Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6673235, 12 pages
https://doi.org/10.1155/2021/6673235

Research Article

Adaptive Reconstruction Algorithm Based on Compressed Sensing Broadband Receiver

Wei-Jian Si,[1,2] Qiang Liu,[1,2] and Zhi-An Deng[1,2]

[1]*College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*Key Laboratory of Advanced Marine Communication and Information Technology, Ministry of Industry and Information Technology, Harbin 150001, China*

Correspondence should be addressed to Zhi-An Deng; dengzhian@hrbeu.edu.cn

Received 23 November 2020; Revised 29 December 2020; Accepted 4 January 2021; Published 15 January 2021

Academic Editor: Fangqing Wen

Existing greedy reconstruction algorithms require signal sparsity, and the remaining sparsity adaptive algorithms can be reconstructed but cannot achieve accurate sparsity estimation. To address this problem, a blind sparsity reconstruction algorithm is proposed in this paper, which is applied to compressed sensing radar receiver system. The proposed algorithm can realize the estimation of signal sparsity and channel position estimation, which mainly consists of two parts. The first part is to use fast search based on dichotomy search, which is based on the high probability reconstruction of greedy algorithm, and uses dichotomy search to cover the number of sparsity. The second part is the signal matching and tracking algorithm, which is mainly used to judge the signal position and reconstruct the signal. Combine the two parts together to realize the blind estimation of the sparsity and the accurate estimation of the number of signals when the number of signals is unknown. The experimental analyses are carried out to evaluate the performance of the reconstruction probability, the accuracy of sparsity estimation, the running time of the algorithm, and the signal-to-noise ratio.

1. Introduction

With the continuous development of radar technology, the electronic countermeasure environment has become increasingly complex, and it will become more complex in the future [1]. First, the analog-to-digital converter (ADC) sampling rate limits the instantaneous bandwidth of the receiver [2]. The use of frequency agile signals requires radar receivers to have increasingly wider instantaneous bandwidths. At present, the development of high-rate ADCs is relatively slow [3–5]. Second, in order to achieve full-probability reception, radar receivers require increasingly hardware, the system is more and more complex, and the volume is getting larger. In order to meet the needs of modern electronic warfare, it is necessary to solve the problems faced by digital channelized receivers [6]. Donoho, Candes et al., Wan et al., and Candes and Tao proved the theory of compressed sensing in 2006 and proposed specific implementation algorithms [7–10]. Compressed sensing is a new sampling method, which is highly concerned in image processing, microwave imaging, wireless communication, radar system, and biomedical engineering [11–18].

In recent years, an analog-to-information converter [19] (AIC) has been proposed based on the compressed sampling theory. The methods to achieve AIC include random demodulation (RD) [20], multicoset sampling (MCS) [21], quadrature compressed sampling (QuadCS) [22], and modulated wideband converter (MWC) [23] structure. The under-Nyquist sampling of the sparse signal can be realized at a sampling rate much lower than the Nyquist theorem, and the compressed sampled data of the signal can be obtained. The modulation broadband converter is currently the most mature multibranch compression sampling structure at. Cohen et al. and Mishali et al. have completed the realization of the hardware structure [24, 25].

Compressed sensing theory is mainly divided into three parts: signal sparse representation, observation matrix, and reconstruction algorithm. Signal sparse representation means that the signal can be sparsely represented in a certain transform domain. The observation matrix reduces the

dimensionality of the signal through domain conversion. Finally, it restores the original signal through a reconstruction algorithm [26]. The reconstruction algorithm is the most important step in compressed sensing. The algorithm can be divided into the following three categories: combination algorithm, convex optimization algorithm, and greedy matching pursuit algorithm [27]. The greedy matching pursuit algorithm has been widely studied due to its fast reconstruction speed and small calculation amount.

Existing greedy matching pursuit algorithms mainly include two types. The first category is greedy matching pursuit algorithms that require sparsity as a priori information, mainly including matching pursuit (MP) [28], orthogonal matching pursuit (OMP) [29], regularized orthogonal matching pursuit (ROMP) [30], compressed sampling matching pursuit (CoSaMP) [31], subspace pursuit (SP) [32], and generalized orthogonal matching pursuit (gOMP) [33] algorithms. The OMP algorithm is one of the simplest matching pursuit algorithms. In each iteration, the atom with the largest inner product of the observation matrix and the residual is selected and added to the support set. Use the least square method to update the residuals. Make the residual continuously approach a certain value to achieve convergence. The OMP algorithm guarantees the orthogonality of iterations, but cannot guarantee the correctness of atoms. So once the wrong atom is introduced, it will affect subsequent iterations. The CoSaMP algorithm adds backtracking ideas when selecting atoms. Each time an atom is selected, not only the atom is added to the support set, but also on the wrong atom to delete. Although the CoSaMP algorithm can improve the reconstruction probability and quickly complete the iterations, as the algorithm increases in sparsity, the reconstruction probability will drop sharply. The gOMP algorithm finds multiple support sets each time, the method of finding at most K times improves the probability of reconstruction. However, the sparsity of the algorithm is greater than K after the iteration.

The other is the adaptive algorithm that does not require sparsity, mainly including stagewise orthogonal matching pursuit (StOMP) [34], stagewise weak orthogonal matching pursuit (SWOMP) [35], and sparsity adaptive matching pursuit (SAMP) [36] algorithms. There are three problems in the aforementioned sparsity adaptive matching pursuit algorithm. The first problem is that the StOMP and SWOMP algorithms' support concentration does not consider the selection of the wrong atom, which leads to the limitation of the reconstruction probability of the algorithm [37]. The second problem is that although the SAMP algorithm has a high reconstruction probability, it cannot accurately estimate the sparsity of the signal during adaptive estimation due to the limitation of the step size [38]. The third problem is that SWOMP algorithm and SAMP algorithm are also susceptible to noise in the signal. Therefore, it is necessary to design a sparsity adaptive matching tracking algorithm that can accurately estimate the signal sparsity and high reconstruction probability in noisy radar signals.

This paper proposes a sparseness adaptive matching tracking algorithm for radar receiver system based on compressed sensing. Aiming at the problem of low reconstruction probability of greedy matching pursuit algorithm. A backtracking-based orthogonal matching pursuit BOMP algorithm is designed using backtracking ideas. In each iteration, the wrong atoms selected in the support set will be eliminated. Compared with other greedy algorithms that require sparsity, it has a higher reconstruction probability. Aiming at the problem of sparsity estimation error. Utilizing the principle of binary search, we devised a method sparsity search. First, set an estimated value of sparsity continuously and change the estimated value of sparsity through judgment criteria. Finally, get an accurate estimated sparsity. The estimated sparsity values are all based on high probability reconstruction. At the same time, due to the characteristics of binary search, it can quickly find the true value within a limited number of steps. For the noise problem in the signal, joining residuals associated with the threshold, the ability dynamically adapts to noise. The matching pursuit algorithm proposed in this paper is compared with similar sparsity adaptive algorithms, which improved the accuracy of output signal-to-noise ratio, reconstruction probability, and sparsity estimation.

The arrangement of this article is as follows. In the second section, the system model of the compressed sensing broadband receiver based on MWC is introduced. In the third section, an orthogonal matching pursuit algorithm based on backtracking is designed, which improves the reconstruction probability. In the fourth section, a sparsity adaptive reconstruction algorithm based on compressed sensing is proposed. In the fifth section, the performance of the algorithm is verified by simulation. Compare the reconstruction probability, sparsity, and running time of several adaptive algorithms under one-dimensional signal. Under the radar signal, the algorithm SNR and sparsity estimation and reconstruction probability are also analyzed. Finally, our conclusions are given in the sixth section.

2. System Model

In this section, the system model of MWC compressed sampling structure and sparse reconstruction algorithm will be introduced. The structure design and working principle of MWC compressed sampling wideband digital receiver are introduced, including the introduction of mathematical expressions and a brief description of the compressed sensing reconstruction algorithm.

2.1. MWC Structure. The radar signal in space is often continuous. In this article, for the convenience of introduction, the signal is set as a discrete signal under the Nyquist sampling rate. This setting will not affect the result. The received signal $x[n]$ is first mixed with the pseudorandom sequence $\tilde{p}_m[n]$ to obtain the mixed signal $\tilde{x}_m[n]$, and the mixed signal is then passed through the corresponding low-pass filter $h[n]$ to obtain the filtered signal $\omega_m[n]$. Finally, the signal is extracted by M_p time and sent to the low-speed ADC for samples to get $y_m[k]$. The discrete compressed sampling structure based on MWC receiver is shown in Figure 1.

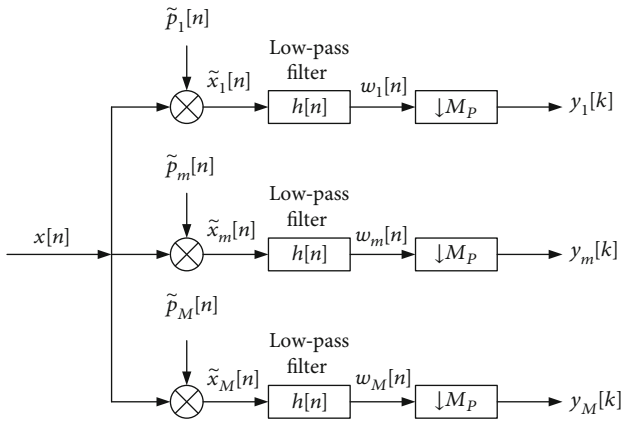

Figure 1: Block diagram of MWC.

The discrete time domain of the radar received signal can be expressed as follows:

$$x[n] = s[n] + \eta[n], 0 \le n \le N. \tag{1}$$

Among them, $s[n]$ is the effective signal component, $\eta[n]$ is the Gaussian white noise component in the signal, and N represents the length of the signal. $x[n]$ is a complex discrete signal; in the Nyquist sampling rate $f_{\mathrm{NYQ}} = 1/T_{\mathrm{NYQ}}$, T_{NYQ} represents the period of Nyquist sampling, and the bandwidth range is $\mathscr{F}_{\mathrm{NYQ}} \triangleq [-f_{\mathrm{NYQ}}/2, f_{\mathrm{NYQ}}/2]$. The discrete-time Fourier transform (DTFT) is $x[n]$ and expressed as follows:

$$X\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}}\right) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi f n T_{\mathrm{NYQ}}}. \tag{2}$$

It can be seen from Figure 1 that the structure contains M parallel branches. Taking the mth path as an example, the period of a periodic pseudorandom sequence $p_m[n]$ is T_{p}, each period contains $M_{\mathrm{p}} = T_{\mathrm{p}} f_{\mathrm{NYQ}}$ elements. The sequence $p_m[n]$ can be expressed as follows:

$$p_m[n] = \begin{cases} \tilde{p}_m[n] & 0 \le n \le M_p - 1 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The discrete Fourier series of the sequence $\tilde{p}_m[n]$ can be expressed as follows:

$$\tilde{P}_m[n] = \frac{1}{M_p} \sum_{l=0}^{M_p-1} P_m(l) e^{j\left(2\pi/M_p\right)nl}. \tag{4}$$

l represents the index number and $0 \le l \le M_{\mathrm{p}} - 1$, and $P_m(l)$ is the discrete Fourier transform coefficient of the sequence $p_m[n]$. The mixing rate of the signal and the sequence is $f_{\mathrm{p}} = 1/T_{\mathrm{p}} = f_{\mathrm{NYQ}}/M_{\mathrm{p}}$. The sequence $p_m[n]$ will be divided $\mathscr{F}_{\mathrm{NYQ}}$ into M_{p} segments. The interval of each segment is the same f_{p}. The position of each segment is called an index. The range of the baseband spectrum segment is $\mathscr{F}_p \triangleq [-f_p/2, f_p/2]$. In order to avoid edge effects, there is a need to meet $f_{\mathrm{p}} \ge B$, where B represents the bandwidth of $s[n]$. The discrete-time Fourier transform (DTFT) of the mixed signal $\tilde{x}_m[n]$ is expressed as follows:

$$\begin{aligned} \tilde{X}_m\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}}\right) &= \sum_{n=0}^{N-1} x[n] \cdot \tilde{p}_m[n] e^{-j2\pi f n T_{\mathrm{NYQ}}} \\ &= \sum_{n=0}^{N-1} x[n] \cdot \frac{1}{M_p} \sum_{l=0}^{M_p-1} P_m(l) e^{j\left(2\pi/M_p\right)nl} e^{-j2\pi f n T_{\mathrm{NYQ}}} \\ &= \frac{1}{M_p} \sum_{l=0}^{M_p-1} P_m(l) X\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}\left(f - lf_p\right)}\right). \end{aligned} \tag{5}$$

The mixed signal $\tilde{x}_m[n]$ is filtered by a low-pass filter $h[n]$. The frequency response is $H(e^{-j2\pi f T_{\mathrm{NYQ}}})$. The cutoff frequency of the low-pass filter is $f_p/2$. The discrete-time Fourier transform (DTFT) filtered signal $w_m[n]$ is expressed as follows:

$$\begin{aligned} W_m\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}}\right) &= \sum_{n=0}^{N-1} \tilde{x}_m[n] \cdot h[n] e^{-j2\pi f n T_{\mathrm{NYQ}}} \\ &= \tilde{X}_m\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}}\right) H\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}}\right) \\ &= \begin{cases} \dfrac{1}{M_p} \displaystyle\sum_{l=0}^{M_p-1} P_m(l) X\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}\left(f - lf_p\right)}\right), f \in F_p \\ 0, f \notin F_p \end{cases}. \end{aligned} \tag{6}$$

After the signal passes through the low-pass filter, the spectral components are within $\mathscr{F}_p \triangleq [-f_p/2, f_p/2]$. But the data rate of the signal is still f_{NYQ}. The data rate can be reduced by decimating the filtered signal. Use low-speed ADC sampling to complete the decimation of the signal. Decimation factor is $M_p = f_{\mathrm{NYQ}}/f_{\mathrm{p}}$. The discrete-time Fourier transform (DTFT) of the sampled signal $y_m[k]$ is expressed as follows:

$$\begin{aligned} Y_m\left(e^{j2\pi \mathrm{fT}_s}\right) &= \sum_{k=0}^{K-1} y_m[k] e^{-j2\pi f n T_s} \\ &= \frac{1}{M_p} \sum_{l=0}^{M_p-1} P_m(l) X\left(e^{j2\pi \mathrm{fT}_{\mathrm{NYQ}}\left(f - lf_p\right)}\right), f \in F_s. \end{aligned} \tag{7}$$

The output can be written in the form of a matrix; the expression is as follows:

$$y(f) = Cz(f), f \in F_s, \tag{8}$$

where $y(f)$ is a column vector of length M. The mth element is $Y_m(e^{j2\pi f T_s})$, $z(f)$ is a column vector of length M_p, and

the lth element is $X(e^{j2\pi f T_{\mathrm{NYQ}}(f-lp)})$. The linear combination can be expressed as follows:

$$P_m(l) = \frac{1}{M_p}\sum_{l=0}^{M_p-1} p_m[n]e^{-j\left(2\pi/M_p\right)nl}. \tag{9}$$

The observation matrix C is $M \times M_p$ dimensional and can be expressed as follows:

$$C = \frac{PF}{M_p}, \tag{10}$$

where P represents $M \times M_p$ dimensional random matrix, F represents $M_p \times M_p$ dimensional discrete Fourier transform matrix, $F_l = [1, e^{j2\pi l/M_p}, \cdots, e^{j2\pi(M_p-1)l/M_p}]^T$ is a column vector, and $(\bullet)^T$ represents transpose. The expanded form of the formula can be obtained.

$$\begin{pmatrix} Y_1\left(e^{j2\pi \mathrm{f}\Gamma_s}\right) \\ Y_2\left(e^{j2\pi \mathrm{f}\Gamma_s}\right) \\ \vdots \\ Y_M\left(e^{j2\pi \mathrm{f}\Gamma_s}\right) \end{pmatrix} = \frac{1}{M_p}\begin{bmatrix} p_{1,0} & \cdots & p_{1,M_p-1} \\ \vdots & \ddots & \vdots \\ p_{M,0} & \cdots & p_{M,M_p-1} \end{bmatrix} \cdot \begin{bmatrix} | & \cdots & | & \cdots & | \\ F_0 & \cdots & F_l & \cdots & F_{M_p-1} \\ | & \cdots & | & \cdots & | \end{bmatrix} \times \begin{bmatrix} X\left(e^{j2\pi \mathrm{f}\Gamma_{\mathrm{NYQ}} f}\right) \\ \vdots \\ X\left(e^{j2\pi \mathrm{f}\Gamma_{\mathrm{NYQ}}\left(f-lf_p\right)}\right) \\ \vdots \\ X\left(e^{j2\pi \mathrm{f}\Gamma_{\mathrm{NYQ}}\left(f-\left(M_p-1\right)f_p\right)}\right) \end{bmatrix}. \tag{11}$$

The above formula is the frequency domain model of MWC compressed sensing receiver, and the feasibility of this structure is proved through mathematical deduction. Therefore, the compressed sensing of signals can be realized through the MWC structure.

2.2. Reconstruction Algorithm. The sampled signal model obtained by using the MWC structure is the infinite measurement vector (IMV) model. In theory, it takes countless iterations to recover the original signal. Xu et al. proposed continuous finiteness in literature [20]. The continuous-to-finite (CTF) algorithm can convert the IMV model of the MWC compressed sample signal into a multiple measurement vector (MMV) model. The signal under this model can be reconstructed from the original signal through the traditional reconstruction algorithm structure. The block diagram of CTF algorithm implementation is shown in the figure.

As can be seen from Figure 2, first, use the sampled signal $y[k]$ to get the matrix Q, and the expression is as follows:

$$Q = \int_{f\in F_s} y(f)y^H(f)df = \sum_{k=0}^{K} y(k)y^T(k), \tag{12}$$

which $(\bullet)^H$ represents the conjugate transpose. Then, decompose Q to get frame V. The expression is as follows:

$$Q = VV^H. \tag{13}$$

Obtain the joint support set S according to the sparse reconstruction of the vector V. The greedy matching pursuit algorithm can find the index set S where the signal exists. Using the sampling signal and the observation matrix, the time-domain waveform $z_s[n]$ of each channel in the baseband can be obtained. The expression is as follows:

$$\widehat{z}_i[n] = \begin{cases} A_i^{\dagger}y[n] & i \in S, \\ 0 & i \notin S, \end{cases} \tag{14}$$

where $(\bullet)^{\dagger}$ represents the pseudoinverse. The reconstructed signal $\widehat{x}$[n] can be finally obtained. The time domain expression is as follows:

$$\widehat{x}[n] = \widehat{x}\left[nT_{\mathrm{NYQ}}\right] = \sum_{i\in S}\left(\widetilde{z}_i[n] * h_i[n]\right)e^{j2\pi \mathrm{if}_p nT_{\mathrm{NYQ}}}. \tag{15}$$

h[n] is an ideal interpolation filter, and the rate is the Nyquist sampling rate f_{NYQ}. The greedy algorithm part will be studied below.

3. BOMP

This section proposes a backtracking orthogonal match pursuit (BOMP) algorithm. The algorithm mainly improves the OMP algorithm by three parts: threshold filtering, maximum selection, and backtracking idea. First, threshold filtering is used to calculate the inner product of the residual and the matrix. Filter out all atoms whose inner product is greater than the threshold u_t and add these atoms to the support set. Secondly, the backtracking idea is used to find the least square solution using the support set. Through backtracking idea, only some atoms in the support set are selected to form a new support set. Finally, use the maximum selection. In order to ensure that the number of atoms' final output is the same as the sparsity, t atoms are selected to form a new support set during the tth backtracking. It is equivalent to only adding one atom for each cycle of support concentration. After the cycle ends, there are only t atoms in the support set. Specific steps are as follows:

As seen in Algorithm 1, we need to enter **A**, y, and K. Find the atom whose inner product is greater than the threshold u_t each time. Combine these atoms with the support set Λ_{t-1} obtained in the previous cycle to obtain a new

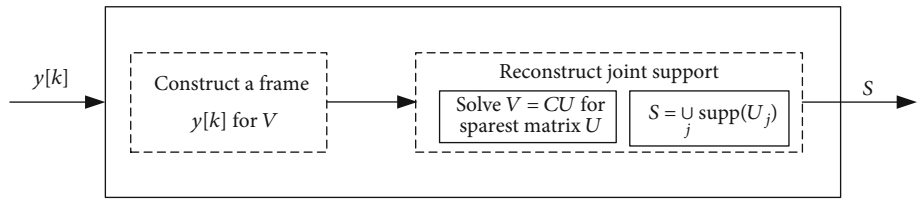

Figure 2: Block diagram of CTF algorithm.

Input: $M \times N$ measurement matrix $\mathbf{A}$
$N \times 1$ sampled measurement vector y
sparsity level K
Output: the estimated support set Λ_t
the nonzero values $\widehat{\theta}_t$
Initialization: $r_0 = y$, $\Lambda_0 = \emptyset$, $\mathbf{A}_0 = \emptyset$, $t = 1$;
(1) Compute inner product $u_t = r_{t-1}, a_j$, find greater than th from u_t, and add the index values j to index set J_0;
(2) $\beta_t = \Lambda_{t-1} \cup J_0$, $\boldsymbol{\alpha}_t = \mathbf{A}_{t-1} \cup a_j (j \epsilon J_0)$
(3) Solve the least squares solution of $y = \boldsymbol{\alpha}_t \widehat{\delta}_t$ and $\widehat{\delta}_t = (\boldsymbol{\alpha}_t^T \boldsymbol{\alpha}_t)^{-1} \boldsymbol{\alpha}_t^T y$;
(4) Replace $\widehat{\theta}_t$、$\mathbf{A}_t$, and Λ_t and find t maximum values from $\widehat{\delta}_t$ denoted by $\widehat{\theta}_t$, $\boldsymbol{\alpha}_t$ number t columns denoted by $\mathbf{A}_t$, and β_t number t columns denoted by Λ_t;
(5) Update residual $r_t = y - \mathbf{A}_t \widehat{\theta}_t = y - \mathbf{A}_t (\mathbf{A}_t^T \mathbf{A}_t)^{-1} \mathbf{A}_t^T y$;
(6) $t = t + 1$; if $t \le K$ return (1); if $t > K$ or $r_t < \varepsilon$, stop iteration.

Algorithm 1: BOMP algorithm.

support set β_t.Use the matrix and residuals corresponding to the support set to find the least squares solution of $\widehat{\delta}_t$. Replace $\widehat{\theta}_t$, $\mathbf{A}_t$, and Λ_t and find the residual r_t according to $\widehat{\theta}_t$ and $\mathbf{A}_t$. Finally, judge whether the iteration is completed through t and r_t. Set ε=10^{-6}. Set the threshold th = $t_s \|r_s\|_2 / \|y\|_2$, where r_s is the residual calculated last time, $\|\bullet\|_2$ is the 2 norms. $t_s \ge 2$ because the found atom needs to be greater than twice the average.

In order to verify the reconstruction probability of the BOMP algorithm, choose OMP, CoSaMP, and gOMP algorithms to compare with the algorithm proposed in this paper. The original signal x uses a Gaussian random signal, and the signal length is $N = 256$. The observation matrix $\boldsymbol{\Phi}$ is a Gaussian random matrix, and the observation value $M = 128$. The interval of sparsity K is [35, 70], and the step size is 5. Each sparsity is measured 1000 times. The same signal is reconstructed by four algorithms. The reconstruction probability under different sparsity is shown in Figure 3.

As shown in Figure 3, when the sparsity is greater than 30, the reconstruction probability of the OMP algorithm begins to gradually decrease. The reconstruction probability of CoSaMP algorithm, gOMP algorithm, and BOMP algorithm is still good. When the sparsity is greater than 50, the reconstruction probability of CoSaMP algorithm, gOMP algorithm, and BOMP algorithm begins to decrease. The reconstruction probability of BOMP algorithm is about 15% higher than that of gOMP algorithm. It can be seen that the BOMP algorithm improves the reconstruction probability. Regarding the reconstruction probability, the CoSaMP algorithm eliminates the wrong atom every time the atom

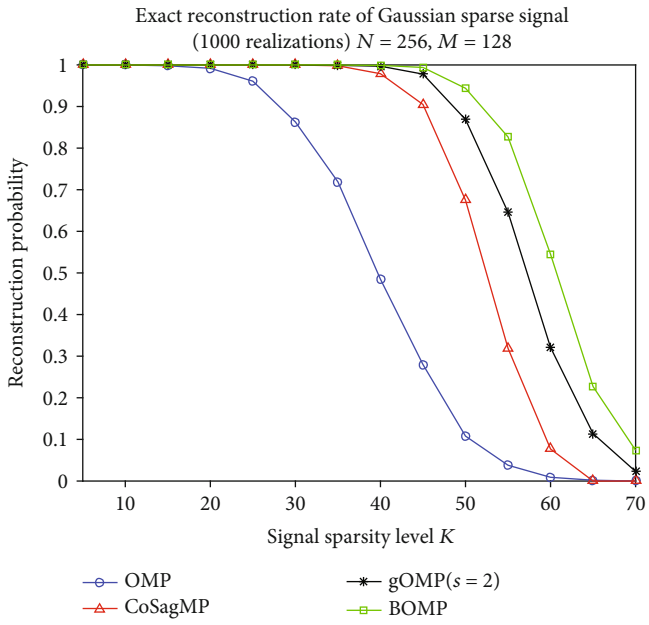

Figure 3: Reconstruction probability diagram under different sparsity.

is selected to prevent the wrong atom from affecting the selection of subsequent atoms, so the reconstruction probability is better than the OMP algorithm. The gOMP algorithm selects multiple atoms each time. According to the conditions for stopping iteration, it can be known that theoretically at most SK atoms can be selected. The correct atom

Input: $M \times N$ measurement matrix $\mathbf{A}$
$N \times 1$ sampled measurement vector y
Output: sparsity level t
the estimated support set Λ_t
the nonzero values $\widehat{\theta}_t$
Initialization: $i = 0, n = <\log_2 M - 1>$ ($<\bullet>$is round), $K_0 = 2^{n-1}$
(1) $i = i + 1$, $r_0 = y$, $\Lambda_0 = \emptyset$, $A_0 = \emptyset$, $t = 1$;
(2) Compute inner product $u_t = r_{t-1}, a_j$, find greater than th from u_t, and add the index values j to index set J_0;
(3) $\beta_t = \Lambda_{t-1} \cup J_0, \boldsymbol{\alpha}_t = \mathbf{A}_{t-1} \cup a_j (j \epsilon J_0)$; if $\beta_t = \Lambda_{t-1}$, stop iteration; otherwise, solve the least squares solution of $y = \boldsymbol{\alpha}_t \widehat{\delta}_t, \widehat{\delta}_t = (\boldsymbol{\alpha}_t^T \boldsymbol{\alpha}_t)^{-1} \boldsymbol{\alpha}_t^T y$;
(4) Replace $\widehat{\theta}_t$、$\mathbf{A}_t$, and Λ_t and find t maximum values from $\widehat{\delta}_t$ denoted by $\widehat{\theta}_t$, $\boldsymbol{\alpha}_t$ number t columns denoted by $\mathbf{A}_t$, and β_t number t columns denoted by Λ_t;
(5) Update residual $r_t = y - \mathbf{A}_t \widehat{\theta}_t = y - \mathbf{A}_t (\mathbf{A}_t^T \mathbf{A}_t)^{-1} \mathbf{A}_t^T y$;
(6) $t = t + 1$; if $t \le K_0$ return (2); if $t > K_0$ or $r_t < \varepsilon$ proceed to (7);
(7) If $t < K_0$, stop iteration; if $i = n + 1$, stop iteration; if $i = n$, proceed to (8); if $t = K_0$ proceed to (9);
(8) If $r_t > \varepsilon$, $K_0 = K_0 + 1$, proceed to (1); otherwise, $K_0 = K_0$, stop iteration;
(9) If $r_t > \varepsilon$, $K_0 = K_0 + 2^{n-1-i}$, proceed to (1); otherwise, $K_0 = K_0 - 2^{n-1-i}$, proceed to (1).

Algorithm 2: DSAMP algorithm.

set is a subset of the supporting set, thus further improving the probability of reconstruction. For the BOMP algorithm, it is finally achieved by selecting the best atom while continuously removing the wrong atom. Improve the probability of reconstruction.

4. DSAMP

This paper proposes a dichotomy-based sparsity adaptive matching pursuit (DSAMP) algorithm for compressed sensing broadband receivers. The algorithm mainly contains two parts. The first part is a fast atom search based on dichotomy. First, select the number of sparsity is 2^{n-1}, in order to achieve sparse coverage under high reconstruction probability. The number of atoms selected for the first time is n. After the preestimated sparsity, the unknown sparsity problem becomes a known sparsity problem. Perform the residual calculation on the basis of this sparsity. Use criteria to determine the calculation results, then use the dichotomy to change the preestimated sparsity according to the evaluation results. The iteration is repeated until the dichotomy satisfies the stop iteration condition or the residual calculation of the nth atomic selection is completed. The sparsity when the iteration is stopped is the sparsity finally estimated by the algorithm. Find the true sparsity through this method. The second part is the BOMP algorithm, which guarantees that the reconstruction algorithm under a given sparsity has a high reconstruction probability. At the same time, it can ensure that the sparsity is the same as the number of atoms in the support set.

Proposition 1. *For the greedy matching pursuit algorithm, the estimated sparsity K_0, the real sparsity K, when $K_0 < K$ and $K_0 \in K$, the residual $r_t > \varepsilon$; when $K_0 \ge K$ and $K \in K_0$, the residual $r_t \le \varepsilon$.*

Proof. According to the principle of the greedy matching pursuit algorithm, r_t is monotonically decreasing when the atom selection is correct in the iteration process, and then according to the iteration stop condition, it is obvious that the proposition is correct.

In Algorithm 2, for the recovery algorithm, the number of channels M must be at least twice the sparsity K to recover the original signal with high probability. Therefore, limit the sparsity range of the algorithm $[0, 2^n]$, where $n = <\log_2 M - 1>$ ($<\bullet>$is round). In the first cycle, the number of atoms is set to the midpoint of the entire range, which is 2^{n-1}. Steps (2)~(6) are BOMP algorithm, and it is mainly to calculate the residual under the current sparsity and iterate under the set sparsity. Steps (7)~(9) are to set K_0, and use Proposition 1 to set K_0. $t < K_0$ indicates that the algorithm has completed convergence ahead of time. The real sparsity is less than the sparsity; at this time, the number of cycles is the estimated sparsity. When $i = n+1$, it means that the last calculation is completed, the estimated sparsity and support set are found, and the iteration can be stopped. When $i = n$, the sparsity K_0 has been selected for the last time. Satisfying $r_t < \varepsilon$ indicates that the sparsity is the estimated sparsity. It can directly output all residuals, sparsity, and support set. When $r_t < \varepsilon$ is not satisfied, K_0 plus 1 is the estimated sparsity. It is necessary to recalculate the residual, sparsity, and support set under the new sparsity. When $t = K_0$, the algorithm has iterated K_0 times. $r_t > \varepsilon$ indicates that the algorithm has not reached the condition to stop iteration. It needs to update the estimated sparsity $K_0 = K_0 + 2^{n-1-i}$ and continue to iterate. In other cases, the iteration continues after updating the estimated sparsity $K_0 = K_0 - 2^{n-1-i}$.

5. Simulation Results

In this section, the performance of the algorithm is verified through simulation of the algorithm. This article simulates the algorithm in two situations. The first is to compare the performance of the algorithm proposed in this paper with several other algorithms in the case of one-dimensional

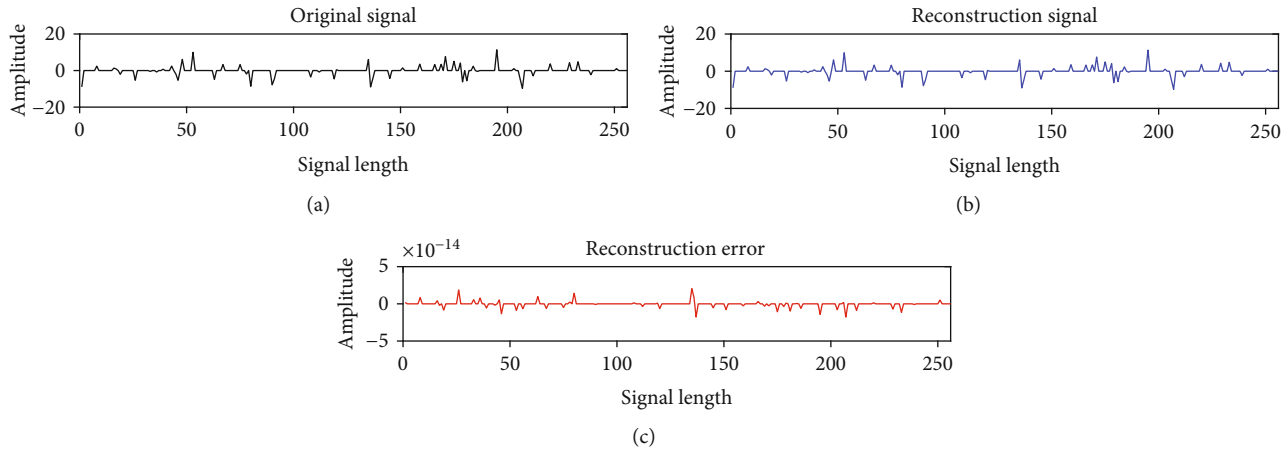

Figure 4: One-dimensional signal reconstruction of BSAMP.

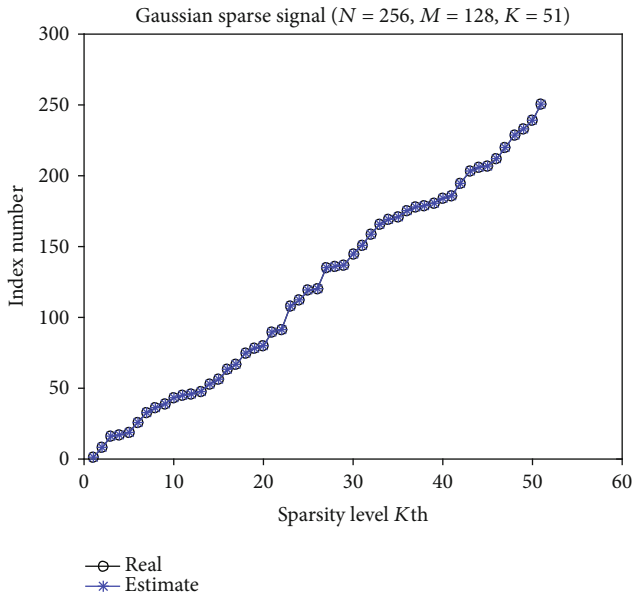

Figure 5: Sparse location diagram.

signals, mainly to verify the accuracy and running time of the algorithm for the estimation of reconstruction probability and sparsity. The other is to simulate under radar signals, which mainly verifies that the algorithm can adapt to multiple sparse signals under low signal-to-noise ratio and finally simulates the signal-to-noise ratio of several algorithms.

5.1. One-Dimensional Signal Simulation. This experiment mainly verifies the correctness of the one-dimensional signal, including signal reconstruction error and sparsity estimation. The signal x is a Gaussian random signal, the signal length is $N = 256$, the observation value is $M = 128$, and the sparsity is $K = 51$. The sparse locations are randomly selected. The observation matrix $\boldsymbol{\Phi}$ is a Gaussian random matrix. Figure 4 is a one-dimensional signal reconstruction diagram of the DSAMP algorithm, including the original signal, reconstructed signal, and reconstruction error. Figure 5 is a diagram of the true position and estimated position of the sparse signal in Figure 4.

As shown in Figure 4, it can be seen that the error between the reconstructed signal and the original signal is $\varepsilon_r < 1 \times 10^{-13}$. It can be seen that the algorithm can accurately reconstruct the original signal.

Figure 5 shows the position and estimated position of the sparse signal when the sparsity is $K = 51$, respectively. It can be seen that the estimated sparsity is the same as the real sparsity of the signal, and the estimated position and the real position are also the same. The algorithm can accurately estimate the sparsity and the location of sparsity.

5.2. Sparsity Estimation Probability Diagram. In this experiment, it is mainly compared with the sparsity adaptive algorithms such as StOMP, SWOMP, and SAMP to illustrate the advantages of the algorithm proposed in this paper. The signal x is a Gaussian random signal, the signal length is $N = 256$, the observation value is $M = 128$, the interval of sparsity K interval is [1, 66], and the step size is 5. The sparse locations are randomly selected. The observation matrix $\boldsymbol{\Phi}$ is a Gaussian random matrix. The sparse position is randomly selected and performs 1000 Monte Carlo experiments at each sparsity. The matrix is a Gaussian random matrix. Figure 6 is the reconstruction probability diagram of 1000 Monte Carlo experiments under different sparsity. Figure 7 is the average sparsity estimation diagram of 1000 Monte Carlo experiments under different sparsity. Figure 8 is the average running time diagram of 1000 Monte Carlo experiments under different sparsity.

As shown in Figure 6, when the sparsity is greater than 40, the reconstruction probability of the StOMP algorithm and the SWOMP algorithm begins to decrease. The reconstruction probability of SAMP algorithm and DSAMP algorithm is still 100%. When the sparsity is greater than 56, the StOMP and SWOMP algorithms cannot be reconstructed. The reconstruction probability of SAMP algorithm and DSAMP algorithm begins to decline, but the reconstruction probability of SAMP algorithm and DSAMP algorithm is similar.

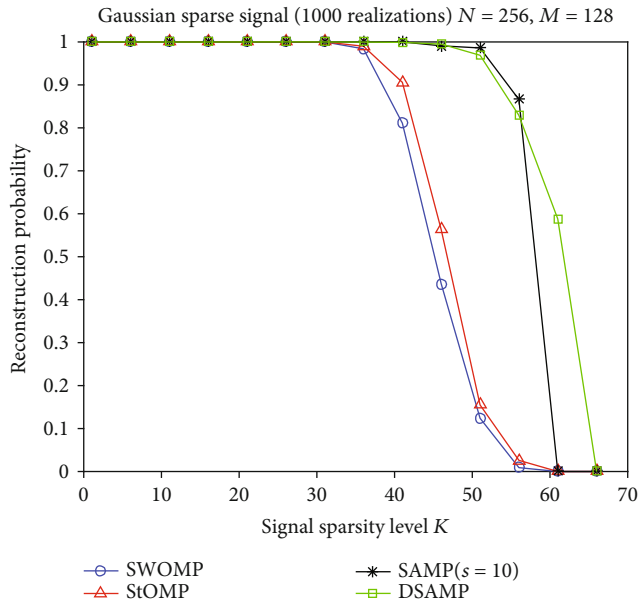

Figure 6: Reconstruction probability diagram under different sparsity.

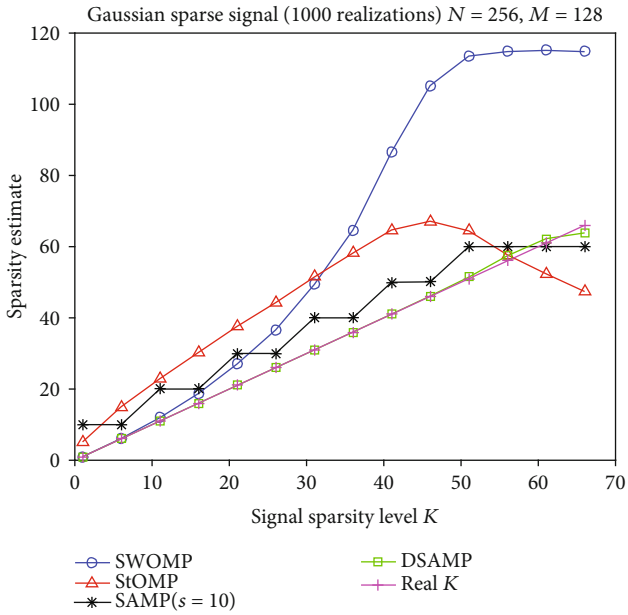

Figure 7: Estimated sparsity diagram under different sparsity.

As shown in Figure 7, the real K represents the real sparsity set, and the remaining four lines represent the corresponding algorithms. When the sparsity is less than 45, the SWOMP algorithm can accurately estimate the sparsity when the sparsity is small, and as the sparsity increases, the error of the estimated sparsity number gradually increases. The number of sparsities estimated by the StOMP algorithm changes with the number of real sparsities, but there is an error with the real sparsity. When the sparsity is greater than 45, the sparsity estimated by the SWOMP algorithm has exceeded 100, which is relatively close to the observed value M. When the sparsity is greater than 45, the amount of sparsities estimated by the StOMP algorithm continues to decrease. From Figure 6 can be seen that the reconstruction probability of these two algorithms is decreasing under this sparsity. When the sparsity is less than 51, due to the setting of the step size of the SAMP algorithm, the amount of sparsity estimates each time is an integer multiple of 10. The DSAMP algorithm can accurately estimate the number of signal sparsity. When the sparsity is greater than 51, the number of sparsity estimated by SAMP algorithm is not changed, and the sparsity estimated by DSAMP algorithm also has errors. It can be seen from Figure 6 that the reconstruction probability of these two algorithms is rapidly decreasing under this sparsity.

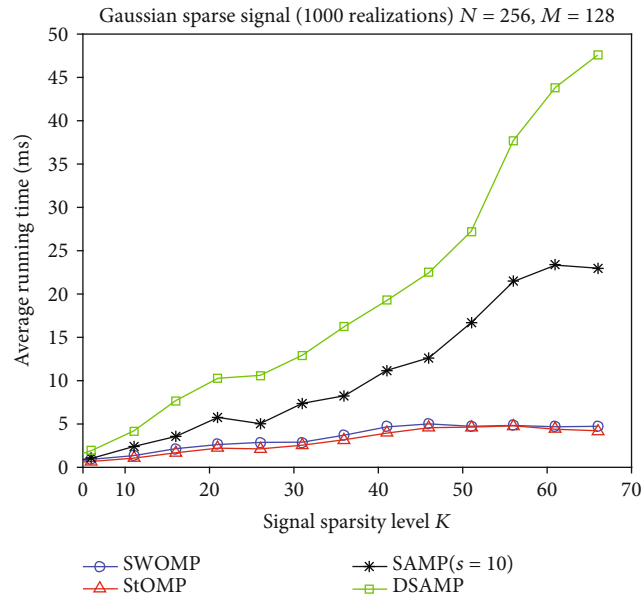

Figure 8: Average running time graph under non-sparseness.

As shown in Figure 8, the average running time of the StOMP algorithm and the SWOMP algorithm is relatively short, the average running time of the DSAMP algorithm is the longest, and the average running time of the SAMP algorithm is between several algorithms. This is because the StOMP algorithm and the SWOMP algorithm can select multiple atoms in each iteration and can quickly complete the iteration. The SAMP algorithm can select multiple atoms in each iteration, but the iterative atoms are repeatedly selected in the iteration, so the average running time is longer than the StOMP algorithm and the SWOMP algorithm. The DSAMP algorithm only adds one atom per iteration, so the average running time is the longest.

5.3. Radar Signal Simulation. In this section of the experiment, radar signals and MWC compressed sampling wideband digital receiver are used to simulate and verify the algorithm proposed in this paper. Set the Nyquist sampling frequency $f_{NYQ} = 20$ GHz of the signal, and the frequency of the signal is $f_1 = 1.1$ GHz, $f_2 = 3.37$ GHz, and $f_3 = 5.13$ GHz, the signal 1 is a linear frequency modulation signal, the frequency modulation width is 50 MHz, signals 2 and 3 are regular signals, the pulse width of the signal is $t = 1.02$

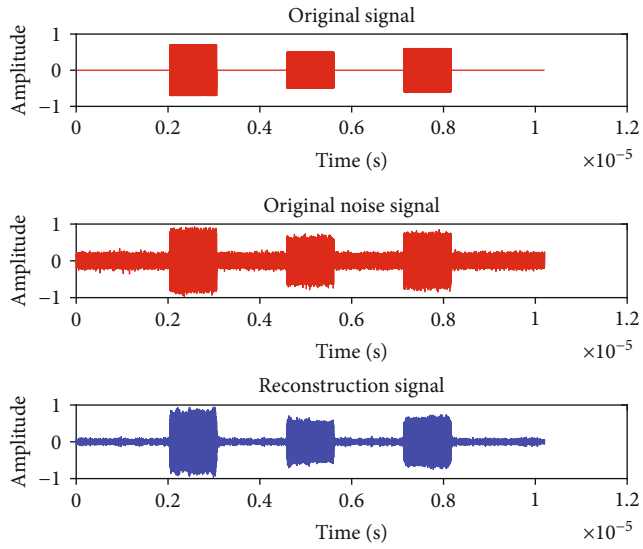

Figure 9: Radar signal reconstruction diagram.

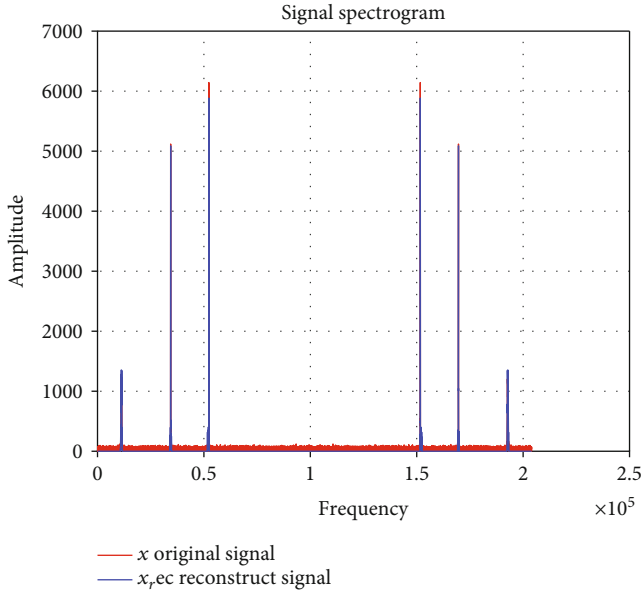

Figure 10: Radar signal spectrum diagram.

us, the noise is Gaussian white noise, the signal-to-noise ratio SNR = 10 dB, and the signal amplitude is $A_1 = 0.7$, $A_2 = 0.5$, and $A_3 = 0.6$. The parameter settings of the MWC compressed sampling broadband digital receiver: the number of sampling channels is $M = 128$, the periodic pseudorandom sequence adopts a Bernoulli pseudorandom sequence of ±1, the period length is $M_P = 255$, the signal sampling time is $T = 10.2$ us, and the original sampling point number is $N = 204000$ points. Figure 9 is the radar signal reconstruction diagram of the compressed sampling broadband receiver of MWC by the DSAMP algorithm, including the original signal, the noise signal, and the reconstructed signal. Figure 10 is the corresponding spectrogram.

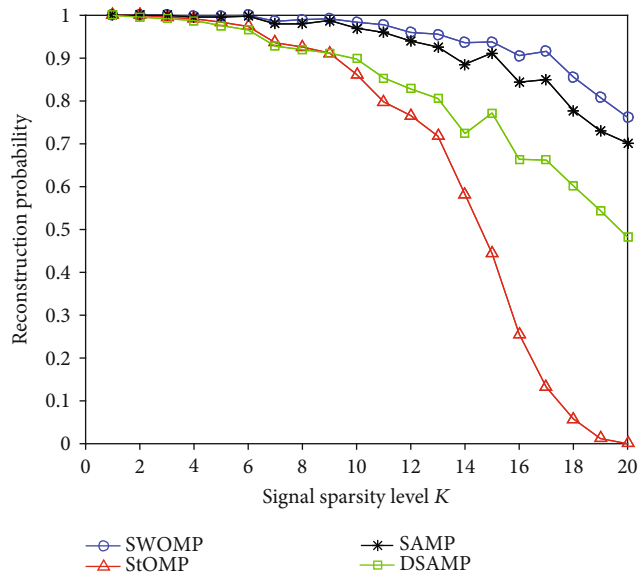

Figure 11: Reconstruction probability diagram under different sparsity.

As shown in Figure 9, the algorithm can estimate 3 signals and reduce the noise of the output signal in the reconstruction process. As shown in Figure 10, the algorithm can accurately estimate the position of the 3 signals. Since the real signal is used in the simulation, the same spectrum will appear in the position of the symmetrical spectrum. It can be seen that the spectrum on the far left and right is the LFM signal. The noise of the four reconstructed signals in the middle is higher than that of the noise-added signal, and all noises beyond the sparsity are eliminated. Therefore, the algorithm has a certain effect on the improvement of signal-to-noise ratio.

In order to prevent multiple signals from appearing in a channel at the same time, the frequencies of the signals are equally spaced. The initial frequency of the signal is, respectively, $f_0 = 1.1555$ GHz, the interval between each signal is 0.315 GHz, the interval of the number of signals is [1, 20], and the step is one. The signal amplitude of is all set to $A_1 = 0.7$. Perform 1000 Monte Carlo experiments under each signal-to-noise ratio. Other conditions are the same as in Figure 9. Figure 11 is a reconstruction probability diagram of different numbers of signals. Figure 12 is an RMSE diagram of sparsity. The simulations use root mean square error (RMSE) to analyze the sparsity estimation performance, expressed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(K_i - K)^2}, \tag{16}$$

where N is the times of independent Monte Carlo simulations, K_i denotes the sparsity estimation, and K denotes real sparsity.

Figures 11 and 12 are the results of the same experiment. The basis for judging the successful reconstruction of the

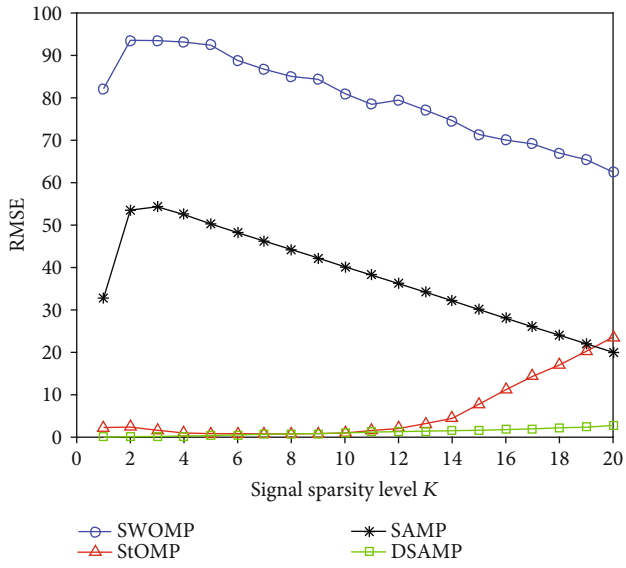

Figure 12: Sparsity estimation RMSE diagram.

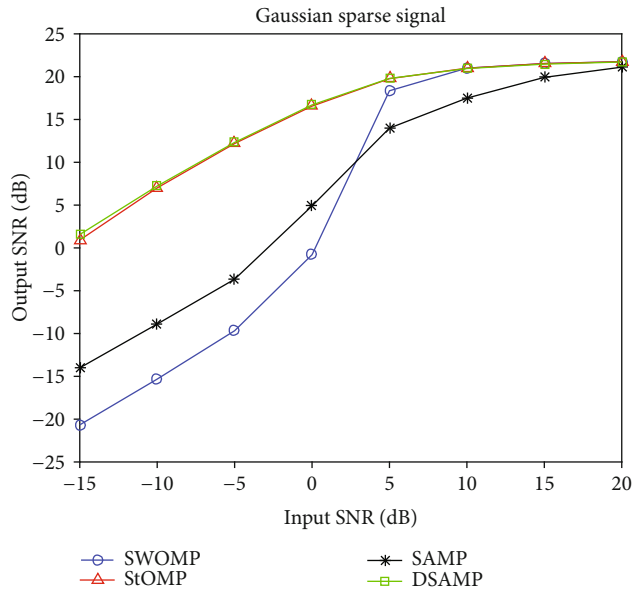

Figure 13: Simulation diagram of radar signal SNR.

algorithm in Figure 11 is the intersection of the estimated sparsity position and the real position. If the intersection is the same as the real position, the reconstruction is considered successful. It can be seen from Figure 11 that the reconstruction probability of SWOMP algorithm and SAMP algorithm is better than DSAMP. The reconstruction probability of StOMP algorithm drops sharply after the number of signals is greater than 10. It can be seen from Figure 12 that the RMSE of the SWOMP algorithm and the SAMP algorithm is very large. It can be seen from Figure 12 that the RMSE of the SWOMP algorithm and the SAMP algorithm is very large. After the number of signals is greater than 3, it decreases. The sparsity estimated by the SWOMP algorithm is close to the observed value M. When the number of signals in the SAMP algorithm is greater than 2, the estimated sparsity is all 60. The RMSE of the StOMP algorithm rises sharply after the number of signals is greater than 10. The RMSE of the DSAMP algorithm has been kept within 5. The SWOMP algorithm and SAMP algorithm can still maintain a reconstruction probability above 0.7 when the number of signals is 20. This is because the two algorithms continue to add atoms to the atomic set until the stop iteration condition is met. Although it can be reconstructed, it introduces a large number of error atoms. When the number of signals in the StOMP algorithm is greater than 10, the reconstruction probability drops rapidly and the RMSE rises rapidly, indicating that the algorithm begins to fail at this time. The DSAMP algorithm maintains a small RMSE when the reconstruction probability decreases. This is because as the signal sparsity increases, the accuracy of the algorithm sparsity estimation decreases, but the estimated sparsity is still close to the true sparsity.

Set the number of signals to one signal, the frequency of the signal is selected randomly between [0, 10 GHz]. The signal-to-noise ratio SNR interval is [-15, 20], and the step size is 5. Perform 1000 Monte Carlo experiments under each signal-to-noise ratio. Other conditions are the same as in Figure 9. Figure 13 is a simulation diagram of the algorithm SNR.

As shown in Figure 13, the SWOMP algorithm and the SAMP algorithm are not effective in improving the output signal-to-noise ratio under low signal-to-noise ratio conditions, while the DSAMP algorithm and StOMP algorithm have a greater improvement in the output signal-to-noise ratio under low signal-to-noise ratio conditions. It can be seen from Figures 11 and 12 that the SWOMP algorithm and the SAMP algorithm are in the case of low signal-to-noise ratio. When the observation matrix is mixed, the noise of all frequency bands is mixed into the baseband, which causes the noise of the baseband signal to be strengthened, and the number of signals recovered during reconstruction is greater than the sparsity. Therefore, the output SNR is lower than the input SNR under the condition of low SNR. For DSAMP algorithm and StOMP algorithm, the sparsity estimation is more accurate at low SNR. It is equivalent to filtering out all the out-of-band noise of the signal, so the signal-to-noise ratio is greatly improved. For a high signal-to-noise ratio, the accuracy of the sparsity estimation of the four algorithms increases, so the output signal-to-noise ratio is improved. The DSAMP algorithm and the StOMP algorithm have a relatively close output signal-to-noise ratio. This is because the two algorithms have the same threshold setting when selecting the threshold. The DSAMP algorithm can have the advantage of improving the signal-to-noise ratio.

6. Conclusion

Adaptive sparsity estimation is necessary for wideband receivers based on MWC compressed sampling in radar systems. This paper proposes a sparsity adaptive matching pursuit algorithm based on dichotomy. In the algorithm proposed in this paper, there is no need to know the sparsity of the signal in advance. We estimate the sparsity of the signal

in a limited number of iterations through the dichotomy. It can be seen through simulation that an accurate and adaptive estimation of the sparsity of the signal is achieved. The orthogonal matching pursuit based on backtracking improves the accuracy of supporting concentrated atoms and finally improves the reconstruction probability of the algorithm. Due to the algorithm's accurate estimation of a single signal, the noise introduced by error atoms is reduced during reconstruction. Improve the signal-to-noise ratio of the algorithm to the signal. As the sparsity increases, the algorithm can still maintain a better reconstruction probability and more accurate sparsity estimation. In the future, the running time of the algorithm can be improved to reduce the calculation time of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors greatly appreciate the editor and reviewers for their work. This work was financially supported in part by the National Natural Science Foundation of China (Grant Nos. 61671168 and 61801143), in part by the National Natural Science Foundation of Heilongjiang Province (Grant No. LH2020F019), in part by the Aeronautical Science Foundation of China (Grant No. 2019010P6001), and in part by the Fundamental Research Funds for the Central Universities (Grant No. 3072019CF0801).

References

[1] C.-T. Liu, R.-J. Wu, Z.-X. He, X.-F. Zhao, H.-C. Li, and P.-Z. Wang, "Modeling and analyzing interference signal in a complex electromagnetic environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, Article ID 1, 2016.

[2] X. Su, Q. X. Fu, Y. Z. Li, and N. Liu, "The hotspots and development trend of wideband digital channelized receiver technology," *Aerospace Electronic Warfare*, vol. 30, no. 1, pp. 32–35, 2014.

[3] T. Huang, Y. Liu, X. Xu, Y. C. Eldar, and X. Wang, "Analysis of frequency agile radar via compressed sensing," *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6228–6240, 2018.

[4] L. Wan, L. Sun, K. Liu, X. Wang, Q. Lin, and T. Zhu, "Autonomous vehicle source enumeration exploiting non-cooperative UAV in software defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[5] J. Skapa and M. Pola, "Ultra wide band receiver design," in *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 304–308, Kosice, Slovakia, 2016.

[6] T. Chen, L. Z. Liu, and L. M. Guo, "Wideband signal detection based on MWC discrete compressed sampling structure," *Transactions of Nanjing University of Aeronautics and Astronautics*, vol. 34, no. 2, pp. 105–114, 2017.

[7] D. L. Donoho, "Compressed sensing," *IEEE Trans Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[9] L. Wan, Y. Sun, L. Sun, Z. Ning, and J. J. P. C. Rodrigues, "Deep learning based autonomous vehicle super resolution DOA estimation for safety driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[10] E. J. Candes and T. Tao, "Decoding by linear programming," *EEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[11] Z. Shi, C. Zhou, Y. Gu, N. A. Goodman, and F. Qu, "Source estimation using coprime array: a sparse reconstruction perspective," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 755–765, 2017.

[12] R. Li, X. Duan, X. Guo, W. He, and Y. Lv, "Adaptive compressive sensing of images using spatial entropy," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 9059204, 9 pages, 2017.

[13] Q. Cheng, A. Alomainy, and Y. Hao, "Compressive millimeter-wave phased array imaging," *IEEE Access*, vol. 4, pp. 9580–9588, 2017.

[14] Y. H. Wang, X. Li, K. Xu, F. B. Ren, and H. Yu, "Data-Driven Sampling Matrix Boolean Optimization for Energy-Efficient Biomedical Signal Acquisition by Compressive Sensing," *IEEE Transactions on Biomedical Circuits ï¼† Systems*, vol. 11, no. 2, pp. 255–266, 2017.

[15] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: a compressive sensing approach," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 40–58, 2018.

[16] L. Sun, L. T. Wan, and X. P. Wang, "Learning-based resource allocation strategy for industrial IoT in UAV-enabled MEC systems," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[17] C. L. An, Z. Y. Zhao, M. Diao, and H. Y. Gao, "A method for direction finding of uncorrelated and coherent signals coexisted based on arbitrary array," *Journal of Harbin Engineering University*, vol. 34, no. 4, pp. 517–523, 2013.

[18] C. Zhou, Y. Gu, S. He, and Z. Shi, "A robust and efficient algorithm for coprime array adaptive beamforming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1099–1112, 2018.

[19] A. Wadhwa, U. Madhow, and N. R. Shanbhag, "Slicer architectures for analog-to-information conversion in channel equalizers," *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1234–1246, 2017.

[20] W. B. Xu, Y. P. Cui, Y. Wang, S. Y. Wang, and J. R. Lin, "A hardware implementation of random demodulation analog-to-information converter," *ICE Electronics Express*, vol. 13, no. 16, pp. 1–6, 2016.

[21] R. Grigoryan, T. L. Jensen, and T. Larsen, "Computational complexity reduction in nonuniform compressed sensing by multicoset emulation," *Signal Processing*, vol. 131, pp. 492–501, 2017.

[22] F. Xi, S. Chen, Y. D. Zhang, and Z. Liu, "Gridless quadrature compressive sampling with interpolated array technique," *Signal Processing*, vol. 133, pp. 1–12, 2017.

[23] M. Mishali and Y. C. Elder, "From theory to practice: sub-Nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.

[24] K. M. Cohen, C. Attias, B. Farbman, I. Teselniker, and Y. C. Eldar, "Channel estimation in UWB channels using compressed sensing," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1966–1970, Florence, Italy, 2014.

[25] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, "Sub-Nyquist acquisition hardware for wideband communication," in *2010 IEEE Workshop On Signal Processing Systems*, pp. 156–161, San Francisco, CA, USA, 2010.

[26] C. Zhou, Y. Gu, Y. D. Zhang, Z. Shi, T. Jin, and X. Wu, "Compressive sensing-based coprime array direction-of-arrival estimation," *IET Communications*, vol. 11, no. 11, pp. 1719–1724, 2017.

[27] L. Li, Y. Fang, L. Liu, H. Peng, J. Kurths, and Y. Yang, "Overview of compressed sensing: sensing model, reconstruction algorithm, and its applications," *Applied Sciences*, vol. 10, no. 17, p. 5909, 2020.

[28] S. G. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1994.

[29] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[30] D. Neenell and R. Vetshynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[31] D. Neenell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[32] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[33] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 6202–6216, 2012.

[34] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[35] T. Blumensath and M. E. Davies, "Stagewise weak gradient pursuits," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4333–4346, 2009.

[36] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 581–587, Pacific Grove, CA, USA, 2008.

[37] L. Sun, L. Wan, K. Liu, and X. Wang, "Cooperative-evolution-based WPT resource allocation for large-scale cognitive industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5401–5411, 2020.

[38] X. L. D, H. Xing, B. B. Gu, B. Chen, and S. M. Qiu, "SAMP-RB compressed sensing reconstruction algorithm based on variable step," *Application Research of Computers*, vol. 35, no. 4, pp. 1084–1087, 2018.