

Research Article

Blind Source Separation Based on Quantum Slime Mould Algorithm in Impulse Noise

Zhiwei Zhang , Hongyuan Gao , Jingya Ma , Shihao Wang , and Helin Sun 

College of Information and Communication Engineering, Harbin Engineering University, No. 145 Nantong Street, Nangang District, Harbin 150001, China

Correspondence should be addressed to Hongyuan Gao; gaohongyuan@hrbeu.edu.cn

Received 13 May 2021; Revised 19 June 2021; Accepted 13 July 2021; Published 29 July 2021

Academic Editor: Rohit Salgotra

Copyright © 2021 Zhiwei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to resolve engineering problems that the performance of the traditional blind source separation (BSS) methods deteriorates or even becomes invalid when the unknown source signals are interfered by impulse noise with a low signal-to-noise ratio (SNR), a more effective and robust BSS method is proposed. Based on dual-parameter variable tailing (DPVT) transformation function, moving average filtering (MAF), and median filtering (MF), a filtering system that can achieve noise suppression in an impulse noise environment is proposed, noted as MAF-DPVT-MF. A hybrid optimization objective function is designed based on the two independence criteria to achieve more effective and robust BSS. Meanwhile, combining quantum computation theory with slime mould algorithm (SMA), quantum slime mould algorithm (QSMA) is proposed and QSMA is used to solve the hybrid optimization objective function. The proposed method is called BSS based on QSMA (QSMA-BSS). The simulation results show that QSMA-BSS is superior to the traditional methods. Compared with previous BSS methods, QSMA-BSS has a wider applications range, more stable performance, and higher precision.

1. Introduction

BSS is a signal processing method that extracts or restores each component of the source signal only through the received observation signals without any prior knowledge of the source signals and the transmission channels. BSS has gradually become a research hotspot and has been successfully applied in various fields, such as image and voice signal processing, biomedical signal analysis and processing, or antenna array signal processing [1–3]. However, the current BSS methods still have many shortcomings.

Firstly, when these traditional BSS methods are implemented, they involve the selection of nonlinear functions for separation operations according to the probability density properties and kurtosis values of the source signal, which contradicts the unknown nature of the source signal and channel. Moreover, these algorithms are difficult to jump out of the local optimum, and the convergence speed is slow, which affects the separation effect [4–6]. The swarm

intelligent algorithms do not require any source signal or channel information as prior knowledge and do not need to select nonlinear functions. Moreover, the swarm intelligence algorithms have faster convergence speed, and it is easier to jump out of the local optimum and achieve higher accuracy. Although the swarm intelligent algorithms, such as particle swarm algorithm (PSO) [7], genetic algorithm (GA) [8], bacterial foraging algorithm (BFA) [9], slime mould algorithm (SAM) [10], artificial bee colony (ABC) algorithm [11], whale optimization algorithm (WOA) [12], grey wolf optimization (GWO) algorithm [13], crow search algorithm (CSA) [14], and bat algorithm (BA) [15], make up the deficiency of the traditional methods to some extent, the swarm intelligent algorithms containing many parameters need to be improved in terms of convergence speed, convergence accuracy, and stability. Improper parameter adjustment can easily degrade performance. In 2020, a new intelligent algorithm named slime mould algorithm (SMA) [10] is proposed, and this algorithm has fewer parameters setting and lower complexity.

In addition, the classical BSS algorithms often design the objective function based on a single independence criterion such as maximum kurtosis (MK), maximum negative entropy (MNE), minimum mutual information, or maximum likelihood criterion of the separated signals [16, 17]. However, using a single independence criterion may cause that the selected criterion is not the best criterion of the algorithm, which leads to the inability to obtain the best result. The best criterion for different algorithms may also be different. Shih-Hsiung and Yang [18] proposed a new algorithm named independent component analysis based on gravitational particle swarm optimization (GPSO-ICA), and the performance of this algorithm based on three different criteria is compared. Ebrahimzadeh and Mavaddati [19] compared the performance of the artificial bee colony (ABC) algorithm based on the different efficient cost functions. Kumar and Jayanthi [20] compared the performance of the three different independence criteria for the fast independent component analysis (FAST-ICA).

In recent years, the BSS in impulse noise has made great progress [21–23]. However, on the one hand, these traditional BSS methods in impulse noise are often only suitable for weak impulse noise and high SNR. However, under the strong impulse noise or the low SNR, the performance of these methods deteriorates or even becomes invalid. On the other hand, most of the existing research rarely applies filtering methods to BSS under impulse noise because traditional impulse noise suppression methods often require information from the source signals or noise as prior knowledge in order to achieve good performance. A dual-parameter variable tailing (DPVT) transformation function based on exponential function is designed by Luo et al. [24], which needs to adjust the threshold according to source signals and noise situation. Arce and Gonzalez [25] proposed the weighted myriad filtering method that needs the ideal signals as prior knowledge. A minimum mean square error (MMSE) without any prior knowledge is proposed by Moon and Weissman [26], but the performance of this method will deteriorate in impulse noise. A transformation function named Gaussian-tailed zero-memory nonlinearity (GZMNL) is optimized by Luo et al. [27], and this method still requires the standard deviation of the dispersion coefficient as prior knowledge.

Aiming at the above shortcomings, the following are the main contributions of this paper:

- (1) Aiming at the above shortcomings of these existing algorithms, we design a new intelligent algorithm named QSMA based on quantum coding, quantum-simulated rotation gate, and slime mould algorithm (SMA) [10] to obtain the optimal solution. The quantum rotation angle is designed, and the simulated quantum rotation gate is introduced to update the quantum positions. And the simulated quantum rotation gate further improves the search accuracy. Due to the parallel nature of quantum computing, quantum evolutionary algorithms have many advantages, such as the small population size does not affect the performance of the algorithm; the number

of iterations is small, but the global optimization ability is strong, and the past historical information of the individual is effectively used in the evolution process [28, 29]. Compared with previous algorithms used to solve other engineering problems, QSMA achieves better convergence performance and fewer parameters setting.

- (2) A new hybrid optimization objective function based on two different independence criteria, MK and MNE, is designed. Corresponding weight coefficients are assigned to the two criteria, and the best criterion of the algorithm is judged according to the change of the performance evaluation index with the weight coefficient, so as to get a more accurate result.
- (3) In this paper, a filtering system named MAF-DPVT-MF is proposed, and we introduce this filtering system into the BSS model. Unlike the DPVT transformation function, MAF-DPVT-MF adjusts the threshold according to the observation signals after filtering. The combination with median filtering [30] method enhances the ability to suppress noise. MAF-DPVT-MF does not require any prior knowledge and can achieve BSS in impulse noise. Simulation results illustrate that this filtering system is not only suitable for weak impulse noise environment but also suitable for strong impulse noise and low SNR noise environment.

The rest of this paper is systemized as follows: The BSS model under source signals in impulse noise is presented in the next section. In Section 3, QSMA is proposed and applied in BSS; moreover, its convergence is further analyzed. Simulation results and conclusions are presented in Sections 4 and 5, respectively.

2. BSS Model under Source Signals in Impulse Noise

2.1. α -Stable Distribution. α Stable distribution is used for impulse noise modeling [31] and is usually defined by characteristic function, as shown in the following equation:

$$\varphi(t) = \begin{cases} \exp\left\{j\mu t - \gamma|t|^\alpha \left[1 + j\beta \operatorname{sgn}(t) \tan\left(\frac{\alpha\pi}{2}\right)\right]\right\}, & \alpha \neq 1, \\ \exp\left\{j\mu t - \gamma|t|^\alpha \left[1 + j\beta \operatorname{sgn}(t) \frac{2}{\pi} \log|t|\right]\right\}, & \alpha = 1, \end{cases} \quad (1)$$

where j denotes the imaginary unit. α ($0 < \alpha \leq 2$) denotes characteristic exponent, and the smaller the α is, the greater the impulse degree of distribution is; the larger α is, the smaller impulse degree of distribution is. β ($-1 \leq \beta \leq 1$) represents symmetry parameter, and α stable distribution is called standard SaS distribution when $\beta = 0$. When $\beta = 0, \alpha = 2$, the SaS distribution becomes Gaussian distribution, and the noise generated is Gaussian noise. When $\beta = 0, 1 \leq \alpha < 2$, the noise generated is weak impulse noise. When $\beta = 0, \alpha < 1$, the noise generated is strong impulse

noise. γ ($\gamma > 0$) denotes the dispersion of impulse noise and is also called scale parameter. μ ($-\infty < \mu < +\infty$) represents location parameter. sgn denotes the sign function. Since the SaS distribution does not have a second moment, the mixed signal-to-noise ratio (MSNR) is defined as follows:

$$\text{MSNR} = 10 \lg \left[\sum_{\bar{t}=1}^{\bar{N}} \frac{\bar{\varepsilon}_{\bar{t}}^2}{(\bar{N}\gamma)} \right], \quad (2)$$

where $\bar{\varepsilon}_{\bar{t}}^2$ represents the power of the \bar{t} th signal and $\bar{t} = 1, 2, \dots, \bar{N}$.

We can generate impulse noise obeying the standard SaS distribution ($0 < \alpha < 2, \beta = 0, \gamma = 1, \mu = 0$) through the

$$X = \begin{cases} S \times \frac{\sin(\alpha(U^* - B))}{(\cos U^*)^{(1/\alpha)}} \times \left(\frac{\cos(U^* - \alpha(U^* - B))}{W^*} \right)^{((1-\alpha)/\alpha)}, & \alpha \neq 1, \\ \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta U^* \right) U^* - \beta \log \left(\frac{W^* \cos U^*}{(\pi/2) + \beta U^*} \right) \right], & \alpha = 1, \end{cases} \quad (3)$$

where

$$B = \frac{\arctan(\beta \tan(\pi\alpha/2))}{\alpha}, \quad (3a)$$

$$S = \left(1 + \beta^2 \tan^2 \frac{\pi\alpha}{2} \right)^{(1/2\alpha)}. \quad (3b)$$

From the above process, the impulse noise of each sampling point is as follows:

$$n = \sqrt{\sum_{\bar{t}=1}^{\bar{N}} \frac{\bar{\varepsilon}_{\bar{t}}^2}{(10^{(\text{MSNR}/10)})}} \times X. \quad (4)$$

Figure 1 illustrates the discrete realization of the Gaussian process when $\alpha = 2$ and the standard SaS process when $\alpha = 1.9, \alpha = 1.5$, and $\alpha = 0.9$.

Figure 1 illustrates that as α decreases, the impulse of noise continues to increase. When $\beta = 0, \alpha = 2$, the standard SaS distribution becomes Gaussian distribution, and the noise generated is Gaussian noise. When $\beta = 0, \alpha = 1.9$, and $\beta = 0, \alpha = 1.5$, the SaS distribution is still the standard SaS distribution, and the noise generated is weak impulse noise. When $\beta = 0, \alpha = 0.9$, the SaS distribution is still the standard SaS distribution, and the noise generated is strong impulse noise. The MSNR is set as 10 dB.

2.2. Construction of the Filtering System

2.2.1. DPVT Transformation Function. The dual-parameter variable tailing (DPVT) [24] is a new nonlinear transformation function proposed in 2019. One parameter controls the linear region threshold, and the other parameter controls

algorithm obtained by the conversion formula in [32]. The specific content of this algorithm is as follows:

- Step 1 A random variable U^* obeying a uniform distribution on $[-(\pi/2), (\pi/2)]$ is generated.
- Step 2 Another exponentially distributed random variable W^* with a mean of 1 is generated.
- Step 3 A random variable X obeying the standard SaS distribution is obtained according to the following formula:

the tail decay speed. The nonlinear function with exponential function as the tail is designed as follows:

$$G(g, \Lambda, c) = \begin{cases} g, & |g| \leq \Lambda, \\ \Lambda c^{|g|-\Lambda} \cdot \text{sgn}(g), & |g| > \Lambda, \end{cases} \quad (5)$$

where $c \in (0, 1]$. By adjusting the value of c , the nonlinear function can achieve different degrees of suppression for large-value samples. Λ denotes the linear region threshold. g denotes the signal to be processed. However, since the noise situation is unknown in the BSS, improper threshold setting may cause the deterioration of performance. Consequently, we need to design a filter system that can adjust the threshold according to the noise situation.

2.2.2. MAF-DPVT-MF. $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_D(t)]^T$ are D independent source signals and $t = 1, 2, \dots, N$. t denotes the number of sampling points, and N denotes the total number of sampling points. $\mathbf{n}(t) = [n_1(t), n_2(t), \dots, n_D(t)]^T$ are impulse noises superimposed on $\mathbf{s}(t)$, $\bar{\mathbf{s}}(t) = \mathbf{s}(t) + \mathbf{n}(t)$. \mathbf{A} is $D \times D$ unknown hybrid system matrix. $\bar{\mathbf{x}}(t) = \mathbf{A}\bar{\mathbf{s}}(t) = [\bar{x}_1(t), \bar{x}_2(t), \dots, \bar{x}_D(t)]^T$ denote observation signals. Because of impulse noises' covering, we cannot directly see the peak of $\bar{\mathbf{x}}(t)$ to determine the Λ . Consequently, we have to use a moving average filter (MAF) [33] that can reduce the amplitude of impulse noise to obtain the $\hat{\mathbf{x}}(t) = [\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_D(t)]^T$.

By this way, we can determine the DPVT's threshold as follows:

$$\Lambda = \sum_{i=1}^D \max_{t=1,2,\dots,N} \{\hat{x}_i(t)\}. \quad (6)$$

Through a lot of simulation experiments, this threshold setting method is proved to be effective and stable. The observation signals after passing DPVT enter the median

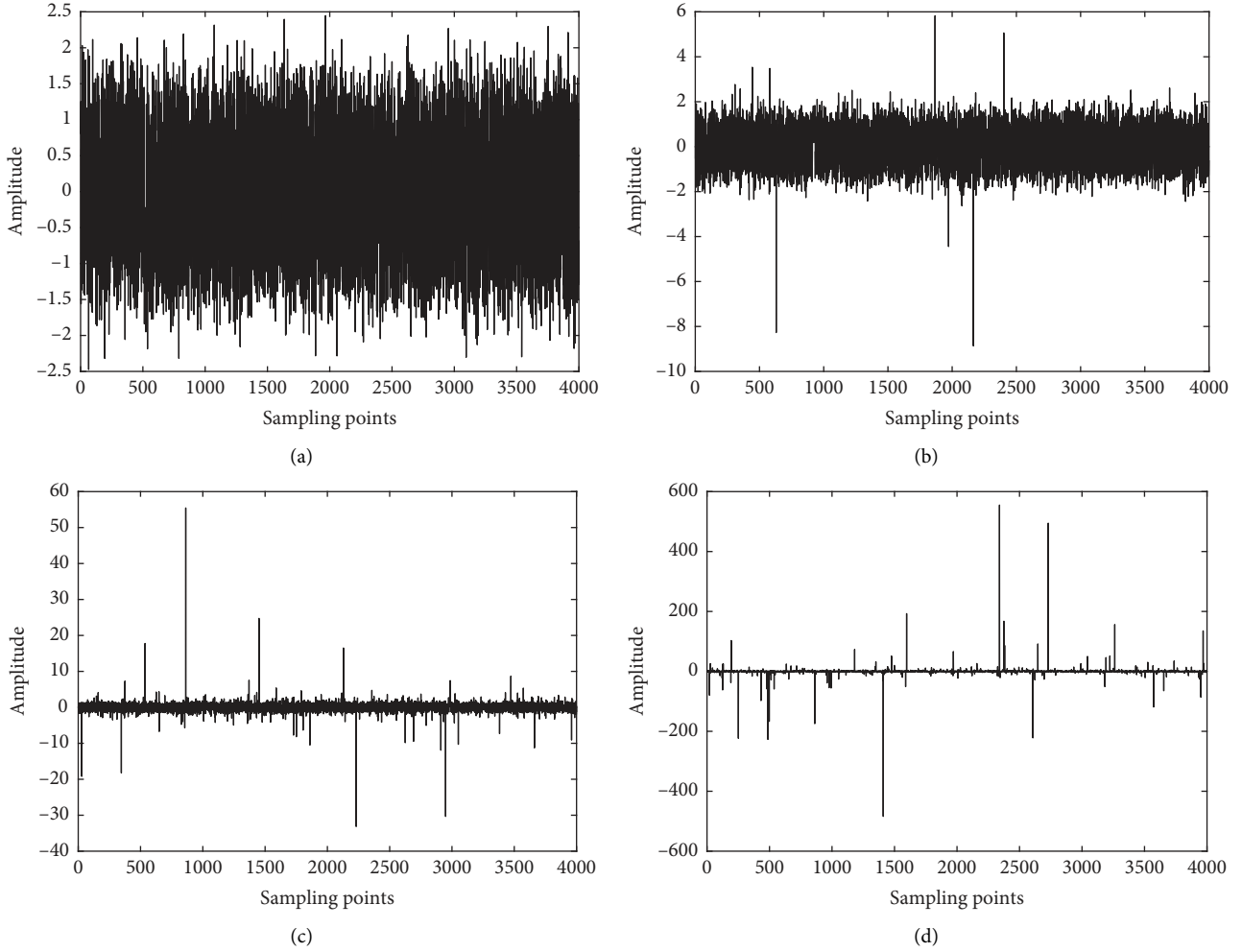


FIGURE 1: Discrete realization of the Gaussian process ($\alpha = 2$) and the standard SaS process ($\alpha = 1.9, 1.5, 0.9$): (a) $\alpha = 2$, (b) $\alpha = 1.9$, (c) $\alpha = 1.5$, and (d) $\alpha = 0.9$.

filter to further suppress the noise. Finally, we can acquire the filtered observation signals $\mathbf{x}(t)$.

2.2.3. Hybrid Optimization Objective Function. Preprocessing includes two steps that are centralization and whitening. The process of centralization is described as either $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - E(\mathbf{x}(t))$ or

$$\bar{x}_i(t) = x_i(t) - \frac{1}{N} \sum_{t=1}^N x_i(t), \quad (7)$$

where $i = 1, 2, \dots, D$, $t = 1, 2, \dots, N$, and E denotes mathematical expectation. We can acquire $\bar{\mathbf{x}}(t) = [\bar{x}_1(t), \bar{x}_2(t), \dots, \bar{x}_D(t)]^T$ that are zero-mean observation signals through centralization. Whitening $\bar{\mathbf{x}}(t)$, that is, performing a linear transformation on $\bar{\mathbf{x}}(t)$ obtains $\mathbf{z}(t) = \mathbf{V}\bar{\mathbf{x}}(t)$, $E(\mathbf{z}(t)\mathbf{z}(t)^T) = \mathbf{I}$. $\mathbf{z}(t)$ named whitened signal is $\mathbf{x}(t)$ preprocessed. \mathbf{V} is called whitening matrix, and \mathbf{I} denotes the identity matrix. The covariance matrix $\mathbf{R} = E[\bar{\mathbf{x}}(t)\bar{\mathbf{x}}(t)^T]$ is calculated according to $\bar{\mathbf{x}}(t)$ and

performs eigenvalue decomposition on \mathbf{R} to obtain $\mathbf{R} = \mathbf{Q}\mathbf{H}\mathbf{Q}^T$. The orthogonal matrix \mathbf{Q} is composed of eigenvectors of \mathbf{R} , and the diagonal matrix \mathbf{H} is composed of eigenvalues corresponding to the eigenvectors. In summary, we can get the whitening matrix, that is, $\mathbf{V} = \mathbf{Q}\mathbf{H}^{-1/2}\mathbf{Q}^T$ and $E(\mathbf{z}(t)\mathbf{z}(t)^T) = E(\mathbf{V}\bar{\mathbf{x}}(t), \bar{\mathbf{x}}(t)^T\mathbf{V}^T) = \mathbf{Q}\mathbf{H}^{-1/2}\mathbf{Q}^T\mathbf{Q}\mathbf{H}\mathbf{Q}^T\mathbf{Q}\mathbf{H}^{-1/2}\mathbf{Q}^T = \mathbf{I}$. The objective function is optimized by the algorithm to obtain the separation matrix \mathbf{W} . $\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t) = [y_1(t), y_2(t), \dots, y_D(t)]^T$ are both the separated signals and the estimated signals of $\mathbf{s}(t)$. Since the characteristics of the source signals and the transmission channel are unknown, $\mathbf{y}(t)$ has randomness in amplitude and arrangement order, which is called the ambiguity of BSS. $\mathbf{y}(t)$ can also be further processed by the median filter to obtain $\hat{\mathbf{y}}(t)$ (Figure 2). Preprocessing is also required for $\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t)$, and $\bar{\mathbf{y}}(t) = \mathbf{W}\bar{\mathbf{z}}(t) = [\bar{y}_1(t), \bar{y}_2(t), \dots, \bar{y}_D(t)]^T$ can be acquired after preprocessing. \mathbf{W} is the separation matrix corresponding to $\bar{\mathbf{y}}(t)$. $\bar{\mathbf{z}}(t)$ is the whitened signal corresponding to $\bar{\mathbf{y}}(t)$. The absolute value of kurtosis is calculated by

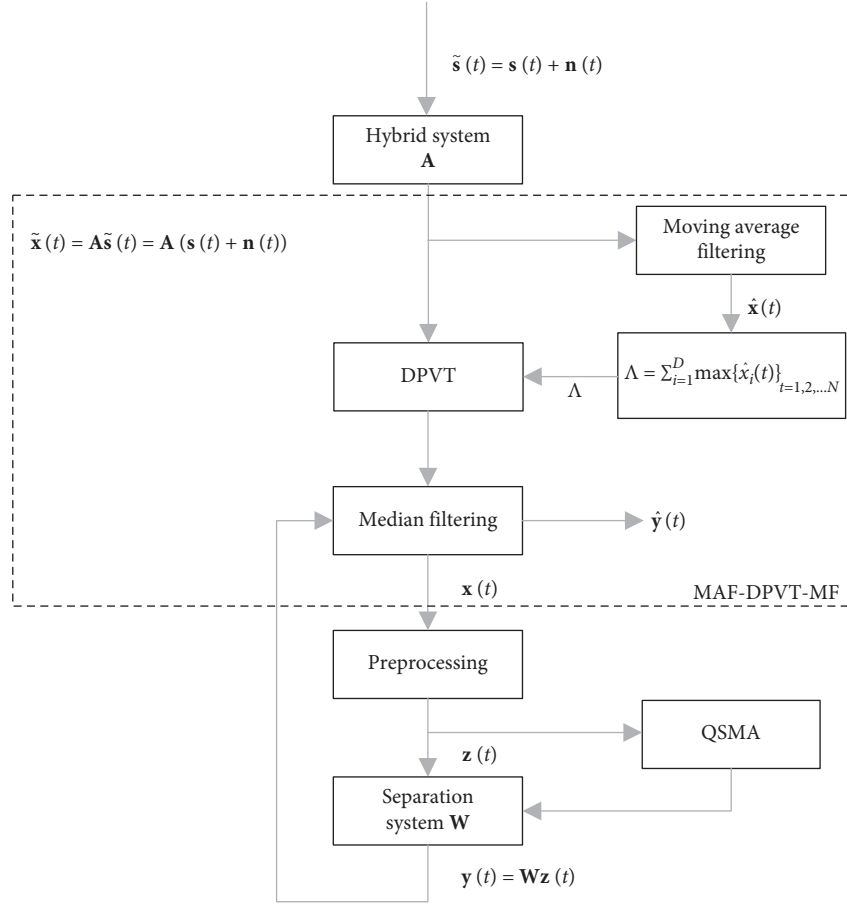


FIGURE 2: BSS model under source signals in impulse noise.

$$\begin{aligned}
 K(\bar{\mathbf{W}}) &= K^*(\bar{\mathbf{y}}(t)) \\
 &= K^*(\bar{\mathbf{W}}\bar{\mathbf{z}}(t)) \\
 &= \sum_{i=1}^D |K^*(\bar{y}_i(t))| \\
 &= \sum_{i=1}^D |E[\bar{y}_i^4(t)] - 3(E^2(\bar{y}_i^2(t)))|,
 \end{aligned} \tag{8}$$

$$E[\bar{y}_i^4(t)] = \frac{1}{N} \sum_{t=1}^N \bar{y}_i^4(t), \tag{8a}$$

$$E[\bar{y}_i^2(t)] = \frac{1}{N} \sum_{t=1}^N \bar{y}_i^2(t). \tag{8b}$$

And the negative entropy is calculated by

$$J(\bar{\mathbf{W}}) = J^*(\bar{\mathbf{y}}(t)) = J^*(\bar{\mathbf{W}}\bar{\mathbf{z}}(t)) = \sum_{i=1}^D \frac{1}{N} K^{*2}(\bar{y}_i(t)), \tag{9}$$

where χ denotes negative entropy function coefficient. So hybrid optimization objective function can be denoted as follows:

$$C(\bar{\mathbf{W}}) = aK(\bar{\mathbf{W}}) + bJ(\bar{\mathbf{W}}), \tag{10}$$

where a and b are both numbers between $[0,1]$ and $a + b = 1$. $K^*(\bar{\mathbf{y}}(t))$ and $J^*(\bar{\mathbf{y}}(t))$ are the calculation formula for the absolute value of kurtosis and the negative entropy with $\bar{\mathbf{y}}(t)$ as the independent variable, respectively. $K(\bar{\mathbf{W}})$ and $J(\bar{\mathbf{W}})$ are the calculation formula for the absolute value of kurtosis and the negative entropy with $\bar{\mathbf{W}}$ as the independent variables, respectively.

Figure 2 shows the BSS model after introducing MAF-DPVT-MF, which is also BSS model under source signals in impulse noise (Figure 2). Figure 2 describes the following process:

- Step 1 The source signals $\tilde{\mathbf{s}}(t) = \mathbf{s}(t) + \mathbf{n}(t)$ suppressed by impulse noise enter the system to obtain the observation signals $\tilde{\mathbf{x}}(t)$.
- Step 2 $\tilde{\mathbf{x}}(t)$ enters the DPVT; meanwhile, it also enters the MAF to obtain $\hat{\mathbf{x}}(t)$, and the threshold Λ can be determined according to $\hat{\mathbf{x}}(t)$. This is the first parallel structure. After the signals are filtered by DPVT, $\mathbf{x}(t)$ can be obtained after entering the MF. MAF-DPVT-MF consists of the above process.
- Step 3 $\mathbf{z}(t)$ can be obtained after $\mathbf{x}(t)$ is preprocessed.

Step 4 The hybrid optimization objective function is established according to $\mathbf{z}(t)$, and QSMA is used to find the optimal solution of the objective function; consequently, the separation system \mathbf{W} is determined. After $\mathbf{z}(t)$ passing through \mathbf{W} , the final separation signals $\mathbf{y}(t)$ can be obtained. This is the second parallel structure.

Step 5 $\mathbf{y}(t)$ can also be further processed by the MF to obtain $\hat{\mathbf{y}}(t)$.

3. QSMA-BSS

3.1. The Proposed QSMA. Quantum slime mould algorithm (QSMA) is an intelligent novel optimization algorithm based on the oscillation mode of slime moulds in nature [10] and the quantum behavior metaphor of slime moulds. The

population of quantum slime moulds is \bar{M} , and the maximum number of iterations for the entire population is G^* . ε denotes iterations, and d denotes the spatial dimension of each quantum slime mould. The quantum position of the k th quantum slime mould is randomly initialize as $\bar{\mathbf{m}}_k^\varepsilon = [\bar{m}_{k1}^\varepsilon, \bar{m}_{k2}^\varepsilon, \dots, \bar{m}_{kd}^\varepsilon]$, $k = 1, 2, \dots, \bar{M}$, $0 \leq \bar{m}_{kr}^\varepsilon \leq 1$, and $r = 1, 2, \dots, d$. The quantum position of quantum slime mould is mapped to the position of quantum slime mould $\mathbf{m}_k^\varepsilon = [m_{k1}^\varepsilon, m_{k2}^\varepsilon, \dots, m_{kd}^\varepsilon]$. The mapping rule is $m_{kr}^\varepsilon = m_r^{\min} + \bar{m}_{kr}^\varepsilon (m_r^{\max} - m_r^{\min})$, where $m_r^{\min} \leq m_{kr}^\varepsilon \leq m_r^{\max}$. m_r^{\max} and m_r^{\min} are the upper and lower limit of the r -th dimensional variable for the quantum slime mould's position, respectively.

The k -th quantum slime mould's r -th dimensional quantum rotation angle is updated as follows:

$$\delta_{kr}^{\varepsilon+1} = \begin{cases} v_1 \times (w(\mathbf{m}_k^\varepsilon) \bar{m}_{kr}^\varepsilon - \bar{m}_{kr}^\varepsilon), & \bar{r}_k \geq \bar{p}_k, \\ v_2 \times \bar{m}_{kr}^\varepsilon + \hat{u}_{kr}^\varepsilon (\bar{m}_{\text{mean},r}^\varepsilon - \bar{m}_{kr}^\varepsilon) + \hat{u}_{kr}^\varepsilon (\bar{m}_{\text{worst},r}^\varepsilon - \bar{m}_{kr}^\varepsilon), & \bar{r}_k < \bar{p}_k, \end{cases} \quad (11)$$

where η and ξ represent two individuals' numbers selected from population and $r = 1, 2, \dots, d$. $w(\mathbf{m}_k^\varepsilon)$ represents the weight of the k -th quantum slime mould. \hat{u}_{kr}^ε and \hat{u}_{kr}^ε are both uniform random numbers between 0 and 1. The k -th quantum slime mould's quantum position will be selected to be updated with the discovery probability \bar{p}_k , for which a random number \bar{r}_k will be produced with uniform distribution among $[0, 1]$. $\bar{\mathbf{m}}_{\text{mean}}^\varepsilon = [\bar{m}_{\text{mean},1}^\varepsilon, \bar{m}_{\text{mean},2}^\varepsilon, \dots, \bar{m}_{\text{mean},d}^\varepsilon] = (1/\bar{M}) \sum_{k=1}^{\bar{M}} \bar{\mathbf{m}}_k^\varepsilon$ and $\bar{\mathbf{m}}_{\text{worst}}^\varepsilon = [\bar{m}_{\text{worst},1}^\varepsilon, \bar{m}_{\text{worst},2}^\varepsilon, \dots, \bar{m}_{\text{worst},d}^\varepsilon]$ denotes the worst quantum position. v_1 is called oscillated weight coefficient with a range of $[-\lambda, \lambda]$, and λ is represented as follows:

$$\lambda = \arctan h\left(-\left(\frac{\varepsilon}{G^*}\right) + 1\right). \quad (12)$$

v_2 is called the inertia weight coefficient and

$$v_2 = v_2^{\max} - (v_2^{\max} - v_2^{\min}) \times \frac{\varepsilon}{G^*}, \quad (13)$$

where v_2^{\max} and v_2^{\min} are the maximum value and the minimum value of v_2 , respectively.

And the k -th quantum slime mould's r -th dimensional quantum position is updated as follows:

$$\bar{m}_{kr}^{\varepsilon+1} = \begin{cases} \left| \bar{m}_{\text{best},r}^\varepsilon \cos(\delta_{kr}^{\varepsilon+1}) - \sqrt{1 - (\bar{m}_{\text{best},r}^\varepsilon)^2} \sin(\delta_{kr}^{\varepsilon+1}) \right|, & \bar{r}_k \geq \bar{p}_k, \\ \left| \bar{m}_{kr}^\varepsilon \cos(\delta_{kr}^{\varepsilon+1}) - \sqrt{1 - (\bar{m}_{kr}^\varepsilon)^2} \sin(\delta_{kr}^{\varepsilon+1}) \right|, & \bar{r}_k < \bar{p}_k, \end{cases} \quad (14)$$

where $\bar{\mathbf{m}}_{\text{best}}^{\varepsilon+1} = [\bar{m}_{\text{best},1}^{\varepsilon+1}, \bar{m}_{\text{best},2}^{\varepsilon+1}, \dots, \bar{m}_{\text{best},d}^{\varepsilon+1}]$ denotes the global optimal quantum position.

QSMA can be used not only in BSS problems but also in other optimization problems.

3.2. QSMA-BSS. The separation matrix \mathbf{W} is an orthogonal matrix, which can be expressed as the product of a series of rotation matrices using the Givens rotation transformation [34] to reduce the amount of calculation. The expression is as follows:

$$\mathbf{W}(\boldsymbol{\theta}) = \left(\prod_{i=1}^{D-1} \mathbf{T}_{(D-i)D} \right) \left(\prod_{i=1}^{D-2} \mathbf{T}_{(D-1-i)(D-1)} \right), \dots, \mathbf{T}_{12}, \quad (15)$$

$$\mathbf{T}_{\psi l} = \begin{bmatrix} \mathbf{I}_{(\psi-1),(\psi-1)} & 0 & \cdots & 0 \\ 0 & \Delta & & \vdots \\ \vdots & & & 0 \\ 0 & \cdots & 0 & \mathbf{I}_{(D-1),(D-1)} \end{bmatrix}, \quad (16)$$

$$\Delta = \begin{bmatrix} \cos(\theta_\tau) & 0 & \cdots & 0 & -\sin(\theta_\tau) \\ 0 & & & & 0 \\ \vdots & & \mathbf{I}_{(l-\psi-1), (l-\psi-1)} & & \vdots \\ 0 & & & & 0 \\ \sin(\theta_\tau) & 0 & \cdots & 0 & \cos(\theta_\tau) \end{bmatrix}, \quad (16a)$$

where $\mathbf{T}_{\psi l}$ denotes the $D \times D$ rotation matrix. $\mathbf{I}_{(\psi-1), (\psi-1)}$ denotes the $(\psi-1) \times (\psi-1)$ identity matrix. $\mathbf{I}_{(D-l), (D-l)}$ denotes the $(D-l) \times (D-l)$ identity matrix. $\mathbf{I}_{(l-\psi-1), (l-\psi-1)}$ denotes the $(l-\psi-1) \times (l-\psi-1)$ identity matrix, and $1 \leq \psi < l \leq D$. Δ denotes a $(l-\psi+1) \times (l-\psi+1)$ matrix. D is the largest dimension of the separation matrix, and ψ, l represent the number of row and column of the element containing the rotation angle in the rotation matrix, respectively. τ represents the sequence number of the rotation matrix arranged from left to right at the right end of the equal sign of equation (15). So θ_τ represents the rotation angle of the τ -th rotation matrix arranged from left to right at the right end of the equal sign, where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]$ and $\tau = 1, 2, \dots, d$. Based on the \mathbf{m}_k^ε and equation (15), we can obtain the separation matrix $\mathbf{W}(\mathbf{m}_k^\varepsilon)$ corresponding to

the k -th quantum slime mould's position using the rotation angle of the rotation matrix as position information of the quantum slime mould. Preprocessing the separated signals obtained from $\mathbf{W}(\mathbf{m}_k^\varepsilon)$ can get $\overline{\mathbf{W}}(\mathbf{m}_k^\varepsilon)$.

Fitness function can be constructed based on the hybrid optimization objective function and the process above. Fitness value for the k -th quantum slime mould's position can be calculated by the fitness function, that is,

$$F(\mathbf{m}_k^\varepsilon) = C[\overline{\mathbf{W}}(\mathbf{m}_k^\varepsilon)]. \quad (17)$$

According to the fitness function, the fitness value of each quantum slime mould's position is calculated, and the fitness values are sorted. Then we can find the quantum position with the largest fitness value and the quantum position with the smallest fitness value in the population up to the current generation. Finally, the global optimal quantum position, that is, $\overline{\mathbf{m}}_{\text{best}}^\varepsilon = [\overline{m}_{\text{best},1}^\varepsilon, \overline{m}_{\text{best},2}^\varepsilon, \dots, \overline{m}_{\text{best},d}^\varepsilon]$ and the worst quantum position, that is, $\overline{\mathbf{m}}_{\text{worst}}^\varepsilon = [\overline{m}_{\text{worst},1}^\varepsilon, \overline{m}_{\text{worst},2}^\varepsilon, \dots, \overline{m}_{\text{worst},d}^\varepsilon]$ can be determined. $\overline{\mathbf{m}}_{\text{best}}^\varepsilon$ and $\overline{\mathbf{m}}_{\text{worst}}^\varepsilon$ can be mapped to get $\mathbf{m}_{\text{best}}^\varepsilon$ and $\mathbf{m}_{\text{worst}}^\varepsilon$. The formula of $w(\mathbf{m}_k^\varepsilon)$ in equation (11) is listed as follows:

$$w(\mathbf{m}_{\overline{\mathbf{I}}(q)}^\varepsilon) = \begin{cases} 1 + u_{\overline{\mathbf{I}}(q)}^\varepsilon \times \log\left(\frac{F(\mathbf{m}_{\text{best}}^\varepsilon) - F(\mathbf{m}_{\overline{\mathbf{I}}(q)}^\varepsilon)}{F(\mathbf{m}_{\text{best}}^\varepsilon) - F(\mathbf{m}_{\text{worst}}^\varepsilon)} + 1\right), & q = 1, 2, \dots, \frac{\overline{M}}{2}, \\ 1 - u_{\overline{\mathbf{I}}(q)}^\varepsilon \times \log\left(\frac{F(\mathbf{m}_{\text{best}}^\varepsilon) - F(\mathbf{m}_{\overline{\mathbf{I}}(q)}^\varepsilon)}{F(\mathbf{m}_{\text{best}}^\varepsilon) - F(\mathbf{m}_{\text{worst}}^\varepsilon)} + 1\right), & q = \frac{\overline{M}}{2} + 1, \frac{\overline{M}}{2} + 2, \dots, \overline{M}, \end{cases} \quad (18)$$

where $\overline{\mathbf{I}}(q)$ denotes the q -th element in the label sequence vector $\overline{\mathbf{I}}$, and according to the fitness values, the individual labels of these quantum slime moulds are sorted (in descending order) to obtain $\overline{\mathbf{I}}$. $u_{\overline{\mathbf{I}}(q)}^\varepsilon$ denotes a uniform random number within $[0,1]$.

And, the discovery probability \overline{p}_k is calculated by

$$\overline{p}_k = \tanh|F(\mathbf{m}_k^\varepsilon) - F(\mathbf{m}_{\text{best}}^\varepsilon)|. \quad (19)$$

The newly generated quantum position $\overline{\mathbf{m}}_k^{\varepsilon+1}$ can be mapped as $\mathbf{m}_k^{\varepsilon+1}$, and the fitness value of the $\mathbf{m}_k^{\varepsilon+1}$ can be calculated according to $F(\mathbf{m}_k^{\varepsilon+1})$. Then the quantum positions are selected by the greedy selection strategy. If $F(\mathbf{m}_k^{\varepsilon+1}) < F(\mathbf{m}_k^\varepsilon)$, then $\overline{\mathbf{m}}_k^{\varepsilon+1} = \overline{\mathbf{m}}_k^\varepsilon$ and $F(\mathbf{m}_k^{\varepsilon+1}) = F(\mathbf{m}_k^\varepsilon)$. The quantum positions of the quantum slime moulds are sorted after greedy selection according to fitness value to find the quantum position with the largest fitness value and the quantum position with the smallest fitness value. In this way, we can record these quantum positions. $\overline{\mathbf{m}}_{\text{best}}^{\varepsilon+1} = [\overline{m}_{\text{best},1}^{\varepsilon+1}, \overline{m}_{\text{best},2}^{\varepsilon+1}, \dots, \overline{m}_{\text{best},d}^{\varepsilon+1}]$ denotes the recorded quantum position, that is, the global optimal quantum position after update. And, $\overline{\mathbf{m}}_{\text{worst}}^{\varepsilon+1} = [\overline{m}_{\text{worst},1}^{\varepsilon+1}, \overline{m}_{\text{worst},2}^{\varepsilon+1}, \dots, \overline{m}_{\text{worst},d}^{\varepsilon+1}]$ denotes the recorded quantum position, that is, the worst quantum position after update.

According to the introduction and analysis above, the description of QSMA-BSS in impulse noise is as follows:

- Step 1 The observational signals are received, and then the proposed filtering system is constructed to filter the observational signals.
- Step 2 The filtered observation signals are preprocessed, and a hybrid optimization objective function is constructed.
- Step 3 The quantum slime moulds' fitness is constructed and calculated; then the global optimal quantum position and the worst quantum position are determined.
- Step 4 Each quantum slime mould's quantum rotation angle is updated by equation (11). The quantum position of each quantum slime mould is updated using a simulated quantum rotation gate according to equation (14).
- Step 5 The updated positions' fitness values are calculated and evaluated by greedy selection to renew the global optimal quantum position and the worst quantum position.

Step 6 If $\varepsilon \geq G^*$, output the global optimal position and its corresponding separation matrix $\mathbf{W}(\mathbf{m}_{\text{best}}^{\varepsilon+1})$; then we can obtain the separated signals $\mathbf{y}(t) = \mathbf{W}(\mathbf{m}_{\text{best}}^{\varepsilon+1})\mathbf{z}(t)$; otherwise, set ε to $\varepsilon + 1$, then return to Step 4.

3.3. Computational Complexity of QSMA. For the iterations of the QSMA, each quantum position needs to be mapped to position, and the computational complexity is $O(\overline{M}d)$, where \overline{M} represents the population size and d represents the dimension of each quantum position. The quantum rotation angle of each quantum position is calculated by equation (11) with the computational complexity $O(\overline{M}d)$. Each quantum position is updated by equation (14) with the computational complexity $O(\overline{M}d)$. Besides, the fitness value of each quantum slime mould is calculated, and the quantum individuals are updated according to the greedy selection strategy; then the global optimal quantum position is updated. The computational complexity is $O(\overline{M})$. The weight of each quantum slime mould is calculated by equation (18) with the computational complexity $O(\overline{M})$. And the discovery probability of each quantum slime mould is calculated by equation (19) with the computational complexity $O(\overline{M})$.

Upon termination of the QSMA after ε iterations, the computational complexity is $O(\varepsilon(3\overline{M} + 3\overline{M}d))$.

3.4. Convergence Analysis of QSMA. Mathematical analysis is performed on the convergence performance of the continuous space optimization algorithm. By using the oscillation operator and the greedy selection strategy of these quantum slime moulds, a better large population can be obtained approximately from a finite population. Then we perform the convergence analysis from a probability perspective. The oscillation operator, that is, a mutation operator increases the population's diversity, and the greedy selection strategy can retain the optimal quantum position until the current iteration in the evolution process. The combination of greedy selection strategy and oscillation operator is able to search for the global optimum in multidimensional space with a sequence of populations. The oscillation equation is designed to ensure that the overall convergence is close to the neighborhood of the optimal point. Next, we will prove that after a sufficient number of iterations, the overall probability density function should be concentrated near the global optimal value of the objective function value.

Let $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{\overline{M}}]^T$. According to the continuous optimization problem shown in equation (17), which is also represented by $\max F: \mathbf{M} \subset \widehat{\mathbf{R}}^d$, where F is called the objective function, $\mathbf{M} \subset \widehat{\mathbf{R}}^d$ denotes the feasible region, d is the dimension of \mathbf{M} , and $\widehat{\mathbf{R}}$ is the set of the real numbers. Then some common assumptions are formulated about \mathbf{M} and F :

(i) F has finite global optimal points in the feasible region \mathbf{M} , and the maximum is called F^*

(ii) $0 < F_{\min} \leq F(\mathbf{m}) \leq F_{\max} < \infty, \forall \mathbf{m} \in \mathbf{M}$, where F_{\max} and F_{\min} mean the maximum and minimum of the objective function value, respectively

(iii) $\text{vol}(\mathbf{M}) > 0$, where vol denotes the d -dimensional Lebesgue measure of a set

(iv) $\forall \sigma > 0, \Omega(\sigma) = \{\mathbf{m} \in \mathbf{M}, F(\mathbf{m}) \geq F^* - \sigma\}$ satisfies $\text{vol}(\Omega(\sigma)) > 0$

Based on the above conditions, a lemma is given in the following.

Lemma 1. For $\forall \sigma > 0$, the probability of the newly generated position falling into the set $\Omega(\sigma)$ satisfies

$$P(\mathbf{m}^\varepsilon \in \Omega(\sigma)) \geq \kappa(\varepsilon) > 0, \quad (20)$$

where $\kappa(\varepsilon)$ is a number that may correspond to ε and P represents the event probability of (\cdot) . If $\sum_{\varepsilon=1}^{\infty} \kappa(\varepsilon)$ diverges, the algorithm is convergent, and the global optimal solution is obtained. And it is irrelevant to the initialized population.

Proof. If the probability of the newly generated position falling into the set $\Omega(\sigma)$ satisfies $P(\mathbf{m}^\varepsilon \in \Omega(\sigma)) \geq \kappa(\varepsilon) > 0$ at the ε -th iteration, the probability that this event will not occur can be expressed as $P(\mathbf{m}^\varepsilon \notin \Omega(\sigma)) \leq 1 - \kappa(\varepsilon)$. Moreover, the probability of the event that a position with ε updates but never falls into the set $\Omega(\sigma)$ can be expressed as follows:

$$P_{\text{no}}(\varepsilon) = \prod_{h=1}^{\varepsilon} P_{\text{no}}^h(\mathbf{m}^h \notin \Omega(\sigma)) \leq \prod_{h=1}^{\varepsilon} (1 - \kappa(h)). \quad (21)$$

Since the greedy selection strategy can preserve the optimal position, there is a possibility that a position falls into the set $\Omega(\sigma)$. Consequently, for $\forall \sigma > 0$, we have the following statement:

$$P(F^* - F_{\max}^\varepsilon > \sigma) \leq P_{\text{no}}(\varepsilon), \quad (22)$$

where F_{\max}^ε is the maximal objective function value at the ε -th iteration.

When ε in equation (21) tends to be infinite, the following equation holds:

$$\lim_{\varepsilon \rightarrow \infty} P_{\text{no}}(\varepsilon) = \prod_{\varepsilon=1}^{\infty} (1 - \kappa(\varepsilon)) = 0. \quad (23)$$

From the statistics mathematical perspective, we can derive that $\prod_{\varepsilon=1}^{\infty} (1 - \kappa(\varepsilon)) = 0 \iff \sum_{\varepsilon=1}^{\infty} \kappa(\varepsilon)$ diverges, where \iff represents equivalent in the equation.

Substitute (23) into (22), and we get

$$\lim_{\varepsilon \rightarrow \infty} P(F^* - F_{\max}^\varepsilon > \sigma) = 0, \quad (24)$$

and $\sum_{\varepsilon=1}^{\infty} \kappa(\varepsilon)$ diverges.

Then the algorithm converges. It is obvious that the convergence performance is irrelevant to the initialized population.

During the oscillation process, the slime moulds are generated in two methods to increase the diversity of its

population, which can be expressed by oscillation equations (11) and (14). QSMA is constantly evolving toward the optimal solution by continuously updating the quantum position, and the quantum position can be linearly mapped to the position. The main generating equation of the quantum rotation angle is equation (11). Consequently, we mainly analyze equations (11) and (14).

When $\bar{r}_k \geq \bar{p}_k$, because equation (12) and $\lim_{\varepsilon \rightarrow G^*} \lambda = \lim_{\varepsilon \rightarrow G^*} (\arctan h(-(\varepsilon/G^*) + 1)) \rightarrow 0$, the value range of v_1 starts to decrease from $[-\infty, \infty]$, that is,

$$\bar{u}_{kr}^\varepsilon (\bar{m}_{\text{mean},r}^\varepsilon - \bar{m}_{kr}^\varepsilon) + \hat{u}_{kr}^\varepsilon (\bar{m}_{\text{worst},r}^\varepsilon - \bar{m}_{kr}^\varepsilon) = \bar{u}_{kr}^\varepsilon \bar{m}_{\text{mean},r}^\varepsilon + \hat{u}_{kr}^\varepsilon \bar{m}_{\text{worst},r}^\varepsilon - (\bar{u}_{kr}^\varepsilon + \hat{u}_{kr}^\varepsilon) \bar{m}_{kr}^\varepsilon. \quad (25)$$

We can obtain $((\bar{u}_{kr}^\varepsilon \bar{m}_{\text{mean},r}^\varepsilon + \hat{u}_{kr}^\varepsilon \bar{m}_{\text{worst},r}^\varepsilon) / (\bar{u}_{kr}^\varepsilon + \hat{u}_{kr}^\varepsilon)) - \bar{m}_{kr}^\varepsilon$ from the above equation (25) by dividing both sides of the equation by $\bar{u}_{kr}^\varepsilon + \hat{u}_{kr}^\varepsilon$ at the same time. Equations (11) and (12) can be thought that the current quantum position is attracted by the quantum position \bar{p}^ε , that is, composed of the $\bar{\mathbf{m}}_{\text{mean}}^\varepsilon = [\bar{m}_{\text{mean},1}^\varepsilon, \bar{m}_{\text{mean},2}^\varepsilon, \dots, \bar{m}_{\text{mean},d}^\varepsilon] = (1/M) \sum_{k=1}^M \bar{\mathbf{m}}_k^\varepsilon$ and $\bar{\mathbf{m}}_{\text{worst}}^\varepsilon = [\bar{m}_{\text{worst},1}^\varepsilon, \bar{m}_{\text{worst},2}^\varepsilon, \dots, \bar{m}_{\text{worst},d}^\varepsilon]$. The quantum position \bar{p}^ε is constantly updated that can be linearly mapped to position \mathbf{p}^ε . And the r -th dimension of quantum position \bar{p}^ε can be described as $((\bar{u}_{kr}^\varepsilon \bar{m}_{\text{mean},r}^\varepsilon + \hat{u}_{kr}^\varepsilon \bar{m}_{\text{worst},r}^\varepsilon) / (\bar{u}_{kr}^\varepsilon + \hat{u}_{kr}^\varepsilon))$ [35]. Both \bar{u}_{kr}^ε and \hat{u}_{kr}^ε are known to be random numbers with uniform distribution between 0 and 1. The effect of the quantum rotation gate operator is slight in the quantum area [0,1]; therefore, we can assume that the quantum rotation gate does not affect the current probability distribution. Then it can be considered that the quantum positions, and their mapping positions are updated toward the quantum position \bar{p}^ε and the position \mathbf{p}^ε with uniform

$v_1 \in [-\lambda, \lambda] \rightarrow 0$ and $\delta_{kr}^{\varepsilon+1} \rightarrow 0$; finally, we have $\bar{m}_{kr}^{\varepsilon+1} \rightarrow \bar{m}_{\text{best},r}^\varepsilon$, where \rightarrow represents infinite approach. The newly generated quantum position in each iteration is constantly approaching the optimal quantum position in the previous iteration. Meanwhile, QSMA introduces a greedy selection strategy to retain the optimal solution and the rotation characteristics of the quantum rotation gate can make the quantum position evolve to the optimal quantum position.

When $\bar{r}_k < \bar{p}_k$, because

random distribution. All the dimensions of $\mathbf{m} = [m_1, m_2, \dots, m_d]$ are independent identically distributed, which follow uniform random distribution $U(m_r^{\min}, m_r^{\max})$, where $r = 1, 2, \dots, d$. The probability density function can be defined as follows:

$$f_u(\mathbf{m}) = \begin{cases} \frac{1}{\prod_{r=1}^d (m_r^{\max} - m_r^{\min})}, & \mathbf{m} \in \mathbf{M}, \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

Owing to the rotation characteristics of quantum rotation gate, the quantum positions are led to evolve towards the optimal quantum position; meanwhile, the positions have a high probability of falling into the set $\Omega(\sigma)$, which is more than the probability generated by the uniform random density function. Thus, for $\forall \mathbf{m} \in \mathbf{M}$ generated by equations (11) and (12), we have

$$P(\{\mathbf{m} \in \Omega(\sigma)\}) \gg \int_{\Omega(\sigma)} f_u(\mathbf{m}) d\mathbf{m} = \text{vol}(\Omega(\sigma)) \frac{1}{\prod_{r=1}^d (m_r^{\max} - m_r^{\min})} > 0. \quad (27)$$

Based on the proof of Lemma 1, it can be concluded that QSMA can converge to the optimal solution through these equations. What needs attention is how the oscillation operator and the greedy selection strategy work. The oscillation operator expands the distribution and guides the solution to converge to the optimal solution. The greedy selection strategy retains the optimal solution to avoid falling into the local optimum. This means that QSMA will eventually converge to the global optimal value, and its convergence performance is irrelevant to the initial population. \square

3.5. Limitation Analysis. Just like other BSS methods, although QSMA-BSS achieves superior performance, this method is only suitable for BSS when the source signal is interfered by impulse noise. When the observation signal is interfered by impulse noise, the performance of the method decreases. Besides, when QSMA is applied to solve the superdimensional optimization problem, the performance may be reduced.

4. Simulation Results

The simulation software used is MATLAB R2018a. The hardware details are as follows: Inter (R) Core (TM) i5-10300H CPU (2.50 GHz), which processes 16 GB of RAM on Microsoft Windows 10.

4.1. Implementation of the Proposed Method. In the experiment, three different basic communication signals are used as source signals. $s_1(t) = \text{sgn}(\cos(2 * \pi * 155 * t))$ is a binary signal. $s_2(t) = \sin(2 * \pi * 90 * t)$ is a low-frequency sinusoidal signal. $s_3(t) = \sin(2 * \pi * 9 * t) * \sin(2 * \pi * 300 * t)$ is an amplitude modulation signal. Three source signals ($D = 3$) are sampled at a frequency of 10 kHz, and the number of sampling points is 4,000 ($N = 4000$). Strong impulse noise ($\alpha = 0.9$) is added to $s_2(t)$. MSNR of impulse noise is set as 10 dB. Figures 3(a)–3(f) show the complete process of BSS in impulse noise.

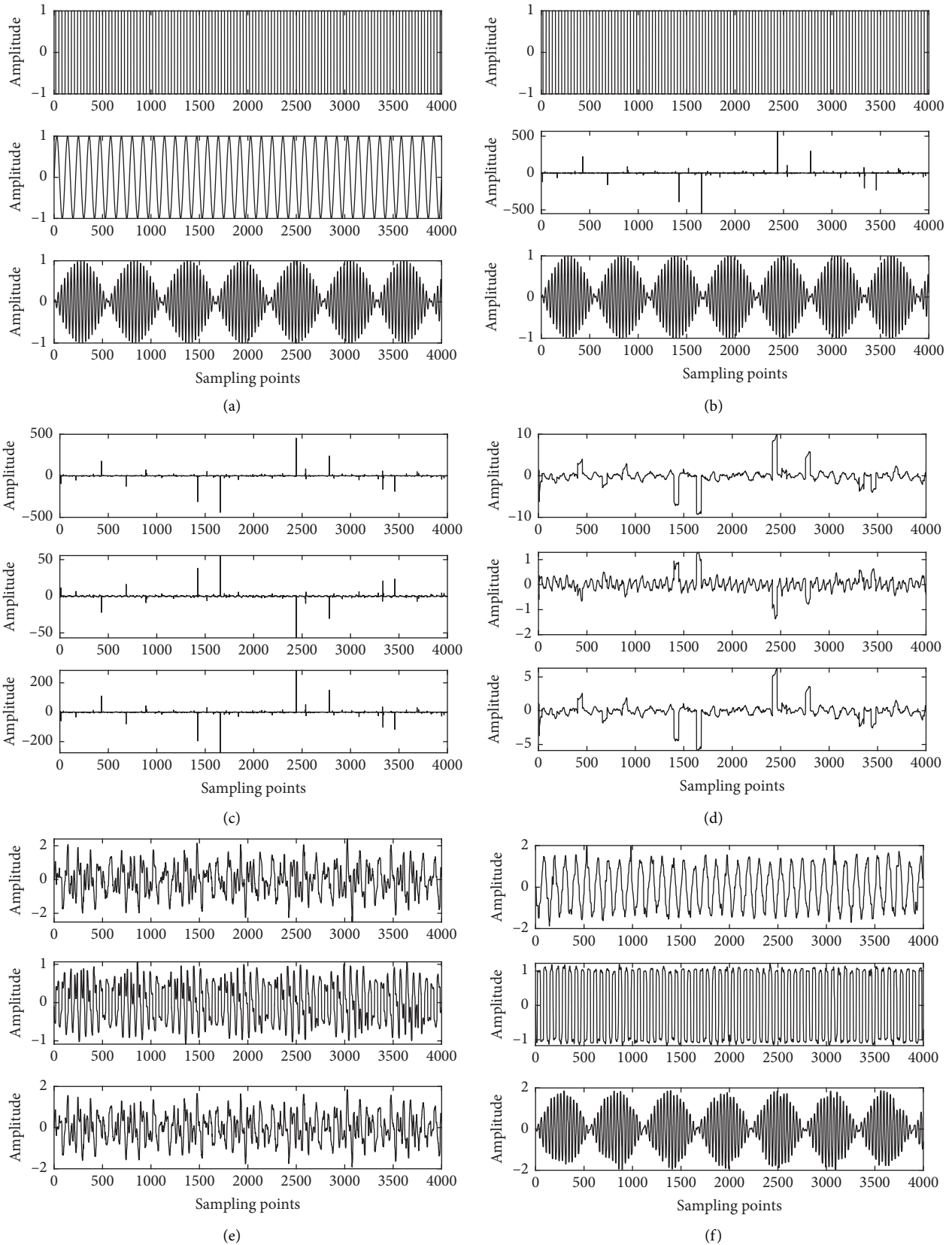


FIGURE 3: The process of BSS in impulse noise: (a) waveform graph of $s(t)$, (b) waveform graph of $\bar{s}(t)$, (c) waveform graph of $\bar{x}(t)$, (d) waveform graph of $\bar{x}(t)$, (e) waveform graph of $x(t)$, and (f) waveform graph of $\hat{y}(t)$.

The waveform graph of $s(t)$ is shown in Figure 3(a). The waveform graph of $\bar{s}(t)$ is shown in Figure 3(b). Figure 3(b) illustrates that $s_2(t)$ is completely covered by strong impulse noise.

The hybrid system matrix \mathbf{A} is set as
$$\begin{bmatrix} 0.5 & 0.8 & 1 \\ -0.5 & -0.1 & 0.5 \\ 0.5 & 0.5 & 0.8 \end{bmatrix}.$$

Because we cannot get any information about source signals, we have to observe the $\bar{\mathbf{x}}(t) = \mathbf{A}\bar{\mathbf{s}}(t)$ to determine Λ . The waveform graph of $\bar{\mathbf{x}}(t)$ is shown in Figure 3(c), and this figure demonstrates that it is impossible for us to determine Λ by observing $\bar{\mathbf{x}}(t)$ due to the large amplitude of $\bar{\mathbf{x}}(t)$. After a lot of experimentation, we find that it is easy to cause noise suppression ability to decrease or the signal waveform after filtering is distorted if a fixed Λ is randomly determined. Figure 3(c) demonstrates that the observation signals have been completely covered by impulse noise. Consequently, it is necessary to filter the $\bar{\mathbf{x}}(t)$ if we want to determine Λ . The neighborhood window size of moving average filter is set to 50.

The waveform graph of $\tilde{\mathbf{x}}(t)$ is shown in Figure 3(d), and this figure demonstrates that the amplitude of impulse noise is reduced through moving average filtering. Consequently, we can determine Λ by equation (6).

And $\tilde{\mathbf{x}}(t)$ are filtered through the MAF-DPVT-MF to obtain $\mathbf{x}(t)$. The waveform graph of $\mathbf{x}(t)$ is shown in Figure 3(e), and this figure demonstrates that impulse noise is suppressed.

DPVT's function coefficient c is set to 0.8, and the neighborhood window size of the median filter is set to 10. The parameter setting of c refers to literature [27].

The negative entropy function coefficient χ is set as 48.

Parameters setting for the proposed QSMA is as follows: $\bar{M} = 30$, $G^* = 40$, $d = 3$, $m_r^{\max} = 2\pi$, $m_r^{\min} = 0$, $v_2^{\max} = 1$, and $v_2^{\min} = 0$. It is known that the convergence performance is irrelevant to the initialized population size from the convergence analysis of QSMA. d is related to the dimension of the optimization problem. Consequently, the performance of QSMA is not limited to parameters setting.

After many experiments, these parameters are found to be suitable, so good simulation results can be achieved.

The separated signals acquired are shown in Figure 3(f). The simulation result illustrates that the proposed method can achieve BSS when source signals are interfered by strong impulse and low MSNR noise. Meanwhile, the proposed method can also recover the source signals covered by noise to a certain extent.

Next, instead of filtering the signals through MAF to determine the threshold value, we delete the MAF and randomly set Λ to fixed value to observe the effect of signal separation (Figure 4).

If Λ is set to a fixed value, Figure 4 demonstrates that the performance of the method will deteriorate. The fixed threshold value may cause the threshold value to be too small for the observation signal covered by noise. While the noise is suppressed, the signal is also suppressed, resulting in the distortion of the signal waveform.

4.2. Performance of QSMA Based on Hybrid Optimization Objective Function. To testify the superiority of the proposed method, the other nine intelligent algorithms that are GA, PSO, BFA, SMA, WOA, ABC, GWO, CSA, and BA are used in the following simulations as comparisons.

The number of population individuals for all algorithms is set to 30, and the iterations' maximum number for all algorithms is set as 500. The simulation result is the mean of 100 simulations. Other parameters setting for QSMA remains unchanged.

In PSO, the inertia weight coefficient is linearly decreased from 0.9 to 0.1 over the course of iterations. The two acceleration constants are both equal to 2.0, and the maximum speed is set as 0.1 [7]. GA's crossover rate is set as 0.8; then the mutation rate is set as 0.1; and other parameters setting refers to [8]. BFA's mutation rate is set as 0.5, and other parameters setting refers to [9]. SMA's parameters setting is derived from original literature [10]. In ABC, the percentage of onlooker bees is 50% of the colony. The percentage of employed bees is 50% of the colony, and the number of scout bees is selected as 1 [11]. In WOA, the constant for defining the shape of the logarithmic spiral is set to 1, and other parameters setting refers to [12]. In GWO, the inertia weight coefficient is linearly decreased from 2 to 0 over the course of iterations [13]. In CSA, the flight step is set as 2, and crow's perception probability is set as 0.1 [14]. In BA, the initial loudness is set to 1 and the initial pulse frequency is set to 0.5. Then the two constants affecting the above two parameters are both set to 0.9, and other parameters setting refers to [15].

The characteristic index of impulse noise is 0.9 ($\alpha = 0.9$), and the MSNR is set as 10 dB.

Figure 5 demonstrates algorithm test results of the hybrid optimization objective function under different weight coefficients.

When $a = 1, b = 0$, based on the MK, the objective function is constructed. The test result shows that QSMA can achieve the best convergence performance. When $a = 0.5, b = 0.5$, based on the MK and MNE, the objective function is constructed. The test result shows that QSMA's convergence performance gradually deteriorates. When $a = 0, b = 1$, based on the MNE, the objective function is constructed. The test result shows that other algorithms' convergence performance is gradually better than QSMA.

Figure 5 demonstrates that different algorithms correspond to different optimal independence criteria. PSO and WOA can obviously show better convergence performance when the objective function is designed based on MNE. QSMA and SMA can obviously show better convergence performance when the objective function is designed based on MK. In addition, the change of the independence criterion has little effect on the performance of some algorithms. In summary, it is significant to design the hybrid optimization objective function.

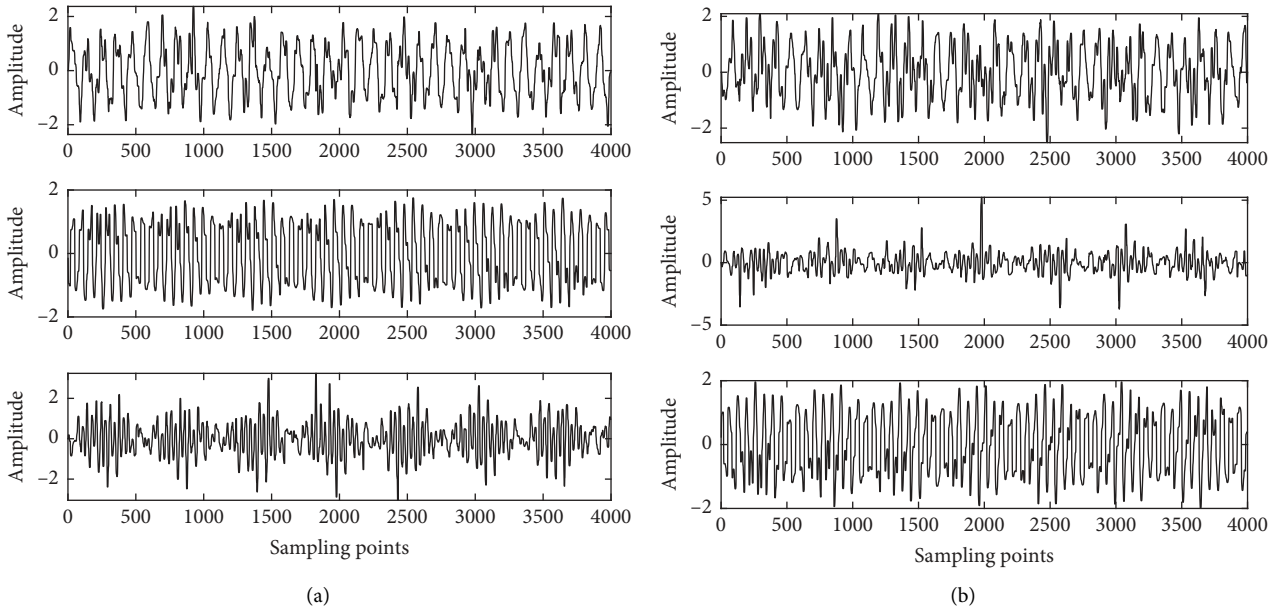


FIGURE 4: Waveform graph of $\hat{y}(t)$ without MAF: (a) $\Lambda = 10$ and (b) $\Lambda = 20$.

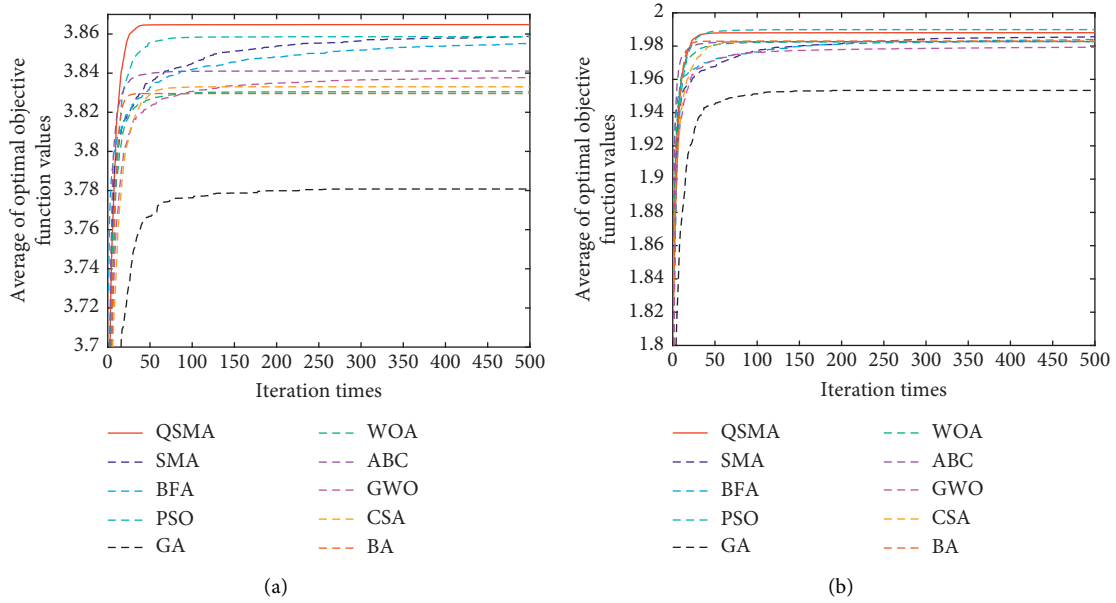


FIGURE 5: Continued.

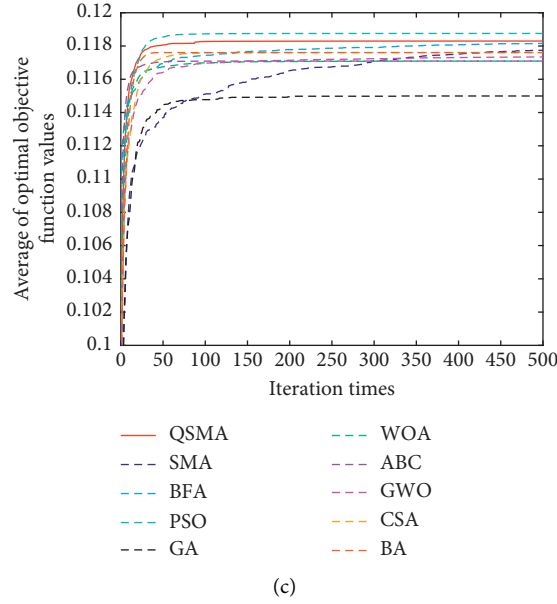


FIGURE 5: Comparison of convergence curves for hybrid optimization objective function: (a) $a = 1, b = 0$, (b) $a = 0.5, b = 0.5$, and (c) $a = 0, b = 1$.

4.3. Performance of QSMA-BSS Based on Hybrid Optimization Objective Function. The crosstalk error is used as the performance evaluation index, and the crosstalk error is calculated by the following formula:

$$\bar{P} = \frac{1}{D(D-1)} \sum_{i=1}^D \left\{ \left(\sum_{\bar{l}=1}^D \frac{|\bar{G}_{i\bar{l}}|}{\max_{\bar{l}} |\bar{G}_{i\bar{l}}|} - 1 \right) + \left(\sum_{\bar{l}=1}^D \frac{|\bar{G}_{\bar{l}i}|}{\max_{\bar{l}} |\bar{G}_{\bar{l}i}|} - 1 \right) \right\}, \quad (28)$$

where $\bar{\mathbf{G}}$ denotes the system matrix and $\bar{\mathbf{G}} = \mathbf{WV}\mathbf{A}$. $|\bar{G}_{i\bar{l}}|$ denotes the i -th row and the \bar{l} -th column element of $|\bar{\mathbf{G}}|$. $\max_{\bar{l}} |\bar{G}_{i\bar{l}}|$ denotes the largest element in the i -th row of $|\bar{\mathbf{G}}|$, and this element is located in the \bar{l} -th column. $|\bar{G}_{\bar{l}i}|$ denotes the \bar{l} -th row and the i -th column element of $|\bar{\mathbf{G}}|$. $\max_{\bar{l}} |\bar{G}_{\bar{l}i}|$ denotes the largest element in the \bar{l} -th row of $|\bar{\mathbf{G}}|$, and this element is located in the i -th column. The separation accuracy of the signal increases with the decrease of \bar{P} .

The number of population individuals for all algorithms is set to 30, and the iterations' maximum number for all algorithms is set as 40. Each experiment runs 100 times in the same MSNR, and the result is the mean of 100 simulations. The other parameters setting for all algorithms remains unchanged.

The BSS methods based on nine algorithms, named PSO-BSS, SMA-BSS, GA-BSS, BFA-BSS, WOA-BSS, ABC-BSS, GWO-BSS, CSA-BSS, and BA-BSS, are shown in Figure 6.

Figure 6 demonstrates the algorithm test results of the hybrid optimization objective function under different weight coefficients; meanwhile, we conduct experiments under strong impulse noise and weak impulse noise environment, respectively. When $a = 1, b = 0$, Figure 6 demonstrates that QSMA-BSS has higher accuracy compared with the other algorithms. When $a = 0, b = 1$, Figure 6 demonstrates that the performance of QSMA-BSS has become worse.

Firstly, the proposed method can achieve BSS no matter under strong impulse noise or weak impulse noise. Besides, it can be known from the simulation results that the average of crosstalk error is still less than 0.3 when MSNR is 0 dB, which indicates that the performance of this method is still very superior under low MSNR. Finally, in addition to QSMA, other algorithms can also be applied to the method proposed in this paper.

The BSS method based on MK and QSMA is named QSMA-MK-BSS. The BSS method based on MNE and PSO is named PSO-MNE-BSS. Therefore, the other comparison methods are named ABC-MNE-BSS, WOA-MNE-BSS, GWO-MNE-BSS, CAS-MNE-BSS, and BA-MNE-BSS accordingly (Figure 7).

Figure 7(a) shows that the performance of the QSMA-MK-BSS is still better than other algorithms designing objective functions based on MNE. Consequently, the proposed QSMA is still significant and useful. Meanwhile, it is necessary for BSS to design hybrid optimization objective function. By the designed function, we can choose different optimal independence criteria for different algorithms and then design the objective function based on the optimal independence criterion of the algorithm. Figure 7(b) shows that as the coefficient a continues to increase, the performance of QSMA-BSS gets better and better. In summary, the optimal independence criterion for QSMA is MK, and it can achieve higher precision when the objective function is designed based on the optimal independence criterion.

Fractional low-order moments (FLOM) are introduced into FAST-ICA in [23]. Consequently, we can also introduce fractional low-order covariance (FLOC) [36] into FAST-ICA. Two methods are achieved as comparisons to show the superiority of the proposed method. Two methods are named FLOM-FAST-ICA and FLOC-FAST-ICA (Figure 8).

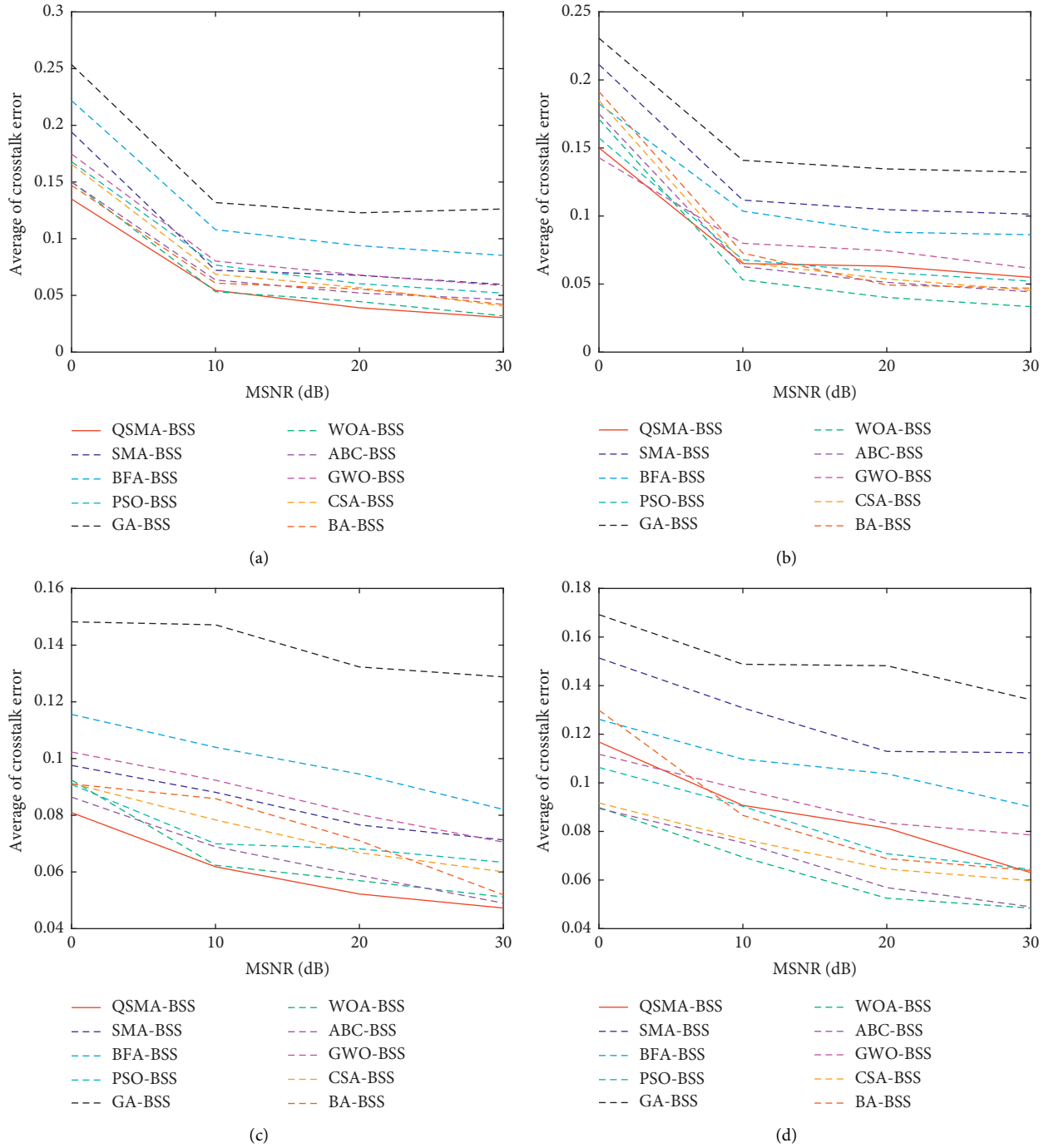


FIGURE 6: Comparison of signal separation effect: (a) $a = 1, b = 0, \alpha = 0.9$, (b) $a = 0, b = 1, \alpha = 0.9$, (c) $a = 1, b = 0, \alpha = 1.5$, and (d) $a = 0, b = 1, \alpha = 1.5$.

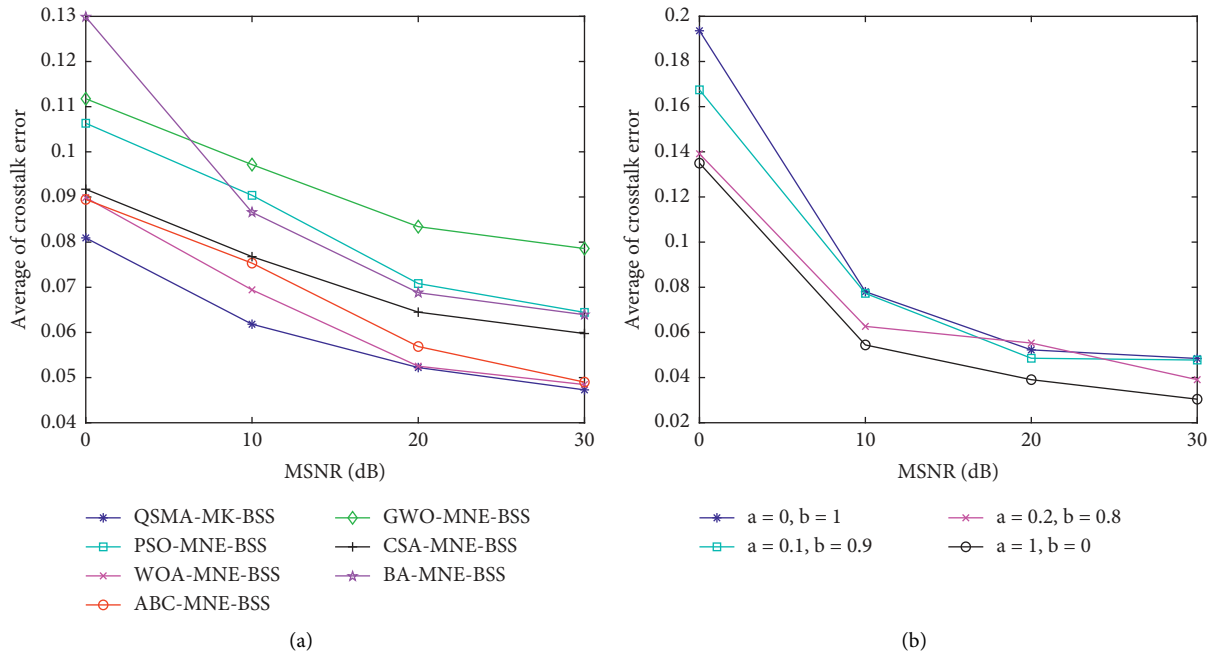


FIGURE 7: Comparison of signal separation effect (a) between QSM-MK-BSS and other methods ($\alpha = 1.5$) and (b) for QSM-BSS under different weight coefficients ($\alpha = 0.9$).

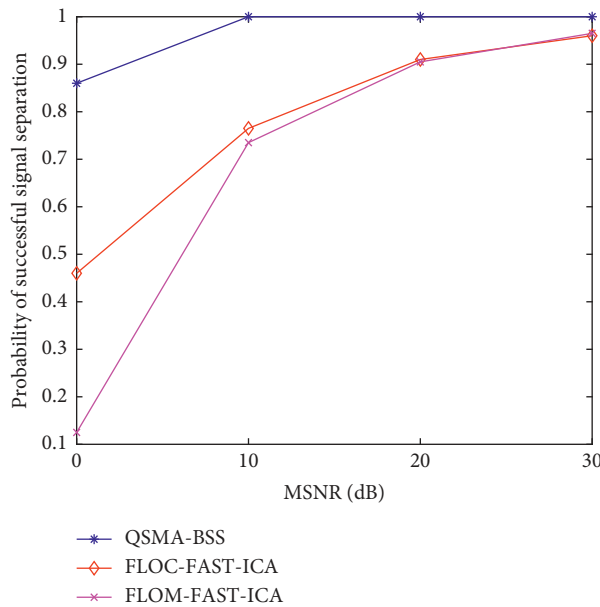


FIGURE 8: Comparison of signal separation effect between QSM-BSS and previous methods.

To testify the superiority of the proposed method, the probability of successful signal separation is calculated. When the crosstalk error is less than 0.3, the experiment is successful. During the simulation, each experiment runs 100 times in the same MSNR.

Figure 8 shows that the performance of the traditional methods deteriorates in strong impulse and low MSNR noise environment, but the QSM-BSS is still effective and robust.

5. Conclusions

In this paper, a filter system named MAF-DPVT-MF is designed. MAF-DPVT-MF does not require any information from source signals or environmental noise as prior knowledge and can effectively suppress impulse noise, thereby eliminating the influence of noise interference on BSS. The hybrid optimization objective function is designed

based on two different independence criteria. Then a new intelligent algorithm named QSMA is proposed to search for the optimal solution of hybrid optimization objective function and its corresponding separation matrix within the search range. According to the change of the performance evaluation index with the weight coefficient, QSMA's optimal independence criterion is determined to get more accurate results. And simulation results illustrate that the designed method is robust and effective when the unknown source signals are interfered by impulse noise with low MSNR. Compared with some traditional methods, the performance of the proposed method is superior. Except these algorithms mentioned in this paper, some of the most representative computational intelligence algorithms can be used to solve the problem, such as monarch butterfly optimization (MBO), earthworm optimization algorithm (EWA), elephant herding optimization (EHO), moth search (MS) algorithm, and Harris hawks optimization (HHO).

In future research, firstly, the proposed filtering system can be applied to other engineering problems to suppress noise. Secondly, the proposed QSMA can also be used to solve other optimization problems. Finally, the work in this paper can be considered a stepping stone to more complicated future research in BSS, such as underdetermined BSS in impulse noise and BSS when the observation signal is interfered by impulse noise.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61571149), the Natural Science Foundation of Heilongjiang Province (no. LH2020F017), and the Initiation Fund for Postdoctoral Research in Heilongjiang Province (no. LBH-Q19098).

References

- [1] B. Pavan, V. J. Louis, A. Subramaniam et al., "Implementation of Fast ICA using memristor crossbar arrays for blind image source separations," *IET Circuits Devices & Systems*, vol. 14, no. 4, pp. 484–489, 2020.
- [2] S. Liu, B. Wang, and L. Zhang, "Blind source separation method based on neural network with bias term and maximum likelihood estimation criterion," *Sensors*, vol. 21, no. 3, pp. 973–998, 2021.
- [3] I. Labounkova, R. Labounek, I. Nestrasil et al., "Blind source separation of retinal pulsatile patterns in optic nerve head video-recordings," *IEEE Transactions on Medical Imaging*, vol. 40, no. 3, pp. 852–864, 2020.
- [4] L. Liu, S. F. Ou, and Y. Gao, "Constrained blind separation algorithm using variable step size and variable momentum factor," *International Journal of Intelligent Information Systems*, vol. 8, no. 4, pp. 77–84, 2019.
- [5] D. P. Acharya, G. Panda, and Y. V. S. Lakshmi, "Effects of finite register length on fast ICA, bacterial foraging optimization based ICA and constrained genetic algorithm based ICA algorithm," *Digital Signal Processing*, vol. 20, no. 3, pp. 964–975, 2009.
- [6] H. Cheng, N. Yu, and J. Liu, "Improved natural gradient algorithms for multi-channel signal separation," *Applied Mechanics and Materials*, vol. 651–653, pp. 2326–2330, 2014.
- [7] M. M. Hoque, M. A. Hannan, and A. Mohamed, "Charging and discharging model of lithium-ion battery for charge equalization control using particle swarm optimization algorithm," *Journal of Renewable and Sustainable Energy*, vol. 8, no. 6, pp. 7847–7858, 2016.
- [8] X. L. Wu and J. Li, "Two layered approaches integrating harmony search with genetic algorithm for the integrated process planning and scheduling problem," *Computers & Industrial Engineering*, vol. 155, pp. 1–14, 2021.
- [9] B. Bhushan and M. Singh, "Adaptive control of DC motor using bacterial foraging algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 4913–4920, 2011.
- [10] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [11] R. J. Wang, "Blind source separation based on adaptive artificial bee colony optimization and kurtosis," *Circuits Systems and Signal Processing*, vol. 40, no. 7, pp. 3338–3354, 2021.
- [12] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: whale optimization algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1–24, 2019.
- [13] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of interconnected power system using grey wolf optimization," *Swarm and Evolutionary Computation*, vol. 27, pp. 97–115, 2016.
- [14] N. A. Al-Thanoon, Z. Y. Algamal, and O. S. Qasim, "Feature selection based on a crow search algorithm for big data classification," *Chemometrics and Intelligent Laboratory Systems*, vol. 212, Article ID 104288, 2021.
- [15] M.-L. Gao, J. Shen, L.-J. Yin et al., "A novel visual tracking method using bat algorithm," *Neurocomputing*, vol. 177, pp. 612–619, 2016.
- [16] R. J. Wang, Y. J. Zhan, and H. F. Zhou, "A class of sequential blind source separation method in order using swarm optimization algorithm," *Circuits, Systems, and Signal Processing*, vol. 35, no. 9, pp. 3220–3243, 2016.
- [17] L. Chen, L. Y. Zhang, Y. J. Guo et al., "Blind source separation based on covariance ratio and artificial bee colony algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 104288, 12 pages, 2014.
- [18] L. Shih-Hsiung and C. S. Yang, "GPSO-ICA: independent component analysis based on gravitational particle swarm optimization for blind source separation," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1943–1957, 2018.
- [19] A. Ebrahimzadeh and S. Mavaddati, "A novel technique for blind source separation using bees colony algorithm and efficient cost functions," *Swarm and Evolutionary Computation*, vol. 14, no. 14, pp. 15–20, 2014.
- [20] M. Kumar and V. E. Jayanthi, "Blind source separation using kurtosis, negentropy and maximum likelihood functions," *International Journal of Speech Technology*, vol. 23, no. 1, pp. 13–21, 2020.

- [21] M. Sahmoudi, K. Abed-Meraim, and M. Benidir, "Blind separation of impulsive alpha-stable sources using minimum dispersion criterion," *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 281–284, 2005.
- [22] J. Long, H. Wang, P. Li, and H. Xie, "A WHT signal detection-based FLO-TF-UBSS algorithm under impulsive noise environment," *Circuits, Systems, and Signal Processing*, vol. 37, no. 7, pp. 2997–3022, 2018.
- [23] C. C. Wang, M. Q. Liu, Q. Chen et al., "Automatic digital modulation recognition in the presence of alpha-stable noise," *Physical Communication*, vol. 43, Article ID 101221, 2020.
- [24] Z. T. Luo, Y. M. Zhan, and R. M. Guo, "Variable tailing nonlinear transformation design based on exponential function in impulsive noise," *Journal of Electronics & Information Technology*, vol. 42, no. 4, pp. 932–940, 2020.
- [25] G. R. Arce and J. G. Gonzalez, "Statistically-Efficient filtering in impulsive environments: weighted myriad filters," *EURASIP Journal on Advances in Signal Processing*, vol. 2002, no. 1, pp. 1–7, Article ID 363195, 2002.
- [26] T. Moon and T. Weissman, "Universal FIR MMSE filtering," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1068–1083, 2009.
- [27] Z. Luo, R. Guo, X. Zhang, and Y. Nie, "Optimal and efficient designs of Gaussian-tailed non-linearity in symmetric α -stable noise," *Electronics Letters*, vol. 55, no. 6, pp. 353–355, 2019.
- [28] K. H. Jong-Hwan Kim and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, 2002.
- [29] H. Gao, Y. N. Du, Y. Du, and C. Li, "Quantum fireworks algorithm for optimal cooperation mechanism of energy harvesting cognitive radio," *Journal of Systems Engineering and Electronics*, vol. 29, no. 1, pp. 18–30, 2018.
- [30] X. Meng, T. Lu, F. Min, and T. Lu, "An effective weighted vector median filter for impulse noise reduction based on minimizing the degree of aggregation," *IET Image Processing*, vol. 15, no. 1, pp. 228–238, 2020.
- [31] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: stable processes and their applications," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 986–1010, 1993.
- [32] J. M. Chambers, C. L. Mallows, and B. W. Stuck, "A method for simulating stable random variables," *Journal of the American Statistical Association*, vol. 71, no. 354, pp. 340–344, 1976.
- [33] A. A. Al-Mbaideen, "Application of moving average filter for the quantitative analysis of the NIR spectra," *Journal of Analytical Chemistry*, vol. 74, no. 7, pp. 686–692, 2019.
- [34] D. Janovská and G. Opfer, "Givens' transformation applied to quaternion valued vectors," *Bit Numerical Mathematics*, vol. 43, no. 5, pp. 991–1002, 2003.
- [35] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [36] G. Zak, A. Wylomanska, and R. Zimroz, "Generalized dependency maps based on fractional lower order covariance in application to local damage detection," *Vibroengineering PROCEDIA*, vol. 7, pp. 82–87, 2016.