



OPEN

Nonstationary signal extraction based on BatOMP sparse decomposition technique

Shuang-chao Ge & Shida Zhou

Sparse decomposition technique is a new method for nonstationary signal extraction in a noise background. To solve the problem of accuracy and efficiency exclusive in sparse decomposition, the bat algorithm combined with Orthogonal Matching Pursuits (BatOMP) was proposed to improve sparse decomposition, which can realize adaptive recognition and extraction of nonstationary signal containing random noise. Two general atoms were designed for typical signals, and dictionary training method based on correlation detection and Hilbert transform was developed. The sparse decomposition was turned into an optimizing problem by introducing bat algorithm with optimized fitness function. By contrast with several relevant methods, it was indicated that BatOMP can improve convergence speed and extraction accuracy efficiently as well as decrease the hardware requirement, which is cost effective and helps broadening the applications.

Background information. Signal extraction and signal–noise separation are always two of the research focuses in modern signal processing¹, which are commonly used in biomedical signal features extraction, vibration signal analysis, seismic signal detection², sound signals recognition³. In practical applications, such signals are often submerged in a variety of vibration or electromagnetic noise, and the occurrence times of the target signals are random, which are typical nonstationary signals. Fourier transform is one of the most classic signal analysis and extraction method, but it cannot accurately describe nonstationary signals⁴. In recent years, new theories and technologies continue to appear in signal extraction. For example, wavelet decomposition (WD)^{5,6}, empirical mode decomposition⁷, Hilbert–Huang Transform (HHT), variational mode decomposition (VMD) algorithm⁸. These methods need to meet certain conditions to work, for example the decomposition levels, modal number, and termination thresholds.

To achieve a more flexible, concise and adaptive signal decomposition, researchers proposed sparse decomposition. This method represents the signal with as few atoms as possible in a given redundant dictionary by matching pursuit (MP) algorithm⁹, which is a greedy algorithm for sparse decomposition. Various new evaluation criteria and basis pursuit, orthogonal matching pursuit algorithm (OMP)¹⁰, and time–frequency spectrum segmentation methods¹¹ were generated to select a set of optimal atoms from the constructed over-complete dictionary. In principle, if the dictionary redundancy is high enough and the iterations is large enough, the target signal can be perfectly extracted by OMP. On this basis, some general improved algorithms were proposed for example Regularized Orthogonal MP (ROMP)¹² and Compressive Sampling MP (CoSaMP)¹³. These methods require the signal Sparsity K for efficient execution, but K is generally unknown in practice. Sparsity Adaptive MP (SAMP) was proposed for signal reconstruction without prior information of the sparsity, but it is more complex than other greedy algorithms under large sparsity level¹⁴. And improper initial step size will lead to excessive decomposition for SAMP. The accuracy of signal sparse decomposition mainly depends on the redundancy and refinement accuracy of the redundant dictionary. Over or under estimation as well as long-time running will appear in these algorithms under the condition of large sparsity. Generally, the greater the redundancy and refinement, the greater the probability of accurate signal decomposition. However, for the greedy algorithm mentioned above, these are at the cost of algorithm efficiency. The accuracy and efficiency are exclusive.

Aiming at two main research hotspots including sparse decomposition algorithm and over-complete atom dictionary of signal sparse decomposition¹⁵, we designed two typical universal atoms, and proposed an adaptive feature-based atom construction method for the extraction of non-stationary signals with unknown sparsity. Redundant dictionary is obtained by extending the feature-based atoms, which can balance the completeness and redundancy. A signal matching tracking extraction algorithm was developed based on the bat algorithm

School of Instruments and Electronics, North University of China, Taiyuan 030051, China. email: geshch@nuc.edu.cn

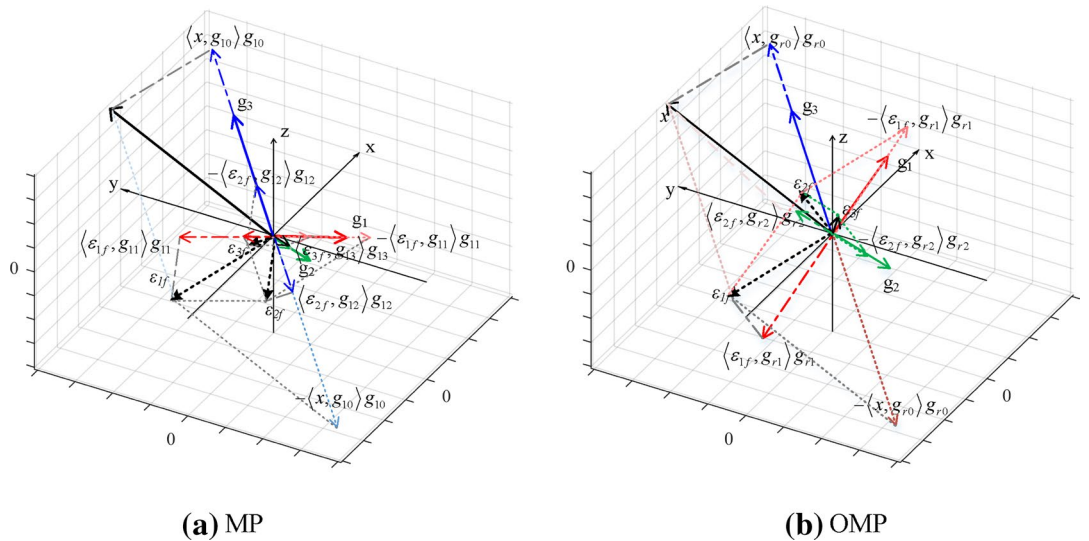


Figure 1. Convergence process of MP and OMP.

and OMP, which successfully combined the accuracy and efficiency and could effectively realize nonstationary time domain signal extraction.

Classical signal sparse decomposition algorithms. Signal sparse decomposition represents a signal by specific combinations of some atoms in a dictionary. For a given dictionary, the optimal combination can be accurately determined when all possible combinations were calculated. However, exhausting all combinations in a dictionary is a non-deterministic polynomial problem that is almost impossible to achieve for large dictionary bases. So, the requirement was changed to finding a suboptimal combination from the dictionary with the lowest possible number of atoms and the smallest possible extraction error. This will reduce the computational complexity significantly, and the MP algorithm is one of the algorithms that can achieve this requirement.

Assume that the represented signal is x with length of N . Let R denote the Hilbert space in which a dictionary matrix D composed of a set of vectors $\{g_1, g_2, \dots, g_n\}$. Each vector is an atom with the same length N and these vectors have been treated as normalized as $\|g_i\|_2 = 1$.

With $\xi_i = x$, the MP algorithm selects one atom at a time from the dictionary matrix D that best matches x , satisfying (1),

$$c_i = |\langle x \cdot g_{i_{best}} \rangle| = \max_{i \in \{1, \dots, n\}} |\langle x \cdot g_i \rangle|, \tag{1}$$

where i_{best} is the index of the best matching atom in D . $\langle \cdot \rangle$ is the inner product function.

The signal x is then decomposed into two parts, a sparse approximation \hat{x} and an approximation residual ξ_2 :

$$x = \hat{x} + \xi_2 = \langle x \cdot g_{i_{best}} \rangle g_{i_{best}} + \xi_2. \tag{2}$$

Continues to select the atoms that best matches ξ_2 , iterating repeatedly and eventually the signal x can be approximated as a linear sum of these atoms:

$$\hat{x} = \sum_{i=0}^{n-1} \langle \xi_i \cdot g_{i_{best}} \rangle g_{i_{best}}, \quad x - \hat{x} = \xi_n. \tag{3}$$

For MP algorithm, the non-orthogonality between the vertical projection of the signal (or residuals) on the selected atoms and the residuals will lead to suboptimal iterative results instead of the best optimal, and convergence requires many iterations. The OMP algorithm is the orthogonalization of all selected atoms at each step of the decomposition, which makes the convergence faster with the same accuracy requirement. The convergence process of MP and OMP are described by a dictionary D with length of three, as shown in Fig. 1. However, although the OMP algorithm reduced iterations to some extent, it had to calculate the current residual and the inner product of all atoms within the current dictionary during each iteration, resulting in unsatisfied effectiveness. Therefore, this paper introduced the bat algorithm (BA) to optimize the matching tracking algorithm.

Bat algorithm presentation. The basic flow of bat algorithm is as follows:

- (1) Initialization: the best fitness Fit_{best} , bat population number N_{pop} , the max bat generation N_{gen} , the current generation $n=0$, initial flight frequency $f_0 = \{f_i^0 | i=1, 2, \dots, N_{pop}\}$, acoustic loudness $A^0 = \{A_i^0 | i=1, 2, \dots, N_{pop}\}$ and pulse emission frequency $r_0 = \{r_i^0 | i=1, 2, \dots, N_{pop}\}$. The initial location of the bat colony is randomly generated according to (4):

$$P^0 = \{P_i^0 | i = 1, 2, \dots, N_{pop}\}, \quad P_i^0 \in (P_{\min} P_{\max}), \quad (4)$$

- (2) The best position P_{best}^n is determined by the fitness function Fit^n .

$$Fit^n(P_{best}^n) = \arg \min_{N_{pop}} (Fit^n). \quad (5)$$

- (3) Update the velocity and position of the individual bat:

$$\begin{cases} f_i = f_{\min} + r_1 \times (f_{\max} - f_{\min}) \\ v_i^n = v_i^{n-1} + (P_{best} - P_i^{n-1}) \cdot f_i, \\ P_i^n = P_i^{n-1} + v_i^n \end{cases} \quad (6)$$

where r_1 was a random number, satisfying $r_1 \in [0, 1]$; f_i was the search pulse frequency of the i -th bat; v_i^n denoted the velocity of the i -th bat in the n -th iteration, P_i^n denoted the position of the i -th bat in the n -th iteration; and P_{best}^n is the current global optimal solution.

- (4) Generate a random number $r_{2i} \in [0, 1]$ for each bat and update bat position according to (7).

$$\begin{cases} \text{global optimization} : P_i^{n+1} = P_i^n + v_i^{n+1}, r_{2i} \leq r_i^n \\ \text{local optimization} : P_i^{n+1} = P_i^n + \lambda_{ri} \bar{A}^n * (P_{\max} - P_{\min}), r_{2i} > r_i^n \end{cases} \quad (7)$$

where: η was a random number, satisfying $\eta \in [-1, 1]$ and \bar{A}^n was the mean fitness of the bat population.

- (5) Update the fitness

$$Fit^{n+1}(i) = Fit(P_i^{n+1}). \quad (8)$$

- (6) For each bat, a random number r_{3i} is generated, and update the position:

$$\begin{cases} P_i^{n+1} = P_i^{n+1}, r_{3i} > A_i^n \&\& Fit_i^{n+1} < Fit_i^n \\ P_i^{n+1} = P_i^n, Fit_i^{n+1} = Fit_i^n, \text{ otherwise} \end{cases} \quad (9)$$

- (7) The fitness and pulse emission frequency are updated:

$$\begin{cases} r_i^{n+1} = r_i^0 (1 - e^{-\gamma^n}) \\ A_i^{n+1} = \lambda A_i^n \end{cases} \quad (10)$$

where, $\lambda \in (0, 1)$, $\gamma > 0$, when $n \rightarrow \infty$, $A_i^n \rightarrow 0$, $r_i^n \rightarrow r_0$.

- (8) Find the current matching atom based on the optimal solution.
 (9) The random perturbation of the current optimal solution in step 4 can effectively avoid the iterative result from falling into a local optimal solution, which helped to find the global optimal solution fast and accurate.
 (10) The Ackley function is used to test the BA. The expression of the Ackley function is as follows:

$$f(x) = -c_1 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{j=1}^n x_j^2}\right) - \exp\left(\frac{1}{n} \sum_{j=1}^n \cos(2\pi x_j)\right) + e. \quad (11)$$

In this study, $n=2$, $c_1=20$, $e=2.71289$. the Ackley function was taken as the fitness function and the global minimum of this function was searched by the above methods. The particle swarm optimization (PSO)¹⁶, artificial fish school algorithm (AFSA)¹⁷ and Cuckoo Search (CS)¹⁸ are used for comparison. The population size and iteration numbers of these intelligent algorithms are the same to ensure rigorous comparison. The search paths and results are shown in Fig. 2.

The detailed values are shown in Table 1. The comparison of the tracking trajectory and the optimization results show that BA has advantage of high convergence speed and computational accuracy because the gradient of the optimization deviation is the largest and the optimization results are closest to the true value.

Methods

BatOMP sparse decomposition. General atomics designed for typical signals. For sinusoidal-like and one-sided decaying oscillatory signals, g -atoms are constructed:

$$g(c, d, t_1, t_2, \tau, f, \varphi) = \begin{cases} ce^{-d(t-t_1)} \cos(2\pi f(t-t_1) + \varphi), & t \in [t_1, t_2] \\ 0, & \text{others} \end{cases} \quad (12)$$

where c is the normalization factor to ensure that the original signal has the same energy as its sparse decomposition results; d : the attenuation factor; t : the sampling time; t_1 : start point of atomic appearance; t_2 : the ending point; f : signal frequency and φ : phase. The time domain waveforms of g -atoms with different parameters are shown in Fig. 3.

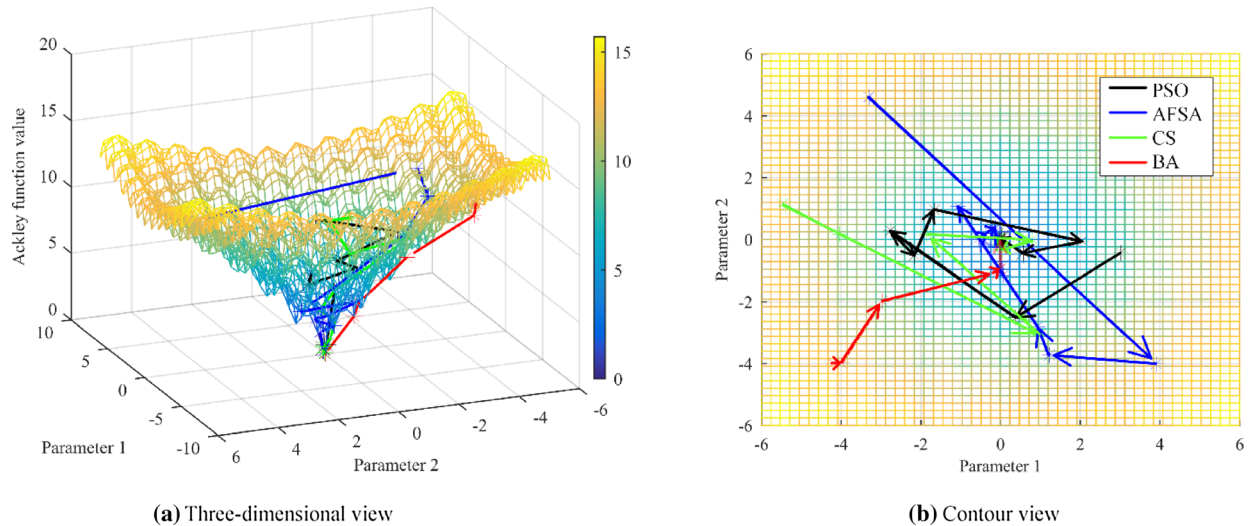


Figure 2. The optimal trajectory of different methods. (a,b) Show the 3D view and contour attempt of the optimal trajectory, respectively. Colors: black: the merit-seeking trajectories of PSO, blue: the merit-seeking trajectories of AFSA, green: the merit-seeking trajectories of CS, red: the merit-seeking trajectories of BA.

Algorithm	Search results	Optimal solution	Time running
PSO	[0.0644, 0.0518]	0.4025	0.484925
AFSA	[0.0011, 0.0052]	0.01583	0.585477
CS	[0.0083, 0.0198]	0.06764	0.254225
BA	[0.0023, -0.0014]	0.007828	0.154801

Table 1. Results analysis of different methods. The ideal search results are [0, 0] and ideal solution is 0.

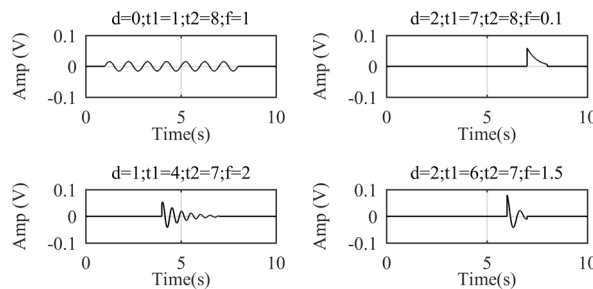


Figure 3. Time domain waveforms of g -atoms with different parameters.

When the attenuation factor $d=0$, the g -atom degenerates to standard sine wave; when d increases, the g -atom performs sinusoidal damped oscillation. Therefore, this atom has a strong match with sinusoidal signals, and single-sided oscillatory decay signals.

For kind of triangle waves, charge–discharge waves, and bilateral decay oscillation signals, t_r atoms are constructed:

$$t_r(c, d_1, d_2, t_0, t_1, t_2, f, \varphi, \alpha) = \begin{cases} c \times e^{(-d_1*(t_1-t))} \times \cos(2\pi f(t_1 - t) + \varphi), & t \in [t_0, t_1] \\ 0, & \text{others} \\ \eta \times c \times e^{(-d_2*(t-t_1))} \times \cos(2\pi f(t - t_1) + \varphi), & t \in [t_1, t_2] \end{cases} \quad (13)$$

where d_1 and d_2 are the bilateral damping factors; t_0 is the bilateral boundary of the atomic; $[t_1, t_2]$ is the atomic time range; and η is the bilateral scaling factor.

The time domain waveforms of t_r -atoms with different parameters are shown in Fig. 4. When the bilateral scaling factor $\eta=0$, the t_r -atom degenerates to single-sided oscillating atom (reverse-order g -atom); when $0 < \eta < 1$ and the atomic frequency is low enough, the t_r -atom behaves as a charge–discharge triangle wave; when $\eta=1$, the t_r -atom with low-frequency behaves as a triangle-like wave, and behaves as bilateral oscillating decay signal with high-frequency.

The above analysis shows that the constructed g -atoms and t_r -atoms are very flexible and could match almost typical testing signals by parameter adjustment.

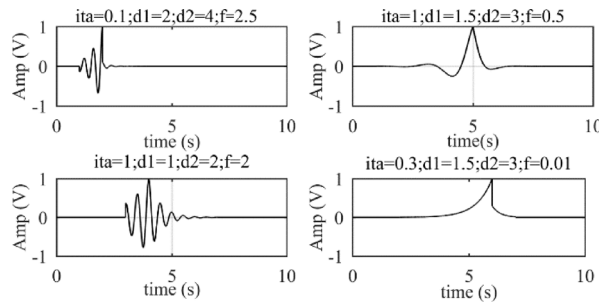


Figure 4. Time domain waveforms of t_r -atoms with different parameters. ‘ita’ represents η .

Dictionary training algorithm. A dictionary training algorithm based on feature parameters was studied to determine the key parameters of feature base-atom and to balance the completeness and redundancy for redundant dictionary library construction.

When constructing the redundancy dictionary, we firstly use the short-time Fourier transform to initially determine the frequency f_s and phase ϕ_s of the target signal in the original data x . Then, standard sine atom $s_0 = \sin(2\pi f_s t + \phi_s)$ is constructed. The correlation detection technique is used by calculating the correlation function of the original data and the sine atom, and the upper and lower envelopes of the correlation function are obtained by searching the positive and negative peaks of the correlation function. The points of the maximum positive gradient between the positive and negative peaks are the extracted range of the target signal. Thus, the signal sample E_i in the original observation sequence is extracted. The time information section obtained by the above calculation determines the time domain parameters such as t_0 , t_1 and t_2 of the characteristic atom.

Accurate time and frequency domain parameters are obtained by Hilbert transform of E_i .

$$R(t) = E_i(t) \times h(t), \tag{14}$$

where: $h(t)$ is the Hilbert transform factor.

Complex analytic signal as follows is constructed:

$$z(t) = x(t) + iR(t) = A e^{i\phi(t)}, \tag{15}$$

where, $A(t)$ is the amplitude function:

$$A(t) = \sqrt{E_i^2(t) + R^2(t)}, \tag{16}$$

and, $\phi(t)$ is the phase function:

$$\phi(t) = \arctan \frac{R(t)}{E_i(t)}. \tag{17}$$

The instantaneous frequency of E_i is given by (18):

$$f(t) = \frac{d\phi(t)}{dt}. \tag{18}$$

The base-atom is obtained with the time information gained by the correlation detection and localization algorithm and the time–frequency parameter information obtained by Hilbert transform as the reference. And the redundant dictionary of this feature atom is constructed by performing equal-step discrete expansion of the time–frequency parameters on both sides of the reference values.

$$D = \left\{ \begin{matrix} g_1^1, g_2^1, \dots, g_M^1 \\ g_1^2, g_2^2, \dots, g_M^2 \\ \vdots \\ g_1^N, g_2^N, \dots, g_M^N \end{matrix} \right\}, \tag{19}$$

where, $D(:, j) = \{g_j^i | i = 1, 2, \dots, N\}$, denoting the set of atoms consisting of extensions of the characteristic base-atom g_i^0 . Atoms has the same length N as x .

BatOMP improved sparse decomposition algorithm. The optimization-seeking process can be viewed as a global optimization problem. In order to solve the problems of large computation and low efficiency of existing matching tracking algorithms, the adaptive matching tracking algorithm, BatOMP, with fast convergence and accurate approximation is studied by combining BA into the OMP algorithm.

For BatOMP, the bat individual positions P_i represent the atoms column index in the redundant dictionary D , thus: $g_i = D(:, P_i)$. And for noise-containing signal extraction, the fitness function of the traditional sparse decomposition is improved to take the ratio of the ℓ_2 norm of the residual and the inner product as the fitness function. The target signal tends to be regular signals and most random noises obeys Gaussian distributions with zero mean error. So, the ℓ_2 norm of the former is greater than the latter. In consequence, the smaller the fitness indicates that the residual sequence

contains smaller effective signal components and higher signal-to-noise separation. In addition, the larger the inner product, the better matches between the atom and the residual. So, the optimal individual bat position P_b is determined and saved according to (20).

$$Fit^n(P_{best}^n) = Fit_n(g_{best}^n) = \arg \min_{N_{pop}} \left(\frac{\|\xi_i^{n+1}\|_2}{\langle \xi_i, g_i \rangle} \right), \tag{20}$$

and

$$\xi_i^{n+1} = \xi_i^n - A^T(A^T \times A)^{-1} \times A^T * \xi_i^n, A = [g_1, g_2, \dots, g_n], \tag{21}$$

where A is the matched dictionary, composed by the selected best matching atoms.

The flow chart of BatOMP is as follows:

Input: x, f_s

Step 1: Redundant dictionary construction

Determined time-frequency parameters of x by dictionary training: $pare^0 = [f_0 \phi_0 d_0 t_0 \tau_1 \tau_2]$, and construct the redundant dictionary $D[N * M]$, where $M \gg N$ (N is the length of x)

Step 2: Algorithm initialization:

Defined the maximum iterations K and optimal threshold σ

The maximum iterations K is usually set to 50~100. If the optimal threshold is reached, the loop will end prematurely and the actual iteration will be smaller than K . The optimal threshold σ is determined by the ℓ_2 norm of the background noise ns (the original segment without the target signal): $\sigma = \lambda * \|ns\|_2$, generally $\lambda \in [1.5 \ 2]$. If the ℓ_2 norm of the residual never reaches σ , the loop will end until the maximum iterations is reached. So, K should not be too large.

Step 3: Circulative iteration and optimization

$k=1; \zeta_k=x, A=\emptyset;$

Start the outer loop of BatOMP:

while ($k < K$ && $\|\zeta_k\|_2 \geq \sigma$)

1) initialize BA:

$$N_{pop} = 300, N_{gen} = 30, f_i^0 = 0.8, r_i^0 = 0.6, A_i^0 = 0.8, n = 0; \\ P^0 = \text{randperm}(M, N_{pop})$$

Where, $\text{randperm}()$ is random permutation, which returns a row vector containing N_{pop} unique integers selected randomly from 1 to M .

2) Calculate the fitness

$$c(i) = \langle g_i, \zeta_n \rangle = \langle D(:, P_i), \zeta_k \rangle \\ \zeta_{k+1}^i = \zeta_k - c(i) * g_i \\ Fit(i) = \|\zeta_{k+1}^i\|_2 / |c(i)|, Fit_{best} = Fit(g_{best}^0) = \arg \min_{N_{pop}} (Fit(i)) \tag{22}$$

3) Select an optimum matching atom g_{best} by the inner loop (BA)

while ($n++ \leq N_{gen}$ && $\|\zeta_{k+1}\|_2 \geq \sigma$)

Search the best matching atom g_{best}^n by BA and calculated the optimal fitness value. Fit_{best}^n

end

4) *if* ($Fit_{best}^n < Fit_{best}$)

update the matched dictionary and residual by OMP

$$A = A \cup g_{best} = A \cup g_{best}^n \\ \theta = (A^T * A)^{-1} * A^T * \zeta_k \\ \zeta_{k+1} = \zeta_k - A * \theta \tag{23}$$

end

5) $k=k+1;$

End the outer loop of BatOMP.

Output: $\hat{s} = x - \zeta_k$ (24)

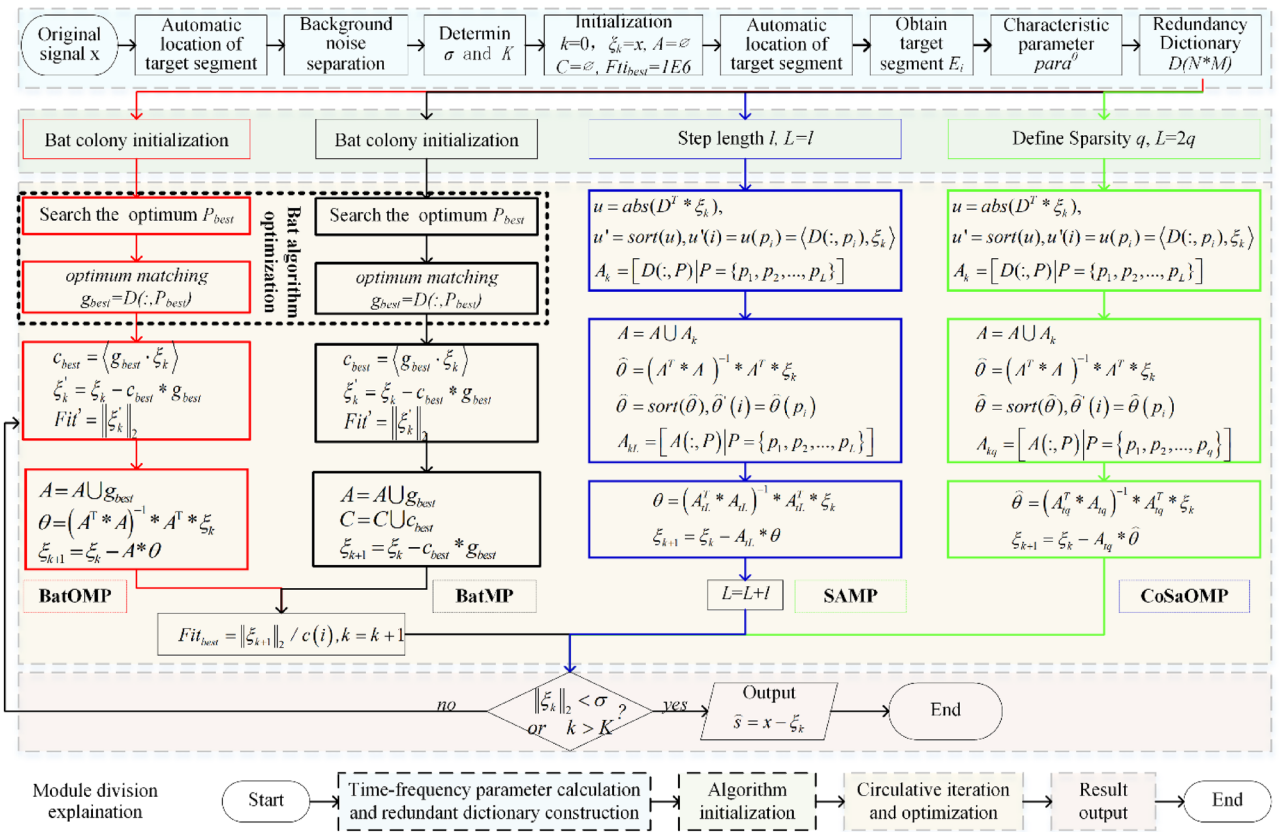


Figure 5. The overall flow chart of signal extraction algorithms.

The overall flow chart of the proposed signal extraction algorithm is shown in Fig. 5. The flow of sparse decomposition based on BA and MP (BatMP), CoSaOMP and SAMP are also presented for comparison. Different algorithms are distinguished by different border colors. All of the four methods consist of four main modules: 1. redundant dictionary construction, 2. algorithm initialization, 3. circulative iteration and optimization, and 4. result output. In this paper, module 1 and 4 are almost same for the different methods, module 2 is slightly different, and the differences are mainly reflected in module 3.

Experiments. We constructed a nonstationary signal x to test the methods described above:

$$x = ns + s = \begin{cases} ns + s_1, & t_{1s} < t < t_{1e} \\ ns + s_2, & t_{2s} < t < t_{2e} \\ ns, & \text{otherwise} \end{cases}$$

$$= \begin{cases} wgn + 20 \times \exp\left(-\frac{5e-4}{\sqrt{1-0.0005^2}} \times (2\pi \times 90 \times (t - 0.156))^2\right) \times \cos(2\pi \times 90 \times (t - 0.156)), & 0 < t < 1 \\ wgn + 180 \times \|\exp(-15 \times t) \times \sin(2\pi \times 30 \times t + \pi/2)\|, & 0.5 < t < 0.67 \\ wgn, & \text{otherwise} \end{cases} \quad (25)$$

where s is nonstationary target signal including pulse signal s_1 and partial discharge signal s_2 distributed in different regions, and ns is background noise subjecting to Gauss distribution. The sampling rate $f_s = 1500$ Hz, sampling time $T = 1$ s, and SNR is 7.402 dB. Thus, the sequence length N is 1500.

The PC for the testing features a i7-8550U CPU Core(TM) @ 1.80 GHz with 16.0 GB RAM, 4 cores and 8 Logic processors, running the 64 bit operating system.

Time–frequency parameter calculation. First, time and frequency analysis was performed, and results were shown in Fig. 6. From Fig. 6, after 0.4 s, s_1 decreases to zeros with the action of attenuation term.

Secondly, time information of the target signal was calculated by the correlation detection and localization algorithm described above, shown in Fig. 7. L_1 and L_2 are the calculated start and end indexes of the target segments, and L is the length of the segmentations. Accordingly, $t_1 = L_1/fs \approx 0$ s, $t_2 = L_2/fs \approx 0.415$ s for \hat{s}_1 , and $t_1 = L_1/fs \approx 0.501$ s, $t_2 = L_2/fs \approx 0.671$ s for \hat{s}_2 .

Different dictionaries construction and testing. After determining the key parameters of s , the redundant dictionary GT consisting of the two new atoms was created by dictionary training algorithm. And two

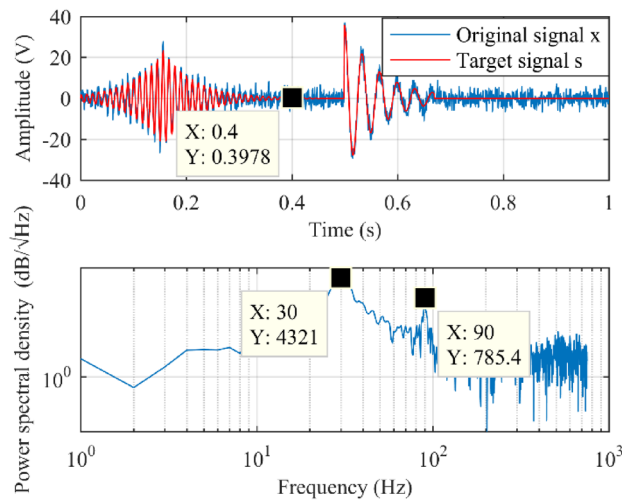


Figure 6. Waveform of the testing signal. The upper and lower subplot denote the time domain and frequency domain waveforms, respectively; red and blue lines in the upper subplot denote the original signal x and the target signal $s = s_1 + s_2$, respectively.

typical redundant dictionaries were built by Discrete Cosine Transformation (DCT) and Gabor dictionary for comparison. The dictionaries' sizes are shown in Table 2.

Since the OMP method will iterate over the whole dictionary repeatedly, the extraction results are relatively accurate. In view of this, we take the results of OMP from statistical analyses to illustrate the performance of the different dictionaries, as shown in Fig. 8. The vertical axis represents the deviation between the extracted signal \hat{s} and actual signal s :

$$Amp_{err} = s - \hat{s}. \quad (26)$$

We quantified the errors by RMS, the root mean square value of Amp_{err} . The time–frequency parameters were not considered when building the DCT dictionary by MATLAB, so there was the maximum deviation in the correlative results. The atom expression of Gabor dictionary is as follows:

$$gabor(i) = \frac{1}{\sqrt{a}} \times \exp \left(-\pi \times \left(\frac{i - \Delta t}{z} \right)^2 \times \cos(2\pi f(i - \Delta t) + \phi) \right). \quad (27)$$

We can see the lack of unilateral oscillation atoms by contrast with g -atom and t_r -atom. Relatively, GT dictionary is completer and more accurate, so OMP based on GT dictionary gave minimum errors and the shortest optimizing time.

Algorithm performance testing. The algorithms involved in the article including MP, OMP, SAMP, CoSaMP, BatMP and the proposed BatOMP were carried out for performance comparison.

The extraction results and corresponding errors obtained by different methods were shown in Fig. 9. Figure 9a,c,e display the extracted signals \hat{s} by different methods based on DCT, Gabor and GT dictionary respectively. And Fig. 9b,d,f present corresponding errors obtained by Eq. (26).

The efficiency analysis of different algorithms and dictionaries are shown in Fig. 10 and Table 3.

Because that the MP and OMP method traversed through the whole dictionary, the accuracies are relatively high and the latter is superior to the former.

The step length l of SAMP and sparsity q of CoSaMP were determined by expert experience. Because a certain amount atoms have to choose every time, there are clearly overextraction for these two methods. It's important to note that better parameters may be obtained by trial and error, but it is not suitable for real-time data processing.

For BatOMP, the best match atoms are determined by bat colony optimization. Every time before the searching, the bat individuals randomly scattered over the whole dictionary, and then gradually gather to the optimum solution through local optimization and global optimization. The optimal trajectories of ten bat individuals were randomly selected to show the convergence process, as shown in Fig. 11. The optimum solution is the index of the optimum matching atom in the redundant dictionary.

The difference between BatMP and BatOMP is similar to MP and OMP. BatOMP based on GT dictionary occupied the highest precision, probably because that MP and OMP took the inner product as the fitness function which leading to suboptimum for signal extraction. So, the results reflect the availability of the new fitness in some extent.

Moreover, the first four methods executed vast and complex matrix computations many times during the optimizing period, so they are time-consuming and require very high CPU occupancy rate compared with BatOMP.

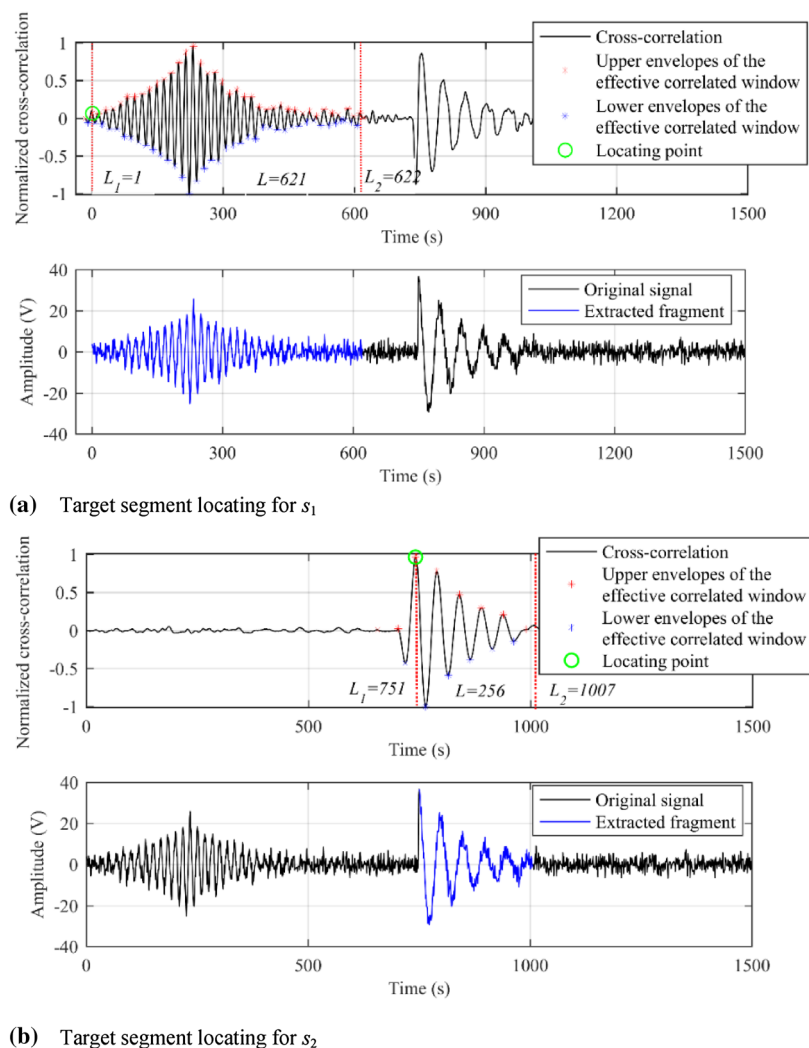


Figure 7. Target segment extraction based on correlation detection technique. The upper and lower subplot denoted the segmentation result of signal components s_1 and s_2 , respectively. In the upper subplots, the black solid lines are cross-correlation functions; red and blue scatters describe the envelopes of the effective correlated windows; and red dotted line are calculated time range. The green cycles indicate the start–end of the target segments.

Dictionary	Size
Gabor	1500 × 863,952
DCT	1444 × 1,048,576
GT	1500 × 593,190

Table 2. The dictionary sizes. When built the DCT dictionary, perfect square was required for the column length, so the nearest perfect square 1444 around N was selected.

In other words, BatOMP can be widely used even on low lever machines. This is important for the occasions without algorithmic workstation and high-performance computer, i.e. field data processing or low cost testing.

Conclusion

For nonstationary signal extraction, the dictionary training algorithm based on feature parameters is firstly used to determine the key parameter range of feature atoms, which can effectively reduce the redundancy while ensuring the completeness of the redundant dictionary; the bat algorithm combined with OMP is proposed to transform the signal sparse decomposition problem into an optimization problem with ratio of the ℓ_2 norm of the residual and the inner product as the fitness, which can improve the efficiency of the sparse

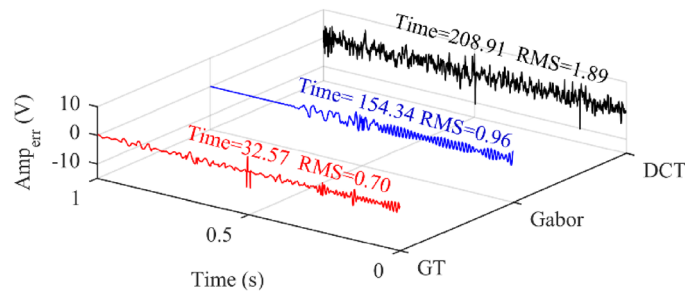


Figure 8. Dictionary testing.

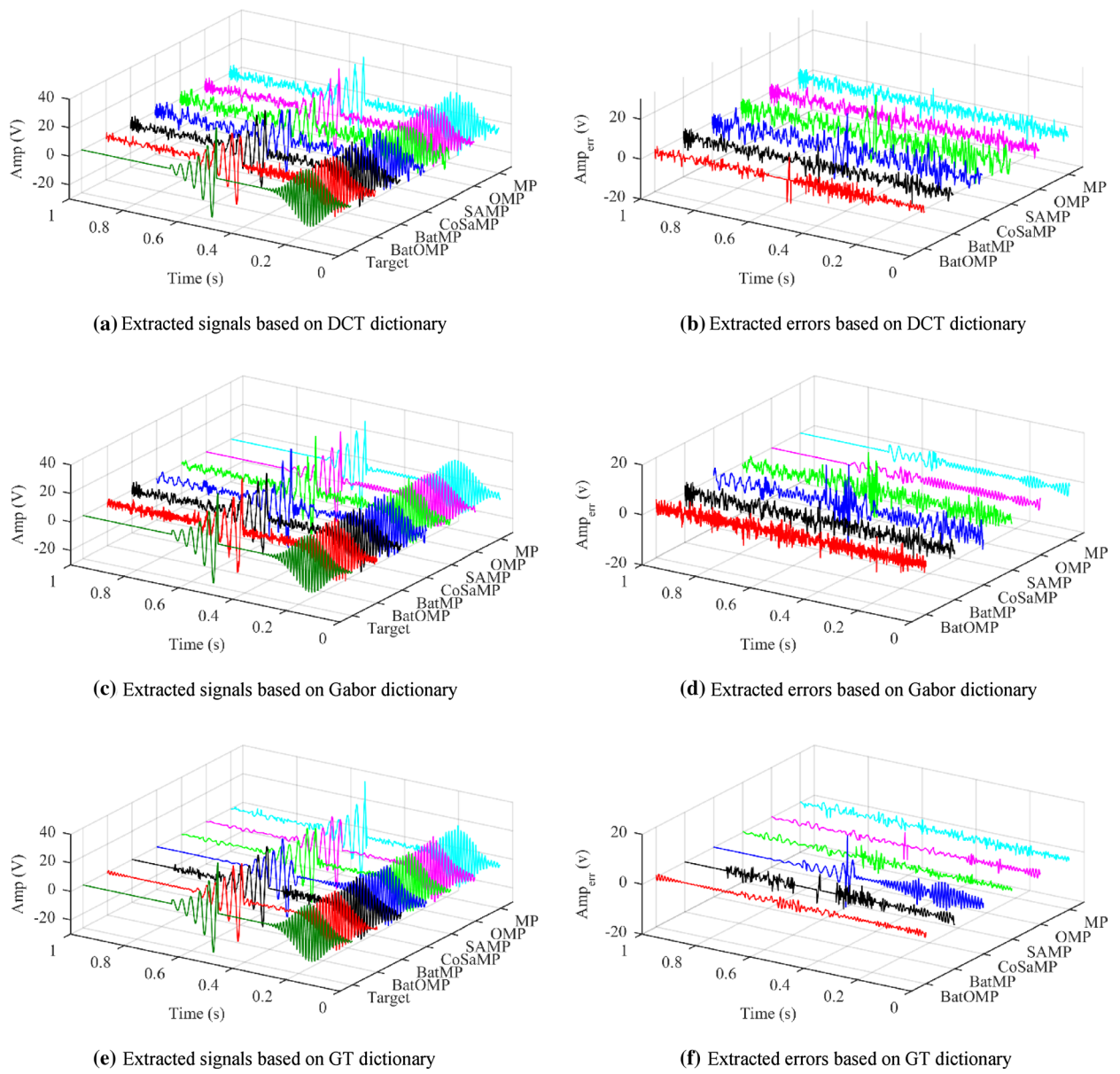


Figure 9. Signal extraction results. Colors: jasper: the target signal s ; red: result of BatOMP; black: result of BatMP; blue: result of CoSaMP; green: result of SAMP; pink: result of OMP; skyblue: result of MP.

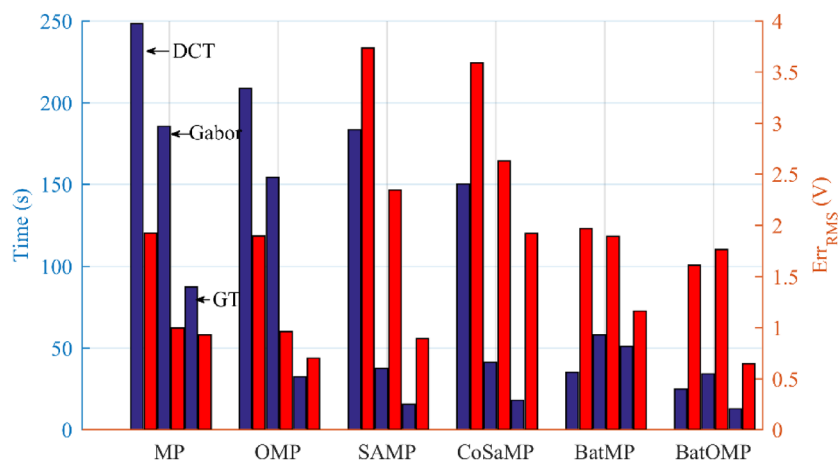


Figure 10. Quantitative analysis of experimental results. Blue bars mark the running time (the left axis) and red bars indicate the extraction errors (the right axis). For each method, the three bars of a sort from left to right represent the result of DCT, Gabor and GT dictionary respectively.

Algorithm	Elapsed time (s)			Square error (V)			CPU occupancy rate (%)
	DCT	Gabor	GT	DCT	Gabor	GT	
MP	248.45	185.62	87.38	1.92	1.00	0.93	58
OMP	208.91	154.34	32.57	1.89	0.96	0.70	57
SAMP	183.6	37.70	15.51	3.74	2.34	0.89	70
CoSaMP	150.11	41.41	18.01	3.59	2.63	1.92	55
BatMP	35.26	58.19	51.16	1.97	1.89	1.16	26
BatOMP	25.09	34.35	12.77	1.61	1.77	0.65	26

Table 3. Quantitative analysis of experimental results.

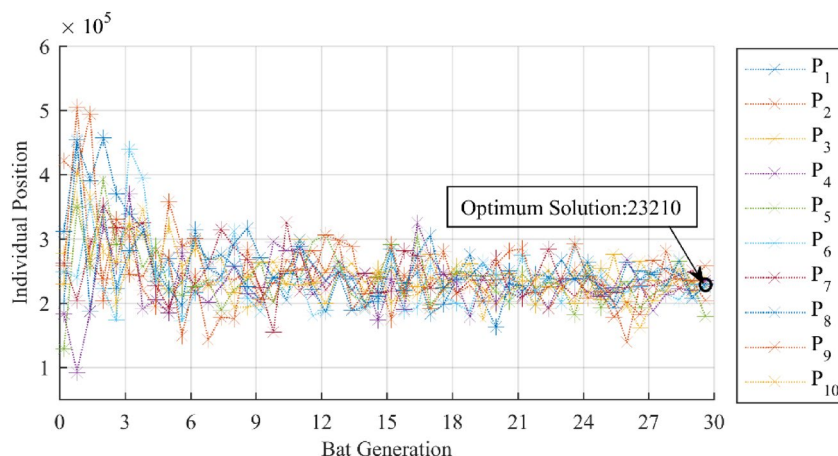


Figure 11. A complete search process of the bat colony. Obviously, the bats gradually converge to the best solution from the original scattered position.

decomposition algorithm. The experimental results showed that compared with other methods, the BatOMP algorithm is occupied with high efficiency, which can extract nonstationary signals form noise background without over constrained prior knowledges and avoid excessive decomposition. Testing results show that the proposed algorithm outperforms previous method in speeding up the convergence procedure and meanwhile ensuring high accuracy. Compared with the existing sparse decomposition algorithm, BatOMP requires much lower levels of hardware configuration. So, the new method will be helpful for to reducing data processing cost and enlarging the application fields.

Received: 15 April 2021; Accepted: 25 August 2021

Published online: 09 September 2021

References

1. Bohra, P. & Unser, M. Continuous-domain signal reconstruction using L_p -norm regularization. *IEEE Trans. Signal Process.* **69**, 1–1 (2020).
2. Gao, Y., Zhao, P., Li, G. & Li, H. Seismic noise attenuation by signal reconstruction: An unsupervised machine learning approach. *Geophys. Prospect.* **69**, 1–50 (2021).
3. Mack, W. & Habets, E. A. P. Deep filtering: Signal extraction and reconstruction using complex time-frequency filters. *IEEE Signal Process. Lett.* **27**, 61–65. <https://doi.org/10.1109/LSP.2019.2955818> (2020).
4. Li, X.-L., Wang, N., Gao, D.-Z. & Li, Q. A sound field separation and reconstruction technique based on reciprocity theorem and fourier transform. *Chin. Phys. Lett.* **35**, 114301. <https://doi.org/10.1088/0256-307x/35/11/114301> (2018).
5. Wang, X.-L. & Wang, W.-B. Harmonic signal extraction from noisy chaotic interference based on synchrosqueezed wavelet transform. *Chin. Phys. B* **24**, 080203. <https://doi.org/10.1088/1674-1056/24/8/080203> (2015).
6. Wolf, G., Mallat, S. & Shamma, S. Rigid motion model for audio source separation. *IEEE Trans. Signal Process.* **64**, 1822–1831 (2016).
7. Ukte, A., Kizilkaya, A. & Elbi, M. D. Two empirical methods for improving the performance of statistical multirate high-resolution signal reconstruction. *Dig. Signal Process.* **26**, 36–49 (2014).
8. Dragomiretskiy, K. & Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Process.* **62**, 531–544 (2014).
9. Mallat, S. G. & Zhifeng, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**, 3397–3415. <https://doi.org/10.1109/78.258082> (1993).
10. Mailhé, B. & Gribonval, R. A low complexity Orthogonal Matching Pursuit for sparse signal approximation with shift-invariant dictionaries. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*.
11. Baokang, Y., Bin, W., Fengxing, Z., Weigang, L. & Bo, X. Sparse decomposition method based on time-frequency spectrum segmentation for fault signals in rotating machinery. *ISA Trans.* **83**, 142–153 (2018).
12. Needell, D. & Vershynin, R. Signal recovery from inaccurate and incomplete measurements via regularized orthogonal matching pursuit. *IEEE J. Select. Top. Signal Process.* **4**(2), 310–316 (2010).
13. Needell, D. & Tropp, J. A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM* **53**, 93 (2010).
14. Sun, G., Zhou, Y., Wang, Z., Dang, W. & Li, Z. Sparsity adaptive compressive sampling matching pursuit algorithm based on compressive sensing. *J. Comput. Inf. Syst.* **8**, 2883–2890 (2012).
15. Zhu, J. & Xiaolu, L. Electrocardiograph signal denoising based on sparse decomposition. *Healthcare Technol. Lett.* **4**, 134–137 (2017).
16. Clerc, M. *Particle Swarm Optimization* (Fractional Order Darwinian Particle Swarm Optimization, 2016).
17. Fei, W., Xu, X. & Jing, Z. J. Single airport ground-holding problem optimizing strategy based on artificial fish school algorithm. *J. Nanjing Univ. Aeronaut. Astronaut.* **41**, 116–120 (2009).
18. Civicioglu, P. & Besdok, E. J. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif. Intell. Rev.* **39**, 315–346 (2013).

Acknowledgements

This research was sponsored by National Natural Science Foundation of China (41904080), the Key Research and Development (R&D) Projects of Shanxi Province (201903D121118) and Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi.

Author contributions

S.C.G. wrote the main manuscript text, and S.D.Z. participated in the experiment. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021