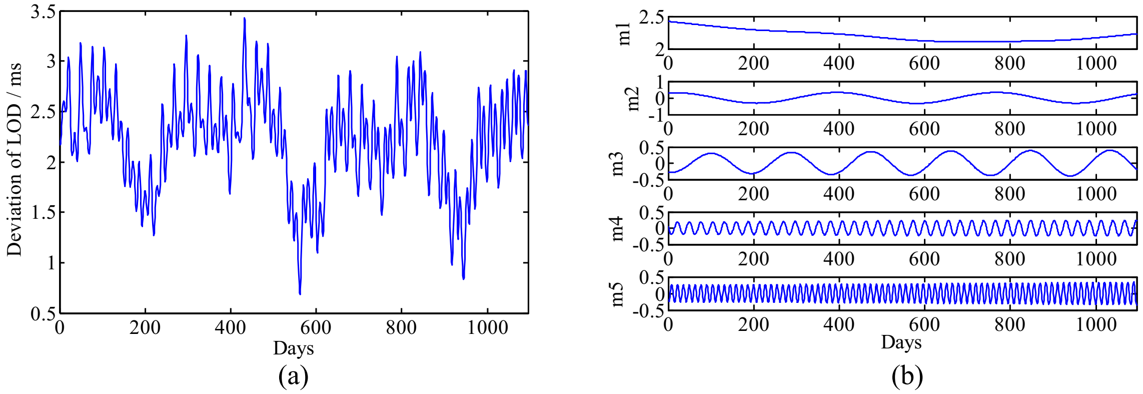


Fig. 10. LOD data and the decomposition results by ACMP. (a) LOD data. (b) The 5 extracted signal modes by ACMP.

annual cycle. Similarly, m3 reflects half-year cycle while m4 and m5 relate to the monthly and half-month tides, respectively. Herein the cost time of the ACMP is about 1.3 s.

4.5. Application to vibration signals of a rotor test rig

For rotating machineries, the vibration response is usually excited by the rotational motion of the rotor. Therefore, the frequency contents of the vibration signal are often closely related to the rotating speed. In addition, rotating machineries often undergo non-stationary operations such as start-up and shut-down processes and the vibration response during these non-stationary processes contains abundant information about the condition of the machinery. Consequently, estimating instantaneous speed is very important for the fault diagnosis and condition monitoring (e.g., order analysis) of varying-speed rotating machineries [44]. In industrial applications, a tachometer is usually used to measure the speed [45]. However,

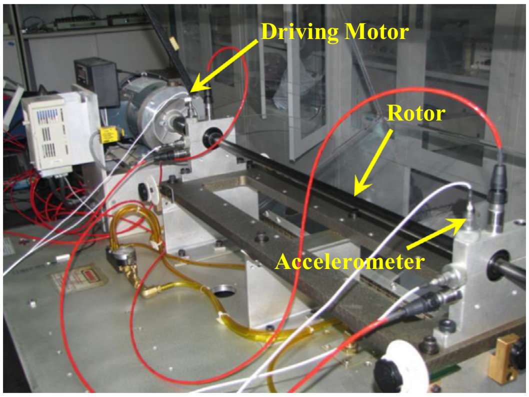


Fig. 11. Bently rotor test rig.

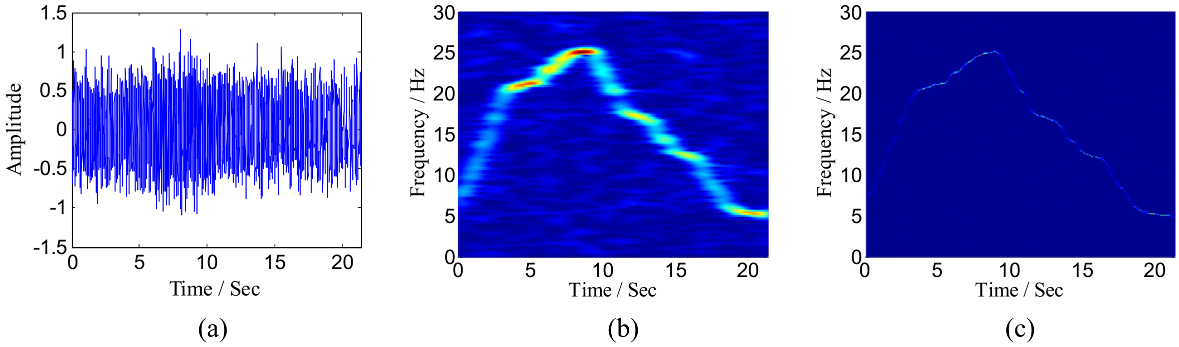


Fig. 12. Vibration signal 1 from the rotor test rig. (a) Waveform. (b) TFD by STFT. (c) TFD by SST.

this method requires additional hardware and device. Since the base frequency of the vibration response can accurately reflect the rotating speed, a more feasible method for instantaneous rotating speed estimation is to directly estimate the IF of the vibration signal. Generally, the IF of a non-stationary signal is estimated by detecting peaks of a TFD [46]. The resolution of the TFD is limited by the Heisenberg's uncertainty principle, which will seriously influence the accuracy of the IF estimation. As shown in Fig. 8(a), the proposed ACMP can accurately estimate the IFs without dealing with the TFD. Accordingly, herein the ACMP is applied to analyze real vibration signals and estimate theirs IFs.

The non-stationary vibration signals considered here is measured by the accelerometer on a Bently rotor test rig [47] during complicated speed-up and shut-down processes. The rotor test rig is shown in Fig. 11. We collected two vibration signals referred to as vibration signal 1 and vibration signal 2, respectively. The sampling frequency is set to 100 Hz for both signals. Vibration signal 1 is shown in Fig. 12. It only contains one signal component, i.e., the rotating-frequency component (see

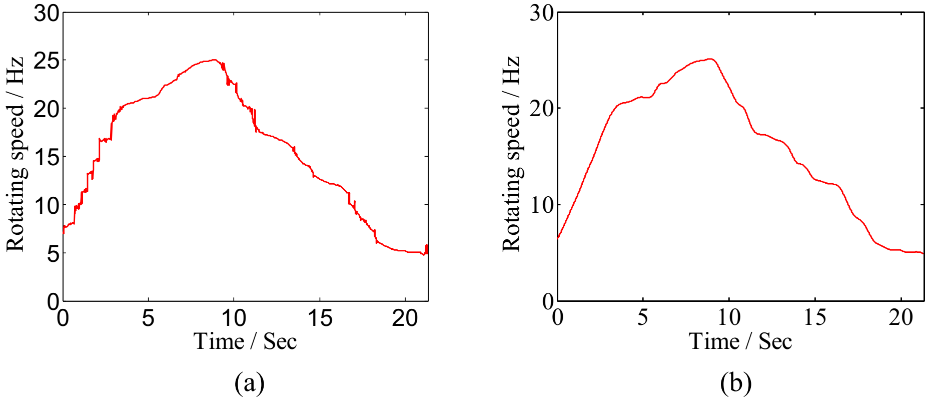


Fig. 13. Estimated rotating speeds for vibration signal 1 by (a) SST and (b) ACMP.

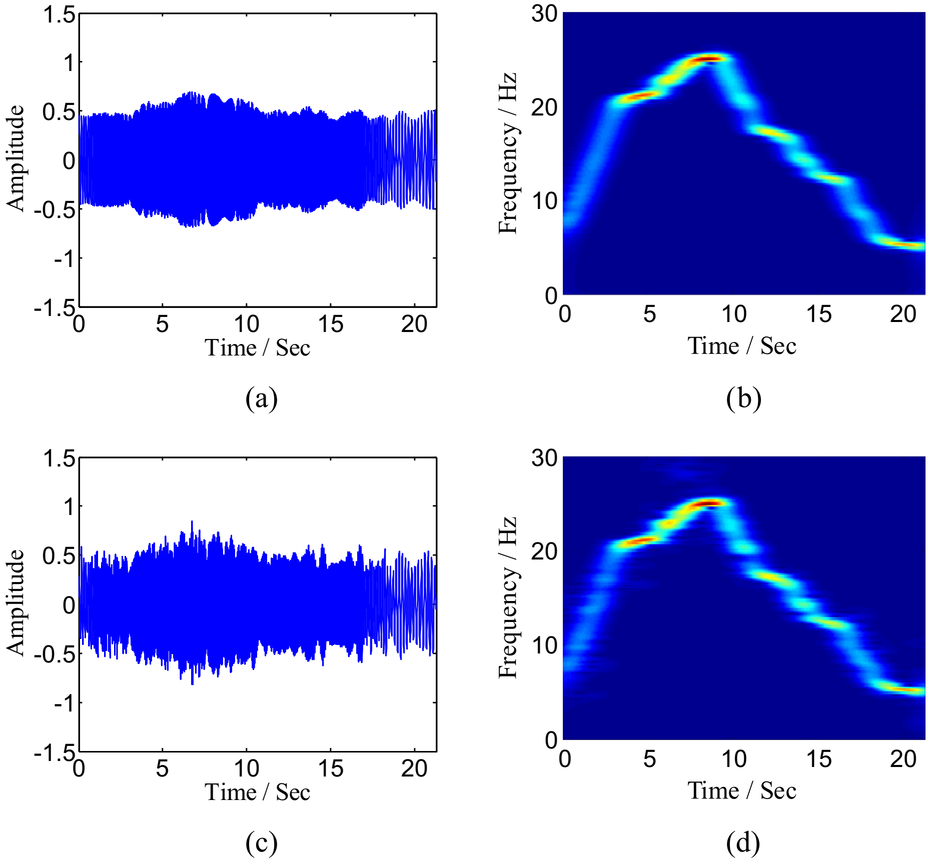


Fig. 14. Reconstructed (or de-noised) vibration signal 1 by different methods. (a) and (b) show the waveform and the TFD (by STFT) of the reconstructed signal by ACMP. (c) and (d) show the results by BT.

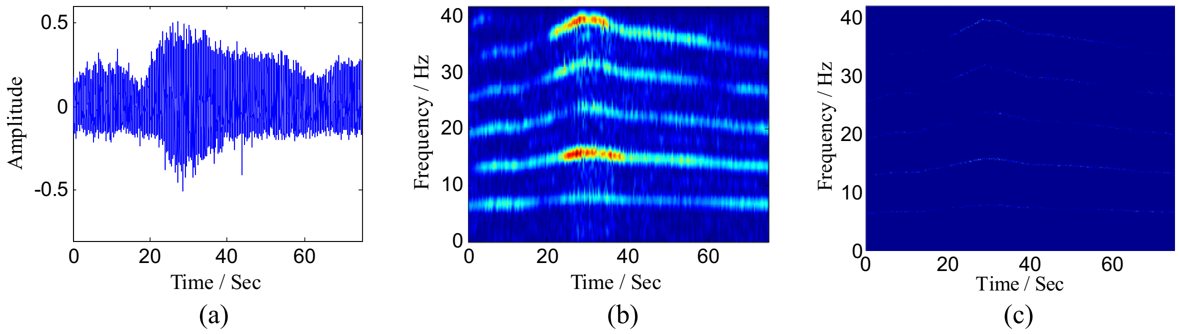


Fig. 15. Vibration signal 2 from the rotor test rig. (a) Waveform. (b) TFD by STFT. (c) TFD by SST.

Fig. 12(b)). It can be seen that the signal is contaminated by strong noise and the rotating speed is rapidly varying with time. In this application, besides the STFT, the SST is also used to generate a higher-resolution TFD for the signal, as shown in Fig. 12(c). It can be found that the TFD by SST has much better resolution than that by STFT. However, the SST is less effective to characterize signals with fast frequency modulations and therefore it suffers from energy diffusions in some areas of Fig. 12(c) (e.g., during 0 s–5 s; the IF is fast varying in this time interval). The SST and the proposed ACMP are used to estimate the IF (i.e., the rotating speed) for the vibration signal, as shown in Fig. 13. Herein the SST estimates the IF by detecting the TFD peaks using the method in [48]. It can be observed that the estimated IF (or speed) trajectory by SST is discontinuous and shows staircase effect due to the energy diffusion of the TFD. Conversely, the ACMP obtains a smooth IF curve and effectively captures enough details including some slight fluctuations of the speed (see Fig. 13(b)). In addition, different from parameterized methods that rely on a predefined model (for the speed curve) [49], the ACMP is fully nonparametric and thus can adapt to various complex speed trajectories. The reconstructed signal by ACMP is illustrated in Fig. 14(a) and (b). It shows that the noise is effectively suppressed (compared to the raw signal in Fig. 12). Note that the ACMP can effectively extract desired signal modes from the noise and therefore it can be used as a de-noising tool especially for non-stationary signals with time-varying frequency contents. For comparison, the de-noising result by the block thresholding (BT) [50], a well-known TF de-noising technique, is provided in Fig. 14(c) and (d). Although the BT shows a good de-noising result, it cannot completely remove the noise close to the signal mode. The reason is that the BT does not fully take into account the IF pattern of the considered signal. For this example, the run times of the ACMP, SST, and the BT are 0.7 s, 4.2 s and 0.1 s, respectively.

Vibration signal 2 mainly contains 5 signal modes which show a harmonic relation as shown in Fig. 15. Since the IFs of this signal do not change as fast as that of vibration signal 1, the SST can generate a more concentrated TFD for this signal (see Fig. 15(c)). The estimated IFs for this signal by SST and ACMP are provided in Fig. 16. Due to the influence of the noise, the SST may smear the energies of some weak portions of the signal. Therefore, some detected TFD peaks may belong to the noise instead of the signal (e.g., the detected IF curve in the black circle in Fig. 16(a)). On the contrary, the ACMP has good anti-noise capability and can effectively extract smooth IF curves as shown in Fig. 16(b). The extracted 5 signal modes by ACMP are shown in Fig. 17. It can be found that the amplitudes of these signal modes show distinct features which can be further investigated for the fault diagnosis purpose. The de-noising results for this vibration signal by ACMP and BT

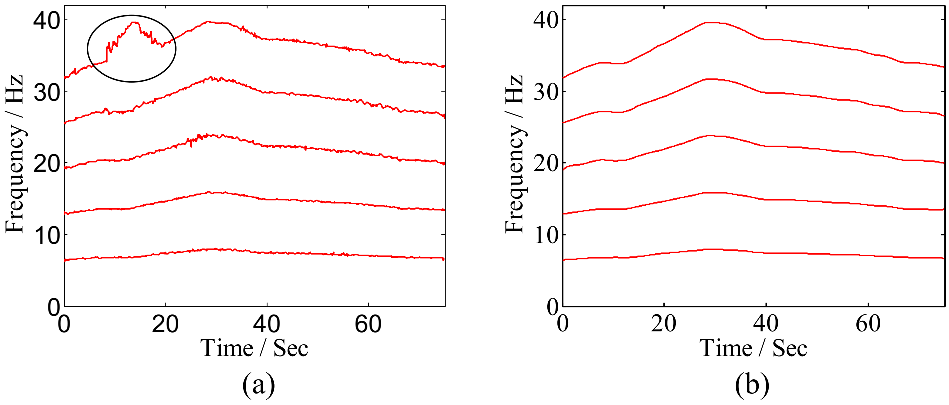


Fig. 16. Estimated IFs for vibration signal 2 by (a) SST and (b) ACMP.

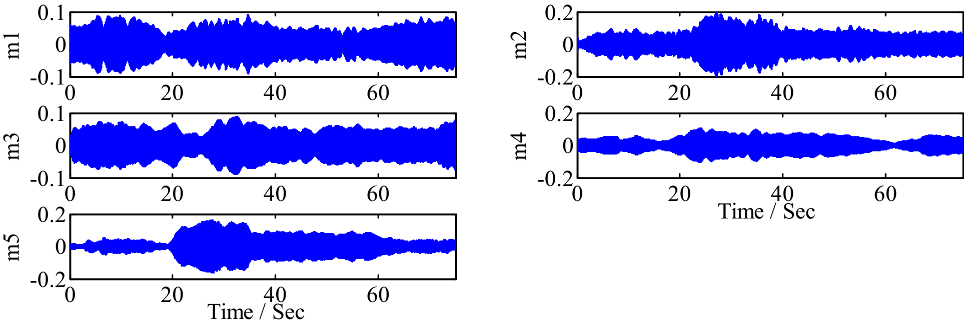
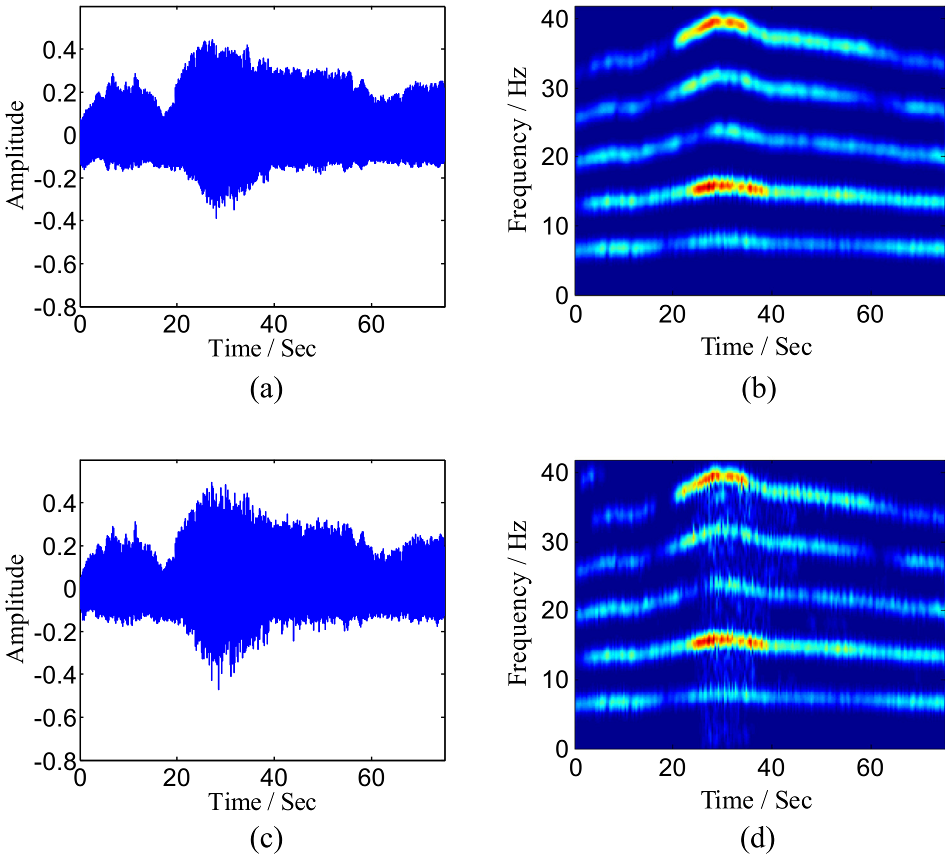


Fig. 17. Estimated signal modes for vibration signal 2 by ACMP (m1–m5 are arranged from low frequency to high frequency).



(a)

(b)

(c)

(d)

Fig. 18. Reconstructed (or de-noised) vibration signal 2 by different methods. (a) and (b) show the waveform and the TFD (by STFT) of the reconstructed signal (i.e., sum of the estimated signal modes in Fig. 17) by ACMP. (c) and (d) show the results by BT.

are shown in Fig. 18. The ACMP de-noises the signal by effectively separating each signal mode and the noise using an adaptive filter, and thus shows satisfactory results (see Fig. 18(a) and (b)). For BT, since the real-life noise is non-Gaussian and non-white, some noise is still retained and some weak signal portions are discarded by BT (see Fig. 18(d), the mode in 30 Hz–40 Hz looks discontinuous due to the loss of some effective portions). For this example, the cost times of the ACMP, SST and the BT are 22.1 s, 205.2 s and 0.9 s, respectively. The two examples in this subsection show the promising advantages of the ACMP in analyzing vibration signals of varying-speed rotating machineries.

5. Conclusion

In this paper, a novel signal decomposition method called ACMP was proposed. The ACMP included three main ingredients: a recursive mode extraction framework, an adaptive bandwidth updating rule and an IF initialization technique based on Hilbert transform. Some simulated examples and the comparison results with the state-of-art methods (i.e., VNCMD, EMD, EEMD, TVF-EMD, and EWT) were provided showing that the ACMP has better filtering and convergence properties in a noisy environment, has better separation ability for two close signal modes, and is more capable of analyzing non-stationary signals. Finally, the ACMP was applied to analyze the LOD data and vibration signals of a rotor test rig indicating its usefulness in industrial applications. In the future, more specific and systematic application researches based on the ACMP will be investigated (e.g., the order tracking for rotating machineries).

Although the ACMP has some promising advantages by using a recursive mode extraction scheme, it cannot properly deal with signals with crossing IFs. The ACMP is mainly confronted with two issues when analyzing crossing signal modes. Firstly, the IF initialization method in Section 3.3 cannot provide good initial IFs for the crossing signal modes. The obtained IFs may switch to wrong signal modes after the crossing point. To address this issue, one may initialize the IF using an improved TF ridge extraction method (e.g., the one in [48]). Most importantly, the recursive mode extraction scheme cannot identify the energy contributions of the crossing modes at the crossing point. Accordingly, there will be large estimation errors for the obtained signal modes [45]. For this issue, a joint-mode-estimation technique such as that of the STNBMD [34] and the VNCMD [35] should be employed to accurately reconstruct the crossing modes. Our future work will be devoted to extending the ACMP to more challenging cases.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 11632011.

References

[1] D. Iatsenko, P.V.E. Mcclintock, A. Stefanovska, Nonlinear mode decomposition: a noise-robust, adaptive decomposition method, Phys. Rev. E 92 (3) (2015) 032916.

[2] Y. Yang, Z. Peng, G. Meng, W. Zhang, Characterize highly oscillating frequency modulation using generalized Warblet transform, Mech. Syst. Signal Process. 26 (2012) 128–140.

[3] J. Cheng, Y. Yang, D. Yu, The envelope order spectrum based on generalized demodulation time–frequency analysis and its application to gear fault diagnosis, Mech. Syst. Signal Process. 24 (2) (2010) 508–521.

[4] Z. Feng, X. Chen, M. Liang, F. Ma, Time–frequency demodulation analysis based on iterative generalized demodulation for fault diagnosis of planetary gearbox under nonstationary conditions, Mech. Syst. Signal Process. 62 (2015) 54–74.

[5] R.U. Maheswari, R. Umamaheswari, Trends in non-stationary signal processing techniques applied to vibration analysis of wind turbine drive train–A contemporary survey, Mech. Syst. Signal Process. 85 (2017) 296–311.

[6] Y. Doweck, A. Amar, I. Cohen, Joint model order selection and parameter estimation of chirps with harmonic components, IEEE Trans. Signal Process. 63 (7) (2015) 1765–1778.

[7] G. Yu, Y. Zhou, General linear chirplet transform, Mech. Syst. Signal Process. 70 (2016) 958–973.

[8] Z. Feng, X. Lin, M.J. Zuo, Joint amplitude and frequency demodulation analysis based on intrinsic time-scale decomposition for planetary gearbox fault diagnosis, Mech. Syst. Signal Process. 72 (2016) 223–240.

[9] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. Royal Soc. A, Math., Phys Eng. Sci. 454 (1971) (1998) 903–995.

[10] Y. Lei, J. Lin, Z. He, M.J. Zuo, A review on empirical mode decomposition in fault diagnosis of rotating machinery, Mech. Syst. Signal Process. 35 (1–2) (2013) 108–126.

[11] H. Huang, J. Pan, Speech pitch determination based on Hilbert-Huang transform, Signal Process. 86 (4) (2006) 792–803.

[12] X. Zhao, T.H. Patel, M.J. Zuo, Multivariate EMD and full spectrum based condition monitoring for rotating machinery, Mech. Syst. Signal Process. 27 (1) (2012) 712–728.

[13] Z. Wu, N.E. Huang, Ensemble empirical mode decomposition: a noise-assisted data analysis method, Adv. Adapt. Data Anal. 1 (01) (2009) 1–41.

[14] H. Li, Z. Li, W. Mo, A time varying filter approach for empirical mode decomposition, Signal Process. 138 (2017).

[15] Q. He, Time-frequency manifold for nonlinear feature extraction in machinery fault diagnosis, Mech. Syst. Signal Process. 35 (1–2) (2013) 200–218.

[16] I. Daubechies, J. Lu, H. Wu, Synchrosqueezed Wavelet transforms: an empirical mode decomposition-like tool, Appl. Computat. Harmon. Anal. 30 (2) (2011) 243–261.

[17] D.-H. Pham, S. Meignen, High-order synchrosqueezing transform for multicomponent signals analysis—With an application to gravitational-wave signal, IEEE Trans. Signal Process. 65 (12) (2017) 3168–3178.

[18] S. Wang, X. Chen, I.W. Selesnick, Y. Guo, C. Tong, X. Zhang, Matching synchrosqueezing transform: a useful tool for characterizing signals with fast varying instantaneous frequency and application to machine fault diagnosis, Mech. Syst. Signal Process. 100 (2018) 242–288.

[19] G. Yu, M. Yu, C. Xu, Synchroextracting transform, IEEE Trans. Ind. Electron. 64 (10) (2017) 8042–8054.

[20] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, IEEE Trans. Signal Process. 62 (3) (2014) 531–544.

[21] X. Hu, S. Peng, W.L. Hwang, Adaptive integral operators for signal separation, IEEE Signal Process. Lett. 22 (9) (2015) 1383–1387.

[22] J. Gilles, Empirical Wavelet transform, IEEE Trans. Signal Process. 61 (16) (2013) 3999–4010.

[23] J. Pan, J. Chen, Y. Zi, Y. Li, Z. He, Mono-component feature extraction for mechanical fault diagnosis using modified empirical wavelet transform via data-driven adaptive Fourier spectrum segment, Mech. Syst. Signal Process. 72 (2016) 160–183.

[24] A. Cicone, J. Liu, H. Zhou, Adaptive local iterative filtering for signal decomposition and instantaneous frequency analysis, Appl. Computat. Harmon. Anal. 41 (2) (2016) 384–411.

[25] M. Feldman, Time-varying vibration decomposition and analysis based on the Hilbert transform, J. Sound Vib. 295 (3–5) (2006) 518–530.

[26] F. Gianfelici, G. Biagetti, P. Crippa, C. Turchetti, Multicomponent AM–FM representations: an asymptotically exact approach, IEEE Trans. Audio, Speech, Lang. Process 15 (3) (2007) 823–837.

[27] S. Peng, W.L. Hwang, Adaptive signal decomposition based on local narrow band signals, IEEE Trans. Signal Process. 56 (7) (2008) 2669–2676.

[28] S. Peng, W.L. Hwang, Null space pursuit: an operator-based approach to adaptive signal separation, IEEE Trans. Signal Process. 58 (5) (2010) 2475–2483.

[29] T.Y. Hou, Z. Shi, Adaptive data analysis via sparse time-frequency representation, Adv. Adapt. Data Anal. 3 (01n02) (2011) 1–28.

[30] T.Y. Hou, Z. Shi, Data-driven time–frequency analysis, Appl. Computat. Harmon. Anal. 35 (2) (2013) 284–308.

[31] T.Y. Hou, Z.Q. Shi, Sparse time-frequency representation of nonlinear and nonstationary data, Sci. China Math. 56 (12) (2013) 2489–2506.

[32] J. Harmouche, D. Fourer, F. Auger, P. Borgnat, P. Flandrin, The sliding singular spectrum analysis: a data-driven nonstationary signal decomposition tool, IEEE Trans. Signal Process. 66 (1) (2017) 251–263.

[33] S. Chen, Y. Yang, K. Wei, X. Dong, Z. Peng, W. Zhang, Time-varying frequency-modulated component extraction based on parameterized demodulation and singular value decomposition, IEEE Trans. Instrum. Meas. 65 (2) (2016) 276–285.
[34] S.I. Mcneill, Decomposing a signal into short-time narrow-banded modes, J. Sound Vib. 373 (2016) 325–339.
[35] S. Chen, X. Dong, Z. Peng, W. Zhang, G. Meng, Nonlinear chirp mode decomposition: a variational method, IEEE Trans. Signal Process. 65 (22) (2017) 6024–6037.
[36] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[37] Q. Lv, D. Ye, S. Qiao, Y. Salamin, J. Huangfu, C. Li, L. Ran, High dynamic-range motion imaging based on linearized doppler radar sensor, IEEE Trans. Microw. Theory Techn. 62 (9) (2014) 1837–1846.
[38] M.P. Tarvainen, P.O. Rantaaho, P.A. Karjalainen, An advanced detrending method with application to HRV analysis, IEEE Trans. Biomed. Eng. 49 (2) (2002) 172–175.
[39] B. Picinbono, On instantaneous amplitude and phase of signals, IEEE Trans. Signal Process. 45 (3) (1997) 552–560.
[40] D. Wei, A.C. Bovik, On the instantaneous frequencies of multicomponent AM-FM signals, IEEE Signal Process. Lett. 5 (4) (1998) 84–86.
[41] B. Guo, S. Peng, X. Hu, P. Xu, Complex-valued differential operator-based method for multi-component signal separation, Signal Process. 132 (2017) 66–76.
[42] G. Rilling, P. Flandrin, One or two frequencies? The empirical mode decomposition answers, IEEE Trans. Signal Process. 56 (1) (2008) 85–95.
[43] IERS, Earth Orientation Parameters, (2010) [Online]. Available: http://hpiers.obspm.fr/eoppc/eop/eopc04/eopc04.62-now.
[44] J. Urbanek, T. Barszcz, J. Antoni, A two-step procedure for estimation of instantaneous rotational speed with large fluctuations, Mech. Syst. Signal Process. 38 (1) (2013) 96–102.
[45] M.-C. Pan, Y.-F. Lin, Further exploration of Vold–Kalman-filtering order tracking with shaft-speed information—II: engineering applications, Mech. Syst. Signal Process. 20 (6) (2006) 1410–1428.
[46] Z. Feng, M. Liang, F. Chu, Recent advances in time–frequency analysis methods for machinery fault diagnosis: a review with application examples, Mech. Syst. Signal Process. 38 (1) (2013) 165–205.
[47] Y. Yang, X. Dong, Z. Peng, W. Zhang, G. Meng, Vibration signal analysis using parameterized time–frequency method for features extraction of varying-speed rotary machinery, J. Sound Vib. 335 (2015) 350–366.
[48] S. Chen, X. Dong, G. Xing, Z. Peng, W. Zhang, G. Meng, Separation of overlapped non-stationary signals by ridge path regrouping and intrinsic chirp component decomposition, IEEE Sensors J. 17 (18) (2017) 5994–6005.
[49] Y. Yang, Z. Peng, X. Dong, W. Zhang, D.A. Clifton, Component isolation for multi-component signal analysis using a non-parametric gaussian latent feature model, Mech. Syst. Signal Process. 103 (2018) 368–380.
[50] G. Yu, S. Mallat, E. Bacry, Audio denoising by time-frequency block thresholding, IEEE Trans. Signal Process. 56 (5) (2008) 1830–1839.