SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2017, Vol. 60 082302:1–082302:13
doi: 10.1007/s11432-015-1016-x

Fast FOCUSS method based on bi-conjugate gradient and its application to space-time clutter spectrum estimation

Gatai BAI[1,3], Ran TAO[1,2,3*], Juan ZHAO[2,3], Xia BAI[2,3] & Yue WANG[1,2,3]

[1]*School of Mathematics and Statistics, Beijing Institute of Technology, Beijing* 100081*, China;*
[2]*School of Information and Electronics, Beijing Institute of Technology, Beijing* 100081*, China;*
[3]*Beijing Key Laboratory of Fractional Signals and Systems, Beijing* 100081*, China*

Received October 8, 2016; accepted December 29, 2016; published online February 24, 2017

Abstract The focal underdetermined system solver (FOCUSS) is a powerful tool for sparse representation in complex underdetermined systems. This paper presents the fast FOCUSS method based on the bi-conjugate gradient (BICG), termed BICG-FOCUSS, to speed up the convergence rate of the original FOCUSS. BICG-FOCUSS was specifically designed to reduce the computational complexity of FOCUSS by solving a complex linear equation using the BICG method according to the rank of the weight matrix in FOCUSS. Experimental results show that BICG-FOCUSS is more efficient in terms of computational time than FOCUSS without losing accuracy. Since FOCUSS is an efficient tool for estimating the space-time clutter spectrum in sparse recovery-based space-time adaptive processing (SR-STAP), we propose BICG-FOCUSS to achieve a fast estimation of the space-time clutter spectrum in mono-static array radar and in the mountaintop system. The high performance of the proposed BICG-FOCUSS in the application is demonstrated with both simulated and real data.

Keywords focal underdetermined system solver (FOCUSS), sparse recovery (SR), bi-conjugate gradient (BICG), space-time adaptive processing (STAP), space-time clutter spectrum

Citation Bai G T, Tao R, Zhao J, et al. Fast FOCUSS method based on bi-conjugate gradient and its application to space-time clutter spectrum estimation. Sci China Inf Sci, 2017, 60(8): 082302, doi: 10.1007/s11432-015-1016-x

1 Introduction

The problem of computing sparse solutions in an underdetermined system arises in a large number of application areas including the electromagnetic inverse problem [1], direction-of-arrival estimation [2, 3], spectral estimation [4], and radar imaging [5, 6]. In these applications, the underdetermined system can be stated as follows: Represent a signal of interest $\boldsymbol{x}$ using the minimum components of vectors from a measurement matrix $\boldsymbol{\Phi}$ (or set of atoms) as $\boldsymbol{x} = \boldsymbol{\Phi\gamma}$, where $\boldsymbol{x}$ is an $m \times 1$ measurement vector, $\boldsymbol{\Phi}$ is an $m \times n$ $(m < n)$ basis matrix with row full rank, and $\boldsymbol{\gamma}$ is an unknown $n \times 1$ sparse vector. Mathematically, this problem can be written as [7]

$$\min ||\boldsymbol{\gamma}||_0, \quad \text{s.t. } \boldsymbol{x} = \boldsymbol{\Phi\gamma}, \tag{1}$$

* Corresponding author (email: rantao@bit.edu.cn)

where $||\boldsymbol{\gamma}||_0$ denotes the number of nonzero entries of the vector $\boldsymbol{\gamma}$. The sparse vector $\boldsymbol{\gamma}$ satisfies that $||\boldsymbol{\gamma}||_0 \leqslant m$. However, this problem has been shown to be NP-hard [8]. Much research effort has been invested in finding low complexity algorithms such as matching pursuit methods [8, 9], convex minimizations [10, 11], and the focal underdetermined system solver (FOCUSS) [1, 2, 12–15]. The matching pursuit methods find a solution by sequentially adding atoms to a set that is used to represent $\boldsymbol{x}$. Convex minimizations approximate the l_0 minimization in (1) by the l_1 minimization. FOCUSS is based on weighted norm minimization of the dependent variable with the weights being a function of the preceding iterative solutions [2]. FOCUSS is an effective tool for sparse recovery-based space-time adaptive processing (SR-STAP) [4] and tomographic source reconstruction in neural electromagnetic inverse problems [1]. However, the computational complexity of large-scale problems overwhelms FOCUSS, which requires a considerable amount of time to obtain a sparse solution. As an example in SR-STAP [4, 16–21], when the spatial-Doppler plane has a high-resolution requirement, or the number of antennas and the number of pulses are large, the model of the space-time clutter spectrum is a large-scale complex undetermined system. In this case, the computation would be highly time consuming and the instantaneity of target detection in the STAP would be reduced if FOCUSS was used to estimate the space-time clutter spectrum. Therefore, the implementation of FOCUSS requires the development of a fast method.

Rao and Kreutz-Delgado [12] implemented FOCUSS in the form of l_p ($p \leqslant 1$) minimization, in which the Moore Penrose inverse $(\boldsymbol{\Phi}_k)^+$ is replaced by $\boldsymbol{\Phi}_k^{\mathrm{H}}(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}})^{-1}$, where $(\cdot)^+$ and $(\cdot)^{\mathrm{H}}$ represent the Moore Penrose inverse and the conjugate transpose, respectively. $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\boldsymbol{W}_k$, where $\boldsymbol{W}_k$ is the weight matrix. For this form of FOCUSS, the conjugate gradient (CG) method is employed to reduce the computational complexity per iterative cycle when $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\gamma}$ is real [14]. However, when the rank of weight matrix $\boldsymbol{W}_k$ is less than m, the matrix $\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}}$ is irreversible. This would result in errors in computation.

In this paper, we employ the bi-conjugate gradient (BICG) method [22, 23], which is an iterative method for linear equations, to reduce the computational complexity of the original FOCUSS [1, 2] when $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\gamma}$ is complex. Since the main computational burden of the original FOCUSS is the computation of $(\boldsymbol{\Phi}_k)^+\boldsymbol{x}$, we transform the Moore Penrose inverse such that it can be inexpensively solved by the BICG method. In addition, we carried out numerical experiments to illustrate the performance of BICG-based FOCUSS, i.e., BICG-FOCUSS. The experimental results show that the time consumption of BICG-FOCUSS is significantly lower than that of FOCUSS without losing accuracy. Moreover, we employ BICG-FOCUSS in the application of SR-STAP to reduce the time required to estimate the space-time clutter spectrum for mono-static array radar and the mountaintop system. The outcomes show that BICG-FOCUSS can significantly reduce the time required for the computation without any reduction in accuracy in estimating the space-time clutter spectrum.

The paper is organized as follows: In Section 2, we introduce the FOCUSS algorithm, describe the development of the fast FOCUSS method, i.e., BICG-FOCUSS, by using the BICG method, and present the performance in terms of time consumption and the estimating accuracy of BICG-FOCUSS with numerical experiments. In Section 3, we employ BICG-FOCUSS to estimate the space-time clutter spectrum in SR-STAP, and discuss the performance of BICG-FOCUSS with both simulated data and mountaintop data. Section 4 provides a summary of the paper.

2 Fast FOCUSS method based on bi-conjugate gradient

2.1 Review of the FOCUSS

The FOCUSS has been proposed to iteratively solve sparse problems [1, 2]. It uses weighted norm minimization to recursively adjust the weighting matrix until most elements of the solution are close to zero. The detailed steps of the original FOCUSS [1, 2] are listed as Algorithm 1. $(\cdot)^{\mathrm{T}}$ and $||\cdot||_2$ in the algorithm denote the transpose operation and the 2-norm, respectively. The parameter l is the power to which to raise the entries of $\boldsymbol{\gamma}^{(k-1)}$. According to the properties of FOCUSS, for any l ($l > 0.5$), FOCUSS converges to a sparse solution by choosing an appropriate initialization. FOCUSS is computationally inefficient, especially for large-scale problems. The main factor influencing the time consumption is the

Algorithm 1 FOCUSS

Input: basis matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m\times n}$, data $\boldsymbol{x} \in \mathbb{C}^{m\times 1}$
Parameters: parameter l, termination level ς
Initialization: $\boldsymbol{\gamma}^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \ldots, \gamma_n^{(0)}]^{\mathrm{T}}$, $k = 1$
Iteration:
1: $\boldsymbol{W}_k = \mathrm{diag}(|\gamma_1^{(k\text{-}1)}|^l, \ldots, |\gamma_n^{(k\text{-}1)}|^l)$
2: $\boldsymbol{q}^{(k)} = (\boldsymbol{\Phi}_k)^+ \boldsymbol{x}$, where $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\boldsymbol{W}_k$
3: $\boldsymbol{\gamma}^{(k)} = \boldsymbol{W}_k \boldsymbol{q}^{(k)}$
4: break until $||\boldsymbol{\gamma}^{(k)} - \boldsymbol{\gamma}^{(k-1)}||_2 / ||\boldsymbol{\gamma}^{(k)}||_2 < \varsigma$, else let $k = k + 1$ and go to Step 1.
Output: $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(k)}$

computational burden per iteration cycle of FOCUSS. This computational burden mainly consists of the computation of the Moore Penrose Inverse $(\boldsymbol{\Phi}_k)^+$. In the following, we transform the computation of the Moore Penrose inverse into solving a complex linear equation by an iterative method, such that the complexity of the computation of $(\boldsymbol{\Phi}_k)^+\boldsymbol{x}$ can be reduced by one order of magnitude. Here, we make two assumptions, stated as follows.

Assumption 1. $\boldsymbol{\Phi}$ satisfies the unique representation property (URP), i.e., any m columns of $\boldsymbol{\Phi}$ are linearly independent [2].

Assumption 2. The initialization $\boldsymbol{\gamma}^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \ldots, \gamma_n^{(0)}]^{\mathrm{T}}$ are strictly nonzero, i.e., $\gamma_i^{(0)} \neq 0$ for $i = 1, \ldots, n$. It is known that, when $\gamma_i^{(0)} = 0$, the i term of $\boldsymbol{\gamma}^{(k)}$ is zero in every iteration. Assumption 2 ensures that all columns of $\boldsymbol{\Phi}$ participate in the computation of the k-th sparse solution $\boldsymbol{\gamma}^{(k)}$.

2.2 Proposed BICG-FOCUSS

In this section, we transform $(\boldsymbol{\Phi}_k)^+\boldsymbol{x}$ into the form $\boldsymbol{\Psi}\boldsymbol{M}^{-1}\boldsymbol{z}$ by matrix transformation. According to the theory of the iterative method for linear equations, we can choose an iterative method to obtain $\boldsymbol{M}^{-1}\boldsymbol{z}$ from the complex linear equation $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$. The BICG method is chosen as the iterative method because of its low complexity and fast convergence.

2.2.1 *Matrix transformation*

The following two properties of the Moore Penrose inverse are used.

(1) For any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Let $R(\boldsymbol{A})$ be the linear space spanned by the columns of $\boldsymbol{A}$. If $\boldsymbol{A}$ is column full rank and $R(\boldsymbol{B}) \subset R(\boldsymbol{A})$, the Moore Penrose inverse of the new matrix $(\boldsymbol{A}\ \boldsymbol{B})$ is given as [24]

$$(\boldsymbol{A}\ \boldsymbol{B})^+ = \begin{pmatrix} \boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\big[\boldsymbol{A}^{\mathrm{H}}(\mathbf{A}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{B}\boldsymbol{B}^{\mathrm{H}})\mathbf{A}\big]^{-1}\boldsymbol{A}^{\mathrm{H}} \\ \boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}\big[\boldsymbol{A}^{\mathrm{H}}(\mathbf{A}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{B}\boldsymbol{B}^{\mathrm{H}})\mathbf{A}\big]^{-1}\boldsymbol{A}^{\mathrm{H}} \end{pmatrix}. \tag{2}$$

(2) For any matrix $\boldsymbol{A}$, define $\boldsymbol{A}(i, j)$ as the matrix which exchanges the ith-row with the jth-row of $\boldsymbol{A}$ and define $\boldsymbol{A}[i, j]$ as the matrix which exchanges the ith-column with the jth-column of $\boldsymbol{A}$. Then we have

$$(\boldsymbol{A}(i, j))^+ = (\boldsymbol{A})^+[i, j]. \tag{3}$$

Since the sparse solution $\boldsymbol{\gamma}$ satisfies $||\boldsymbol{\gamma}||_0 \leqslant m$, the number of nonzero entries in the diagonal $\boldsymbol{W}_k$ may be less than m. The matrix $\boldsymbol{\Phi}_k$ is the product of the basis matrix $\boldsymbol{\Phi}$ and the weight matrix $\boldsymbol{W}_k$. Then, the rank of the matrix $\boldsymbol{\Phi}_k$ may be less than m. Then, we consider the following two cases.

(1) When $\mathrm{rank}(\boldsymbol{W}_k) \geqslant m$. According to the URP condition for $\boldsymbol{\Phi}$, the matrix $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\boldsymbol{W}_k$ is row full rank. By using the property of the Moore Penrose inverse, we have

$$(\boldsymbol{\Phi}_k)^+\boldsymbol{x} = \boldsymbol{\Phi}_k^{\mathrm{H}}(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}})^{-1}\boldsymbol{x}. \tag{4}$$

This is the form $\boldsymbol{\Psi}\boldsymbol{M}^{-1}\boldsymbol{z}$, where $\boldsymbol{\Psi} = \boldsymbol{\Phi}_k^{\mathrm{H}}$, $\boldsymbol{M} = \boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}}$, and $\boldsymbol{z} = \boldsymbol{x}$. Therefore, the computation of $(\boldsymbol{\Phi}_k)^+\boldsymbol{x}$ can be obtained by the product of $\boldsymbol{\Psi}$ and $\boldsymbol{M}^{-1}\boldsymbol{z}$, where $\boldsymbol{M}^{-1}\boldsymbol{z}$ can be obtained from $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$ by iterative method.

(2) When $\mathrm{rank}(\boldsymbol{W}_k) < m$. For convenience, we assume $\mathrm{rank}(\boldsymbol{W}_k) = r$. Then there are r nonzero elements in $\boldsymbol{\gamma}^{(k-1)}$ in FOCUSS. Since $\mathrm{rank}(\boldsymbol{\Phi}) = m$, according to the theory of the SVD (i.e., singular value decomposition), there are two unitary matrices $\boldsymbol{U} \in \mathbb{C}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{C}^{n\times n}$ satisfying

$$\boldsymbol{\Phi} = \boldsymbol{U}\,(\Sigma\ \boldsymbol{0})\,\boldsymbol{V}^{\mathrm{H}}, \tag{5}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m)$, and $\sigma_i > 0$ $(1 \leqslant i \leqslant m)$. Thus, we have

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\boldsymbol{W}_k = \boldsymbol{U}\,(\Sigma\ \boldsymbol{0})\,\boldsymbol{V}^{\mathrm{H}}\boldsymbol{W}_k. \tag{6}$$

Let $\boldsymbol{V} = [\ \boldsymbol{V}_1^{\mathrm{H}}, \boldsymbol{V}_2^{\mathrm{H}}]$, where the size of $\boldsymbol{V}_1$ is $m \times n$. Then Eq. (6) can be written as

$$\boldsymbol{\Phi}_k = \boldsymbol{U}\Sigma\boldsymbol{V}_1\boldsymbol{W}_k. \tag{7}$$

According to the property of the Moore Penrose inverse, $(\boldsymbol{\Phi}_k)^{+}\boldsymbol{x}$ is the minimum 2-norm solution of $\boldsymbol{\Phi}_k\boldsymbol{y} = \boldsymbol{x}$, i.e.,

$$(\boldsymbol{\Phi}_k)^{+}\boldsymbol{x} = \min_{\boldsymbol{\Phi}_k\boldsymbol{y}=\boldsymbol{x}} \|\boldsymbol{y}\|_2. \tag{8}$$

Since $\boldsymbol{U}$ is a unitary matrix and Σ is a diagonal matrix, $\boldsymbol{\Phi}_k\boldsymbol{y} = \boldsymbol{x}$ is identical with $(\boldsymbol{V}_1\boldsymbol{W}_k)\boldsymbol{y} = \hat{\boldsymbol{x}}$, where $\hat{\boldsymbol{x}} = \Sigma^{-1}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{x}$. Then Eq. (8) is equivalent to the equation

$$(\boldsymbol{\Phi}_k)^{+}\boldsymbol{x} = \min_{(\boldsymbol{V}_1\boldsymbol{W}_k)\boldsymbol{y}=\hat{\boldsymbol{x}}} \|\boldsymbol{y}\|_2. \tag{9}$$

Since $(\boldsymbol{V}_1\boldsymbol{W}_k)^{+}$ is the minimum 2-norm solution of $(\boldsymbol{V}_1\boldsymbol{W}_k)\boldsymbol{y} = \hat{\boldsymbol{x}}$, we know that

$$\min_{(\boldsymbol{V}_1\boldsymbol{W}_k)\boldsymbol{y}=\hat{\boldsymbol{x}}} \|\boldsymbol{y}\|_2 = (\boldsymbol{V}_1\boldsymbol{W}_k)^{+}\hat{\boldsymbol{x}}. \tag{10}$$

Thus we have

$$(\boldsymbol{\Phi}_k)^{+}\boldsymbol{x} = (\boldsymbol{V}_1\boldsymbol{W}_k)^{+}\hat{\boldsymbol{x}}. \tag{11}$$

Assume that $\boldsymbol{V}_1 = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$. Let $\boldsymbol{V}_{1k} = \boldsymbol{V}_1\boldsymbol{W}_k$. Since $\boldsymbol{W}_k$ is a diagonal matrix as in Algorithm 1, $\boldsymbol{V}_{1k}$ can be written as

$$\boldsymbol{V}_{1k} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n] \cdot \mathrm{diag}(|\gamma_1^{(k\text{-}1)}|^l, \ldots, |\gamma_n^{(k\text{-}1)}|^l) = \left[|\gamma_1^{(k-1)}|^l\boldsymbol{v}_1, |\gamma_2^{(k-1)}|^l\boldsymbol{v}_2, \ldots, |\gamma_n^{(k-1)}|^l\boldsymbol{v}_n\right]. \tag{12}$$

Since $\mathrm{rank}(\boldsymbol{W}_k) = r$, $(n-r)$ columns of the matrix $\boldsymbol{V}_{1k}$ are zero vectors. Then, exchange the columns of $\boldsymbol{V}_{1k}$ such that $\boldsymbol{V}_{1k}$ is transformed into a new matrix as

$$\tilde{\boldsymbol{V}}_{1k} = [\tilde{\boldsymbol{V}}_k\ \boldsymbol{0}_{m\times(n-r)}], \tag{13}$$

where the size of $\tilde{\boldsymbol{V}}_k$ is $m \times r$. Then $\mathrm{rank}(\tilde{\boldsymbol{V}}_k) = r$. According to the theory of linear algebra, $\tilde{\boldsymbol{V}}_{1k}$ in (13) can be obtained by multiplying an elementary matrix $\boldsymbol{T}$ to the right of $\boldsymbol{V}_{1k}$. Then Eq. (13) can be written as

$$\tilde{\boldsymbol{V}}_{1k} = \boldsymbol{V}_{1k}\boldsymbol{T}. \tag{14}$$

Our aim here is to derive $(\boldsymbol{V}_{1k})^{+}$ from $(\tilde{\boldsymbol{V}}_{1k})^{+}$. Since $R(\boldsymbol{0}_{m\times(n-r)}) \subset R(\tilde{\boldsymbol{V}}_k)$, it is known from (2) that $(\tilde{\boldsymbol{V}}_{1k})^{+}$ can be computed as

$$(\tilde{\boldsymbol{V}}_{1k})^{+} = \begin{pmatrix} \tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k[(\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k)^2]^{-1}\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}} \\ \boldsymbol{0} \end{pmatrix}. \tag{15}$$

Since $\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k[(\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k)^2]^{-1} = (\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k)^{-1}$, Eq. (15) can be written as

$$(\tilde{\boldsymbol{V}}_{1k})^{+} = \begin{pmatrix} (\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k)^{-1}\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}} \\ \boldsymbol{0} \end{pmatrix}. \tag{16}$$

Algorithm 2 BICG

Input: $\boldsymbol{M} \in \mathbb{C}^{n\times n}$, vector $\boldsymbol{z} \in \mathbb{C}^{n\times 1}$, tolerance ε

Initialization: Initial vector $\boldsymbol{y}_0 \in \mathbb{C}^{n\times 1}$. Let $\boldsymbol{r}_0 = \boldsymbol{z} - \boldsymbol{M}\boldsymbol{y}_0$, $\tilde{\boldsymbol{r}}_0 = \boldsymbol{r}_0^*$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$, $\tilde{\boldsymbol{p}}_0 = \tilde{\boldsymbol{r}}_0$, and $k = 1$

Step 1: Calculate

$\alpha_k = \mathrm{Re}(\langle \boldsymbol{r}_k, \tilde{\boldsymbol{r}}_k\rangle)/\mathrm{Re}(\langle \boldsymbol{M}\boldsymbol{p}_k, \tilde{\boldsymbol{p}}_k\rangle)$, $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \alpha_k \boldsymbol{p}_k$,

$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{M}\boldsymbol{p}_k$, $\tilde{\boldsymbol{r}}_{k+1} = \tilde{\boldsymbol{r}}_k - \alpha_k \boldsymbol{M}^{\mathrm{H}}\tilde{\boldsymbol{p}}_k$

$\beta_k = \mathrm{Re}(\langle \boldsymbol{r}_{k+1}, \tilde{\boldsymbol{r}}_{k+1}\rangle)/\mathrm{Re}(\langle \boldsymbol{r}_k, \tilde{\boldsymbol{r}}_k\rangle)$.

$\boldsymbol{p}_{k+1} = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_k$, $\tilde{\boldsymbol{p}}_{k+1} = \tilde{\boldsymbol{r}}_k + \beta_k \tilde{\boldsymbol{p}}_k$

Step 2: If $||\boldsymbol{r}_{k+1}||_2/||\boldsymbol{z}\,||_2 < \varepsilon$, break and output y $= \boldsymbol{y}_{k+1}$; else let $k = k + 1$ and turn to Step 1.

The function of the elementary matrix $\boldsymbol{T}$ in (14) is only to exchange the columns of $\boldsymbol{V}_{1k}$. Then, according to (3), $(\boldsymbol{V}_{1k})^+$ is obtained by multiplying $\boldsymbol{T}$ with the left-hand side of $(\tilde{\boldsymbol{V}}_{1k})^+$, i.e.,

$$(\boldsymbol{V}_{1k})^+ = \boldsymbol{T}(\tilde{\boldsymbol{V}}_{1k})^+. \tag{17}$$

Then, we derive

$$(\boldsymbol{V}_{1k})^+\hat{\boldsymbol{x}} = (\boldsymbol{V}_1\boldsymbol{W}_k)^+\hat{\boldsymbol{x}} = \boldsymbol{T}(\tilde{\boldsymbol{V}}_{1k})^+\hat{\boldsymbol{x}}. \tag{18}$$

According to (11), (16), and (18), we obtain that

$$(\boldsymbol{\Phi}_k)^+\boldsymbol{x} = \boldsymbol{T}\begin{pmatrix} \boldsymbol{H}_k{}^{-1}\tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\hat{\boldsymbol{x}} \\ \boldsymbol{0} \end{pmatrix}, \tag{19}$$

where $\boldsymbol{H}_k = \tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k$ and its size is $r \times r$. Since $\boldsymbol{T}$ in (19) is only a matrix of row exchange, the right-hand side of (19) can be regarded as the form $\boldsymbol{\Psi}\boldsymbol{M}^{-1}\boldsymbol{z}$, where $\boldsymbol{\Psi} = \boldsymbol{I}$, $\boldsymbol{M} = \boldsymbol{H}_k$, and $\boldsymbol{z} = \tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\hat{\boldsymbol{x}}$. Thus, the computation of $\boldsymbol{\Phi}_k{}^+\boldsymbol{x}$ can be obtained by the product of $\boldsymbol{\Psi}$ and $\boldsymbol{M}^{-1}\boldsymbol{z}$, where $\boldsymbol{M}^{-1}\boldsymbol{z}$ can be obtained from $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$ by the iterative method.

It can be seen that the computation of $\boldsymbol{\Phi}_k{}^+\boldsymbol{x}$ is transformed into the computation of $\boldsymbol{\Psi}\boldsymbol{M}^{-1}\boldsymbol{z}$ when $\mathrm{rank}(\boldsymbol{W}_k) \geqslant m$ and $\mathrm{rank}(\boldsymbol{W}_k) < m$.

2.2.2 *Bi-conjugate gradient*

According to the above subsection, we can develop a type of fast FOCUSS method in which the computation of $\boldsymbol{M}^{-1}\boldsymbol{z}$ is achieved by iterative methods.

It can easily be verified that $\boldsymbol{M}$ is a positive-definite Hermite matrix irrespective of when $\mathrm{rank}(\boldsymbol{W}_k) \geqslant m$ and when $\mathrm{rank}(\boldsymbol{W}_k) < m$. In many iterative methods, the classical successive over-relaxation (SOR) can converge to the true solution of $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$ [25] according to its convergent condition. Then there exists a SOR-based FOCUSS, i.e., SOR-FOCUSS, in which $\boldsymbol{M}^{-1}\boldsymbol{z}$ is obtained by the SOR method. However, its convergent rate is low and it is difficult to determine the optimal relax factor ω because it relates to the spectral radius of $\boldsymbol{M}$. The conjugate gradient (CG) method is a fast iterative method, which is independent of any parameter, but it is only feasible for a linear equation with a positive-definite symmetrical coefficient matrix. For general real linear equations, Fletcher [22] proposed a generalization of the CG, i.e., the BICG method. Iteratively, BICG converges to the solution by minimizing the residual. Similar to the CG method, BICG is a most useful and computationally inexpensive technique for solving complex linear equations and the convergent rate of BICG is high according to the analysis in [26]. $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$ is a complex linear equation, and then we use the complex-version of the BICG [23], which has ability to process linear equations with a complex coefficient matrix, to obtain $\boldsymbol{M}^{-1}\boldsymbol{z}$ from $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$. The detailed steps of BICG are illustrated in the following as Algorithm 2.

In the algorithm, $(\cdot)^*$, $\mathrm{Re}(\cdot)$, and $\langle\cdot,\cdot\rangle$ represent the conjugate operator, the real part of $(\cdot)$, and the inner product in the unitary space, respectively. Since the matrix $\boldsymbol{M}$ is the Hermite matrix, i.e., $\boldsymbol{M} = \boldsymbol{M}^{\mathrm{H}}$, the computation of the conjugate transpose $\boldsymbol{M}^{\mathrm{H}}$ in Step 2 can be avoided. This further reduces the computational burden in BICG.

For convenience, we write the solutions of $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{z}$ obtained by the BICG as $\mathrm{BICG}(\boldsymbol{M}, \boldsymbol{z}, \boldsymbol{y}_0, \varepsilon)$. In practice, the pre-conditioner can be used to speed up the convergence of BICG [25]. More specifically,

Algorithm 3 BICG-FOCUSS

Input: basis matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m\times n}$, data $\boldsymbol{x} \in \mathbb{C}^{m\times 1}$

Perform SVD on matrix $\boldsymbol{\Phi}$ to obtain $\boldsymbol{U}$, Σ and $\boldsymbol{V}$ as (5). Let $\boldsymbol{V}$=$[\boldsymbol{V}_1^{\mathrm{H}}, \boldsymbol{V}_2^{\mathrm{H}}]$, where the size of $\boldsymbol{V}_1$ is $m \times n$, and let $\hat{\boldsymbol{x}} = \Sigma^{-1}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{x}$.

Parameters: parameter l , tolerance ε in BICG, termination level ς

Initialization: $\boldsymbol{\gamma}^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \ldots, \gamma_n^{(0)}]^{\mathrm{T}}$, $k = 1$

Iteration:

1: $\boldsymbol{W}_k = \mathrm{diag}(|\gamma_1^{(k\text{-}1)}|^l, \ldots, |\gamma_n^{(k\text{-}1)}|^l)$

2: If $\mathrm{rank}(\boldsymbol{W}^k) \geqslant m$

$\boldsymbol{\varphi}^{(k)} = \mathrm{BICG}(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}}, \boldsymbol{x}, \boldsymbol{\varphi}^{(0)}, \varepsilon)$

$\boldsymbol{q}^{(k)} = \boldsymbol{\Phi}_k^{\mathrm{H}}\boldsymbol{\varphi}^{(k)}$

3: Else

Compute $\boldsymbol{V}_{1k}$ as (12) and design a column exchanging matrix $\boldsymbol{T}$, and obtain a matrix as

$\tilde{\boldsymbol{V}}_{1k} = \boldsymbol{V}_{1k}\boldsymbol{T} = [\tilde{\boldsymbol{V}}_k \boldsymbol{0}_{m\times(n-r)}]$, $\boldsymbol{H}_k = \tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\tilde{\boldsymbol{V}}_k$

$\boldsymbol{\eta}^{(k)} = \mathrm{BICG}(\boldsymbol{H}_k, \tilde{\boldsymbol{V}}_k{}^{\mathrm{H}}\hat{\boldsymbol{x}}, \boldsymbol{\eta}^{(0)}, \varepsilon)$

$\mathbf{G}_k = \begin{pmatrix} \boldsymbol{\eta}^{(k)} \\ \boldsymbol{0} \end{pmatrix}$, $\boldsymbol{q}^{(k)} = \boldsymbol{T}\mathbf{G}_k$

4: End

5: $\boldsymbol{\gamma}^{(k)} = \boldsymbol{W}_k\boldsymbol{q}^{(k)}$

6: break until $||\boldsymbol{\gamma}^{(k)} - \boldsymbol{\gamma}^{(k-1)}||_2/||\boldsymbol{\gamma}^{(k)}||_2 < \varsigma$, else let $k = k + 1$ and go to Step 1.

Output: $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(k)}$

we can use a non-singular matrix $\boldsymbol{C}$ to transform $\boldsymbol{My} = \boldsymbol{z}$ into $\boldsymbol{C}^{-\mathrm{H}}\boldsymbol{MC}^{-1}(\boldsymbol{Cy}) = \boldsymbol{C}^{-\mathrm{H}}\boldsymbol{z}$ such that the eigenvalue distribution of $\boldsymbol{C}^{-\mathrm{H}}\boldsymbol{MC}^{-1}$ is more concentrated. In this method, the pre-conditioner $\boldsymbol{P} = \boldsymbol{C}^{\mathrm{H}}\boldsymbol{C}$. When $\boldsymbol{P}$ is effective, the convergence of BICG can be accelerated. However, no single pre-conditioner is the best for all types of linear equation and the effectiveness of $\boldsymbol{P}$ varies from one linear equation to another. There are many types of pre-conditioning strategies such as the symmetric successive over-relaxation (SSOR) scheme [25] and the incomplete cholesky (IC) factorization scheme [25]. In the experiments, we use the two pre-conditioning schemes, which we term SSOR-BICG and IC-BICG for convenience.

2.2.3 *BICG-FOCUSS*

The use of the analysis in Subsections 2.2.1 and 2.2.2 enables us to obtain BICG-based FOCUSS, i.e., BICG-FOCUSS. Since the algorithm avoids the computation of the inverse, it is a fast algorithm of FOCUSS. The detailed procedures of BICG-FOCUSS are given as Algorithm 3.

When there exists additive noise in $\boldsymbol{x}$, FOCUSS becomes the regularized FOCUSS as [27, 28]

$$\boldsymbol{\gamma}^{(k)} = \boldsymbol{W}_k\boldsymbol{\Phi}_k^{\mathrm{H}}(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}} + \sigma\boldsymbol{I})^{-1}\boldsymbol{x}, \tag{20}$$

where $\boldsymbol{I}$ is an identity matrix, and $\sigma(\sigma > 0)$ is the regularization parameter that relates to the noise level. It is known that, regardless of $\mathrm{rank}(\boldsymbol{W}_k) < m$ or $\mathrm{rank}(\boldsymbol{W}_k) \geqslant m$, $\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}} + \sigma\boldsymbol{I}$ is invertible when the parameter σ is appropriate. Then, $(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}} + \sigma\boldsymbol{I})^{-1}\boldsymbol{x}$ can be written as $\mathrm{BICG}((\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}} + \sigma\boldsymbol{I}), \boldsymbol{x}, \boldsymbol{\varphi}^{(0)}, \varepsilon)$. The detailed procedures of BICG-FOCUSS under noisy conditions are illustrated as in Algorithm 4.

2.2.4 *Computational complexity*

It can be seen that BICG-FOCUSS is easy to implement in hardware because it avoids the calculation of matrix inversion and uses an iterative method to compute the solution of a linear equation. We know from the iteration of BICG that the computational complexity of BICG to solve a linear equation $\boldsymbol{My} = \boldsymbol{x}(\boldsymbol{M} \in \mathbb{C}^{r\times r})(r \leqslant m)$ is $O(r^2)$ and the direct way to compute $\boldsymbol{M}^{-1}\boldsymbol{x}$ is $O(r^3)$. For the original FOCUSS, the conventional method (e.g., singular value decomposition) is used to calculate the Moore Penrose inverse $(\boldsymbol{\Phi}_k)^+$, where its computational complexity is $O(nm^2)$ [29]. It shows that, compared with the existing FOCUSS, our proposed algorithm reduces the computational complexity by one order.

Algorithm 4 BICG-FOCUSS (noise)

Input: basis matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m\times n}$, data $\boldsymbol{x} \in \mathbb{C}^{m\times 1}$

Parameters: parameter l, tolerance ε in BICG, termination level ς, noise level σ

Initialization: $\boldsymbol{\gamma}^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \ldots, \gamma_n^{(0)}]^{\mathrm{T}}$, $k = 1$

Iteration:

1: $\boldsymbol{W}_k = \mathrm{diag}(|\gamma_1^{(k\text{-}1)}|^l, \ldots, |\gamma_n^{(k\text{-}1)}|^l)$

2: $\boldsymbol{\varphi}^{(k)} = \mathrm{BICG}(\boldsymbol{\Phi}_k\boldsymbol{\Phi}_k^{\mathrm{H}} + \sigma\boldsymbol{I}, \boldsymbol{x}, \boldsymbol{\varphi}^{(0)}, \varepsilon)$

3: $\boldsymbol{\gamma}^{(k)} = \boldsymbol{W}_k\boldsymbol{\Phi}_k^{\mathrm{H}}\boldsymbol{\varphi}^{(k)}$

4: break until $||\boldsymbol{\gamma}^{(k)} - \boldsymbol{\gamma}^{(k-1)}||_2/||\boldsymbol{\gamma}^{(k)}||_{\,2} < \varsigma$, else let $k = k + 1$ and go to Step 1.

Output: $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(k)}$

Thus, BICG-FOCUSS reduces the computational complexity of the original FOCUSS, which becomes a fast FOCUSS algorithm.

2.3 Performance analysis

In this subsection, we use two experiments to show the performance of BICG-FOCUSS. We compare the runtimes and the estimating accuracies of FOCUSS, SOR-FOCUSS, BICG-FOCUSS, SSOR-BICG-FOCUSS, and IC-BICG-FOCUSS. According to [25], when the relax factor ω satisfies $1 < \omega < 2$, SOR can converge to the true solution. The optimal relax factor ω is commonly obtained on trial by different ω $(1 < \omega < 2)$, and it is unrealistic in actual computations. In the simulations, we choose an arbitrary relax factor as $\omega = 1.6$. The tolerance ε and the initial vector in BICG, SSOR-BICG, IC-BICG, and SOR are set as 0.01 and the zero vector, respectively. The estimating accuracy is determined by the signal-to-interference ratio (SIR), which measures the error between the true sparse solution $\boldsymbol{\gamma}$ and its estimation $\tilde{\boldsymbol{\gamma}}$, and is defined as follows:

$$\mathrm{SIR}(\tilde{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) = -20\log_{10}\frac{||\tilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma}||_2}{||\boldsymbol{\gamma}||_2} \ \ (\mathrm{dB}). \tag{21}$$

We ensure that the experimental result is more representative by using Algorithm 3 in the experiments. Since our derivations of BICG-FOCUSS holds for any parameter l in FOCUSS and, when $l > 0.5$, FOCUSS converges to a sparse solution, we take an arbitrary parameter l as 0.6. $\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{x}$ is taken as the initialization $\boldsymbol{\gamma}^{(0)}$ of the five algorithms.

In the first experiment, we show the influence of the size of the basis matrix on the runtimes and the SIRs of the five algorithms. The basis matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m\times n}$ is taken as an over-complete Fourier matrix. The ith $(1 \leqslant i \leqslant n)$ column of $\boldsymbol{\Phi}$ is the vector $[1, \exp(\mathrm{j}2\pi 2t_i), \ldots, \exp(\mathrm{j}2\pi(m-1)t_i)]^{\mathrm{T}}$, where $t_i = -0.5 + i/n$ $(1 \leqslant i \leqslant n)$. Take $m = 500$, and the number of the columns of $\boldsymbol{\Phi}$ is determined by N as $n = m + 100 \cdot N$. Then any m columns of $\boldsymbol{\Phi}$ are linearly independent. Assume that the observable data $\boldsymbol{x}$ is set as a linear combination of fifty columns of $\boldsymbol{\Phi}$, and their amplitudes are 10. Figure 1 shows the comparisons of the runtimes of the five algorithms under $\varsigma = 0.1$ and $\varsigma = 0.01$, respectively. We determine that BICG-FOCUSS has minimal costs in terms of time in the five algorithms, and SSOR scheme and IC scheme cannot speed up the convergence rate of BICG in the experiment. Figure 2 shows the SIRs of the five algorithms under $\varsigma = 0.1$ and $\varsigma = 0.01$, respectively. It can be seen from Figure 2 that, under the same ς, the SIRs of the five algorithms are almost the same and decrease slightly as N increases except that SOR-FOCUSS is slightly lower when $\varsigma = 0.01$.

In the second experiment, we compare the runtimes and the SIRs of these five algorithms for different sparsities of the observable data $\boldsymbol{x}$ under a fixed basis matrix. The basis matrix $\boldsymbol{\Phi} \in \mathbb{C}^{1024\times 4039}$ is generated by MATLAB 2012, and its real and imaginary parts are distributed uniformly on the open interval (0, 1). The observable data $\boldsymbol{x}$ is set as a linear combination of some columns of $\boldsymbol{\Phi}$, and their amplitudes are 10. Then the sparsity of the data is the number of columns that are the components of the data. Let $\varsigma = 0.01$. Figure 3(a) plots the runtimes of the five algorithms. It can be seen that BICG-FOCUSS and SSOR-BICG-FOCUSS are significantly less time consuming than SOR-FOCUSS and IC-BICG-FOCUSS. The runtimes of BICG-FOCUSS is less than that of SSOR-BICG-FOCUSS. This shows

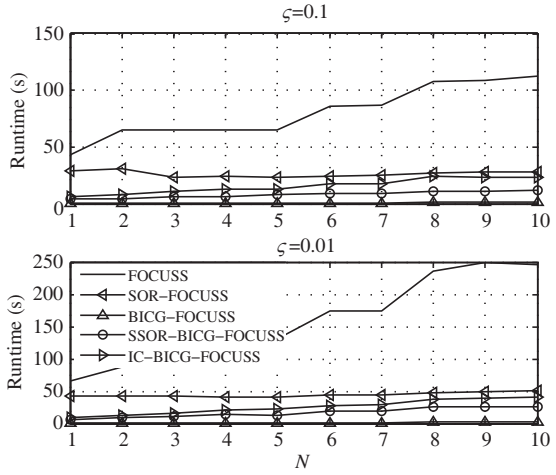

Figure 1 Runtime comparisons.

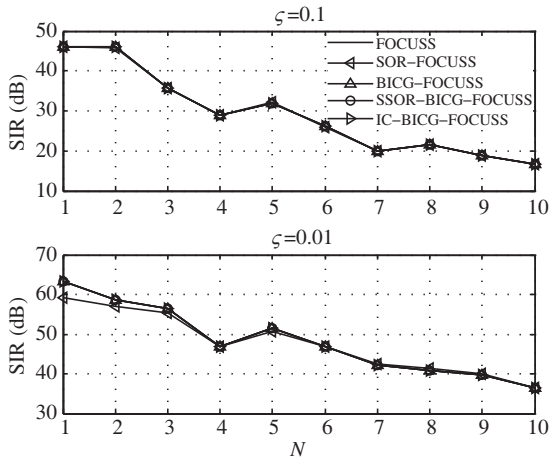

Figure 2 SIR comparisons.

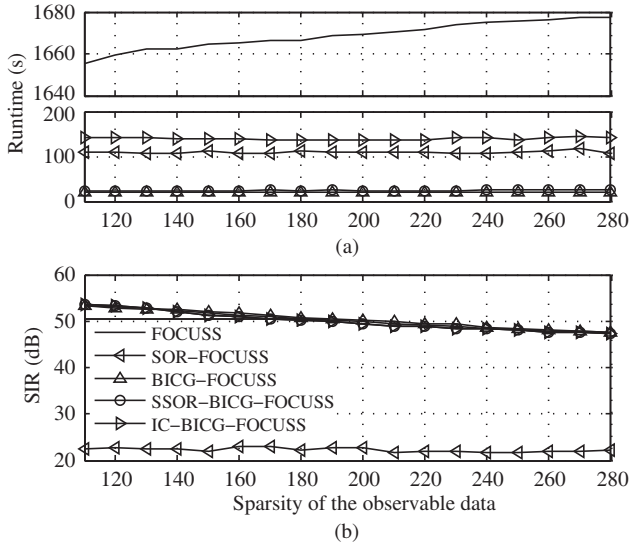

Figure 3 (a) Runtime comparisons; (b) SIR comparisons.

that SSOR scheme and IC scheme cannot speed up the convergence rate of BICG in this experiment. Figure 3(b) shows the outcomes of SIR for the five algorithms. The SIRs of FOCUSS, BICG-FOCUSS, SSOR-BICG-FOCUSS, and IC-BICG-FOCUSS are almost equal and are reduced as the sparsity of the data increases. This shows that the SIRs of these algorithms are the same. In addition, the SIR of SOR-FOCUSS is lower than that of the other algorithms by a large extent. Therefore, BICG-FOCUSS consumes less time than other algorithms without losing accuracy.

3 Application in space-time clutter spectrum estimation

The SR-STAP is a promising technique in detecting low-velocity targets in a strongly cluttered environment [4, 16–21]. In SR-STAP, the space-time clutter spectrum is the sparse solution of a complex underdetermined system, and can be used to estimate the clutter covariance which is used to detect the target [4, 16–21, 30, 31]. FOCUSS is used to estimate the space-time clutter spectrum and has been shown to be capable of good estimation performance [4]. However, when the number of coherent pulses N and the number of array antennas M are larger, or the requirement of the resolution of the spatial-Doppler frequency is high, the scale of the complex underdetermined system is very large. Then, it is

Table 1 Some of the radar parameters

Parameter	Symbol	Value
Platform velocity	v	200 m/s
Pulse repetition frequency	T	625 Hz
Radar wavelength	λ	0.6897 m
Inter-sensor spacing	d	0.333 m

expected to take less time to estimate the space-time clutter spectrum. In this section, we employ our proposed BICG-FOCUSS to estimate the space-time clutter spectrum.

3.1 Application in space-time clutter spectrum estimation

The model of the space-time clutter spectrum is obtained by discretizing the spatial frequency area into $N_{\rm s} = \rho_{\rm s} M$ ($\rho_{\rm s} > 1$) grids and the Doppler frequency interval into $N_{\rm d} = \rho_{\rm d} M$ ($\rho_{\rm d} > 1$) grids as in [4, 19–25], where $\rho_{\rm s}$ and $\rho_{\rm d}$ are the resolution scale. Each grid node in the discretized plane corresponds to a certain space-time steering vector and all of these vectors form a basis matrix $\boldsymbol{\Phi}$. The clutter $\boldsymbol{x}$ in the test cell can be written as a complex underdetermined system as [4, 16–21]

$$\boldsymbol{x}{=}\boldsymbol{\Phi}\gamma{+}\boldsymbol{n}, \tag{22}$$

where $\boldsymbol{n}$ is the additive noise. The basis matrix $\boldsymbol{\Phi}(NM \times N_{\rm s}N_{\rm d})$ contains all the possible space-time steering vectors of $\boldsymbol{x}$ and is given as

$$\boldsymbol{\Phi} = [\boldsymbol{v}(f_{{\rm s},1}, f_{{\rm d},1}), \boldsymbol{v}(f_{{\rm s},1}, f_{{\rm d},2}), \ldots, \boldsymbol{v}(f_{{\rm s},N_{\rm s}}, f_{{\rm d},N_{\rm d}})], \tag{23}$$

where $\boldsymbol{v}(f_{{\rm s},n}, f_{{\rm d},m})$ is the space-time steering vector with the spatial frequency $f_{{\rm s},n}$ and the Doppler frequency $f_{{\rm d},m}$, and its form can be found in [4, 16–21]. The vector $\boldsymbol{\gamma}(N_{\rm s}N_{\rm d}{\times}1)$ represents the distribution of the clutter $\boldsymbol{x}$ in the basis matrix $\boldsymbol{\Phi}$ (i.e., the space-time clutter spectrum).

3.2 Experiments

Let $\Delta f_{\rm s}$ and $\Delta f_{\rm d}$ be the intervals of the discretized spatial frequency and the discretized Doppler frequency, respectively. According to the form of the space-time steering vector, $f_{{\rm s},n}$ and $f_{{\rm d},m}$ can be normalized into $(-0.5, 0.5]$. Then we can take $\Delta f_{\rm s} = 1/N_{\rm s}$ and $\Delta f_{\rm d} = 1/N_{\rm d}$. Let $f_{{\rm s},n} = -0.5 + \Delta f_{\rm s}(n-1)(1 \leqslant n \leqslant N_{\rm s})$, and let $f_{{\rm d},m} = -0.5 + \Delta f_{\rm d}(m-1)(1 \leqslant m \leqslant N_{\rm d})$. In the following, we design experiments to show the performance of BICG-FOCUSS in estimating the space-time clutter spectrum. We compare BICG-FOCUSS with FOCUSS, SOR-FOCUSS, SSOR-BICG-FOCUSS, and IC-BICG-FOCUSS. The initialization of these algorithms is $\boldsymbol{\Phi}^{\rm H}\boldsymbol{x}$. Similar as in Subsection 2.3, the parameters l and ω are taken as 0.6 and 1.6, respectively. The tolerance ε and the initial vector in BICG, SSOR-BICG, IC-BICG, and SOR are set as 0.01 and zero vector, respectively. The termination level ς is set as 0.001.

3.2.1 *Simulations in mono-static array radar*

In this section, we present the simulations to illustrate the performance of BICG-FOCUSS in estimating the space-time clutter spectrum under mono-static array radar. Some parameters of the radar are given in Table 1. The geometry of the radar array configuration we employ is same as that in [4]. Set the crab angle ψ as 30^0. Our simulations are designed to show the performance of Algorithm 3 under noise-free conditions. It is known that the relationship between the Doppler frequency and the look direction is as follows [4]:

$$f_{\rm d} = \frac{v}{\lambda}\left(\sqrt{3}\cos\beta - \sqrt{1 - \left(\frac{H}{R}\right)^2 - \cos^2\beta}\right), \tag{24}$$

where R is the range, β is the look direction, and H is the height of the radar. For a certain range R, the space-time clutter spectrum is supposed to be located on the curve of $(f_{\rm d}, f_{\rm s})$, where $f_{\rm s}{=}\sin\beta \cdot d/\lambda$. Set

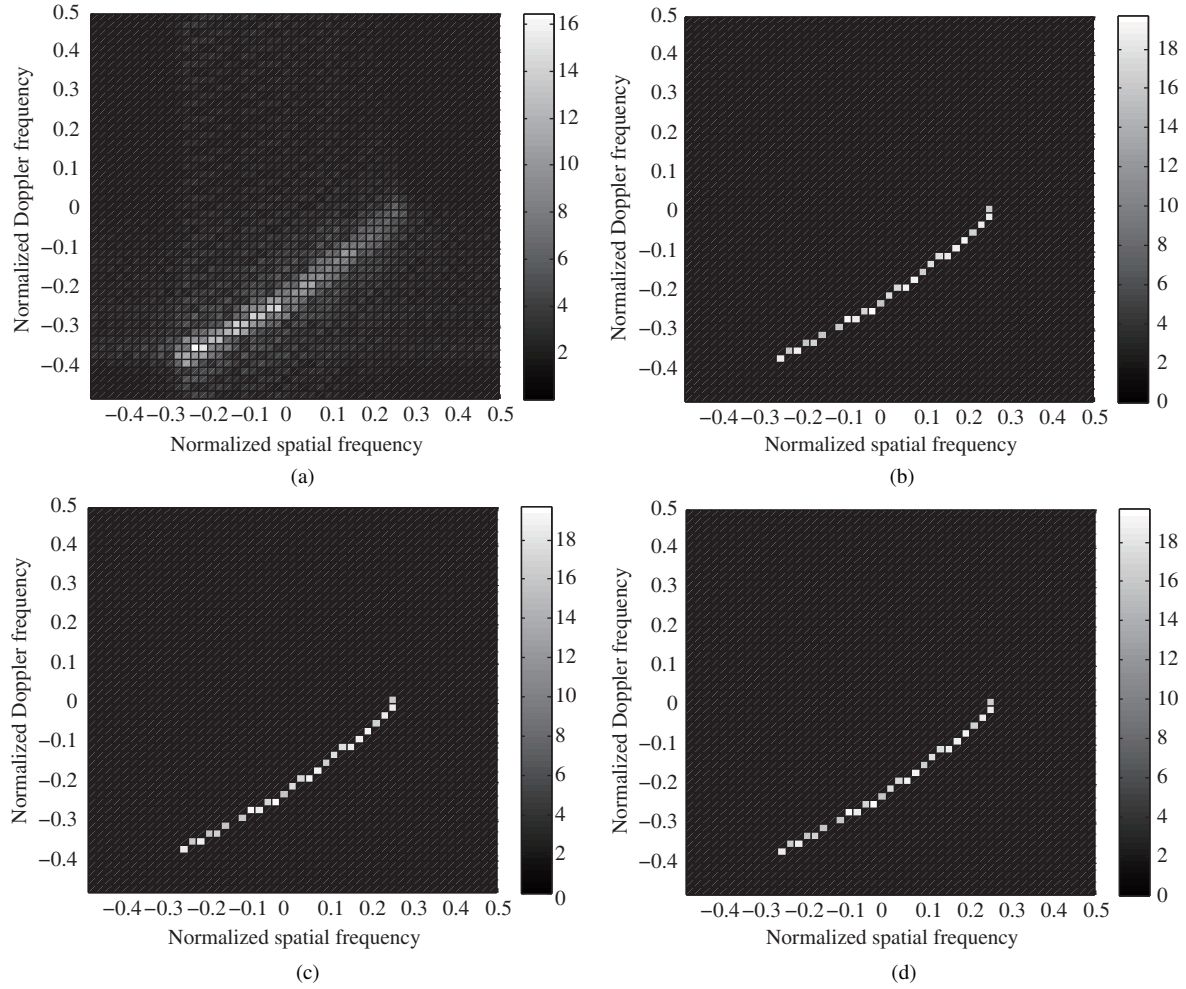

Figure 4 Space-time clutter spectrums estimated by (a) Fourier transform, (b) FOCUSS, (c) SOR-FOCUSS, and (d) BICG-FOCUSS.

$H = 1000$ m, and choose $R = 12$ km as the range cell. We take $M = 30$ and $N = 30$, and set $N_{\rm s} = 50$ and $N_{\rm d} = 50$. Then the size of the basis matrix $\boldsymbol{\Phi}$ is 900×2500. We generate clutter scatters such that they are located on the curve $(f_{\rm d}, f_{\rm s})$. The look directions of the clutter scatters are -30^0 to 30^0. The expectation of their amplitudes is 20. These clutter scatters are then processed such that they are located on the grid nodes of the spatial-Doppler plane. The traditional space-time clutter spectrum is obtained by Fourier transform. We show the traditional space-time clutter spectrum in Figure 4(a) to facilitate comparison. Since the estimating accuracies of BICG-FOCUSS, SSOR-BICG-FOCUSS, and IC-BICG-FOCUSS are the same, we do not show the outcomes of SSOR-BICG-FOCUSS and IC-BICG-FOCUSS on the spatial-Doppler plane. Figure 4(b), (c), and (d) show the space-time clutter spectrums estimated by FOCUSS, SOR-FOCUSS, and BICG-FOCUSS, respectively. It can be seen that the traditional space-time clutter spectrum is blurred on the plane, whereas the space-time clutter spectrum estimated by other algorithms can clearly concentrate the clutter energy on the expected curve. Table 2 presents the comparisons of the SIRs and the runtimes. It can be seen that BICG-FOCUSS can estimate the space-time clutter spectrum more accurately and faster.

3.2.2 *Experiments in the mountaintop system*

We evaluate the performance of BICG-FOCUSS by using the mountaintop data obtained [32]. Except for the parameters provided in Table 1, we let $M = 14$ and $N = 16$. We take $N_{\rm s} = 50$ and $N_{\rm d} = 50$. Then the size of the basis matrix $\boldsymbol{\Phi}$ is 224×2500. We take the data in the 147th range cell and use

Table 2 Performance of the algorithms

Algorithm	SIR	Runtime (s)
Fourier transform	0.345	81.32
FOCUSS	45.74	236.43
SOR-FOCUSS	44.18	60.86
SSOR-BICG-FOCUSS	45.73	59.36
IC-BICG-FOCUSS	45.74	92.71
BICG-FOCUSS	45.73	28.36

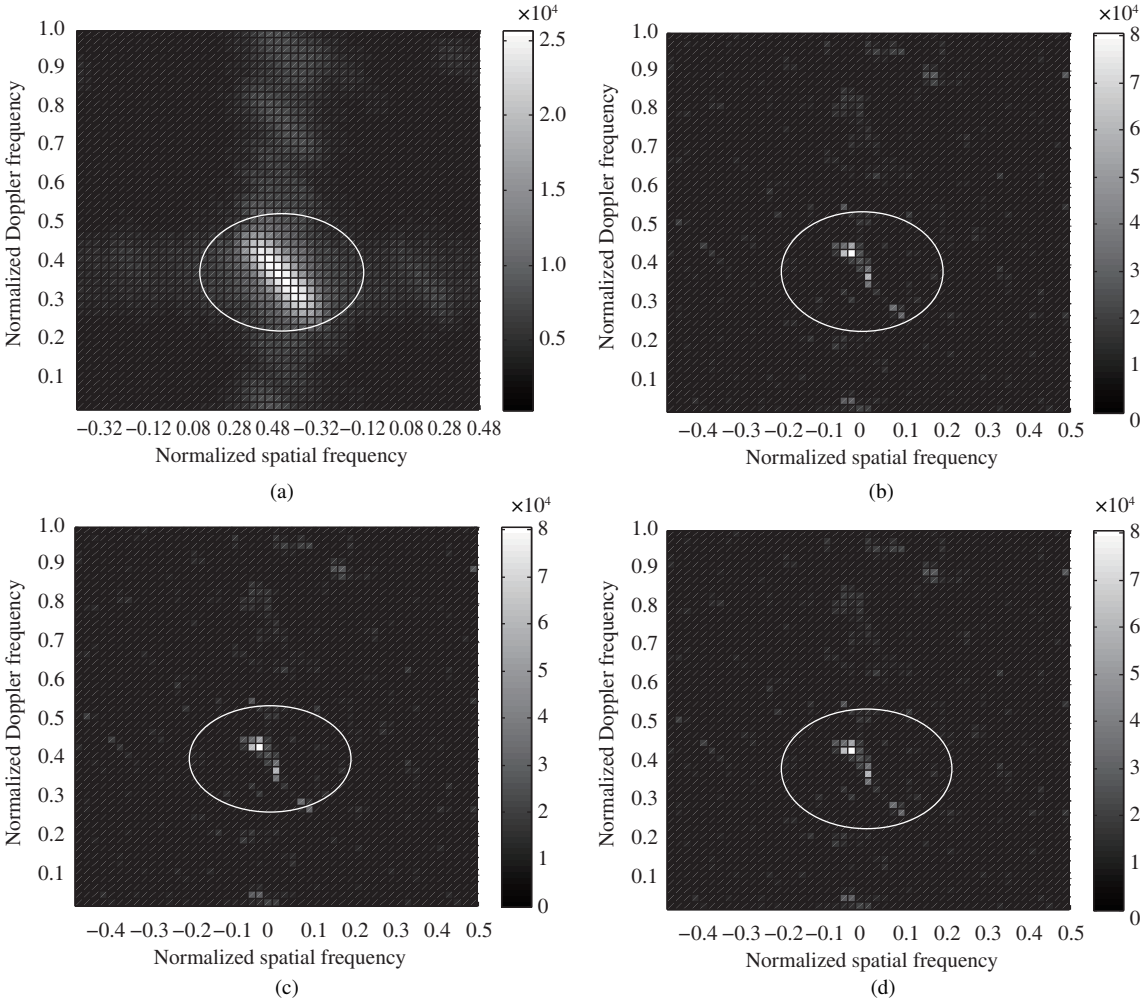

Figure 5 Space-time clutter spectrums estimated by (a) Fourier transform, (b) FOCUSS, (c) SOR-FOCUSS, and (d) BICG-FOCUSS.

Algorithm 4 in the experiments. Set $\sigma = 1$ in the algorithm. Figure 5(a) shows the traditional space-time clutter spectrum obtained by Fourier transform. Figure 5(b), (c), and (d) show the space-time clutter spectrums estimated by FOCUSS, SOR-FOCUSS, and BICG-FOCUSS, respectively. In Figure 5, the space-time clutter spectrums are circled by ellipses. We can see from Figure 5(a) that the space-time clutter spectrum is blurred in some areas. However, the spatial frequencies and the Doppler frequencies of the clutter scatters in Figure 5(b), (c), and (d) are showed clearly. This shows that FOCUSS, SOR-FOCUSS, and BICG-FOCUSS can accurately display the space-time clutter spectrum. Since we do not know the true space-time clutter spectrum, we cannot compare the estimating accuracies of these algorithms and show only the runtimes of these algorithm in Table 3. It can be seen that BICG-FOCUSS is the most computationally efficient of the five algorithms. Thus, BICG-FOCUSS is desirable in space-

Table 3 Performance of the algorithms

Algorithm	Runtime (s)
Fourier transform	41.23
FOCUSS	70.33
SOR-FOCUSS	15.82
SSOR-BICG-FOCUSS	14.11
IC-BICG-FOCUSS	22.34
BICG-FOCUSS	13.73

time clutter spectrum estimation for mountaintop data.

4 Conclusion

This paper presents the fast version of FOCUSS, i.e., BICG-FOCUSS, which we developed to reduce the computational complexity of the original FOCUSS. The purpose of BICG-FOCUSS is to largely reduce the computational complexity of FOCUSS by the BICG method. We show that the computational complexity of BICG-FOCUSS is lower than that of the FOCUSS. The experiments, which compare the runtimes and SIRs of FOCUSS, SOR-FOCUSS, SSOR-BICG-FOCUSS, IC-BICG-FOCUSS, and BICG-FOCUSS, show that the time consumption of BICG-FOCUSS is significantly lower than that of other algorithms without losing accuracy. In addition, we use BICG-FOCUSS to estimate the space-time clutter spectrum in SR-STAP. Experiments with both the simulated and mountaintop data show that BICG-FOCUSS is capable of obtaining the accurate space-time clutter spectrum within minimum time.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61421001, 61331021, 61671060).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Gorodnitsky I F, George J S, Rao B D. Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm. Electroencephalogr Clin Neurophysiol, 1995, 95: 231–251
2 Gorodnitsky I F, Rao B D. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. IEEE Trans Signal Process, 1997, 45: 600–616
3 Yang X Y, Chen B X, Chen Y H. An eigenstructure-based 2D DOA estimation method using dual-size spatial invariance array. Sci China Inf Sci, 2011, 54: 163–171
4 Sun K, Meng H, Wang Y, et al. Direct data domain STAP using sparse representation of clutter spectrum. Signal Process, 2011, 91: 2222–2236
5 Alonso M T, Lopez-Dekker P, Mallorqui J J. A novel strategy for radar imaging based on compressive sensing. IEEE Trans Geosci Remote Sens, 2010, 48: 4285–4295
6 Bu H X, Bai X, Tao R. Compressed sensing SAR imaging based on sparse representation in fractional Fourier domain. Sci China Inf Sci, 2012, 55: 1789–1800
7 Yang J Y, Peng Y G, Xu W L, et al. Ways to sparse representation: an overview. Sci China Ser F-Inf Sci, 2009, 52: 695–703
8 Tropp J. Greed is good: algorithmic results for sparse approximation. IEEE Trans Inf Theory, 2004, 50: 2231–2242
9 Wu R, Huang W, Chen D R. The exact support recovery of sparse signals with noise via orthogonal matching pursuit. IEEE Signal Process Lett, 2013, 20: 403–406
10 Chen S S, Donoho D L, Saunders M A. Atomic decomposition by basis pursuit. SIAM J Sci Comput, 1998, 20: 33–161
11 Selesnick I W, Bayram I. Sparse signal estimation by maximally sparse convex optimization. IEEE Trans Signal Process, 2014, 62: 1078–1092
12 Rao B D, Kreutz-Delgado K. An affine scaling methodology for best basis selection. IEEE Trans Signal Process, 1999, 47: 187–200
13 Xie K, He Z, Cichocki A. Convergence analysis of the FOCUSS algorithm. IEEE Trans Neural Netw Learn Syst, 2015, 26: 601–613
14 He Z, Cichocki A, Zdunek R, et al. Improved FOCUSS method with conjugate gradient iteration. IEEE Trans Signal Process, 2009, 57: 399–404
15 Hu C X, Liu Y M, Li G, et al. Improved FOCUSS method for reconstruction of cluster structured sparse signals in radar imaging. Sci China Inf Sci, 2012, 55: 1776–1788

16 Sun K, Zhang H, Li G, et al. Airborne radar STAP using sparse recovery of clutter spectrum. arXiv:1008.4185
17 Yang Z, de Lamare R C, Li X. Sparsity-aware space-time adaptive processing algorithms with L1-norm regularization for airborne radar. IET Signal Process, 2012, 6: 413–423
18 Yang Z, Li X, Wang H, et al. On clutter sparsity analysis in space-time adaptive processing airborne radar. IEEE Geosci Remote Sens Lett, 2013, 10: 1214–1218
19 Wang L, Liu Y, Ma Z, et al. A novel STAP method based on structured sparse recovery of clutter spectrum. In: Proceedings of IEEE Radar Conference (RadarCon), Arlington, 2015. 561–565
20 Yang Z, Liu Z, Li X, et al. Performance analysis of STAP algorithms based on fast sparse recovery techniques. Prog Electromagn Res B, 2012, 41: 251–268
21 Sen S. Low-rank matrix decomposition and spatio-temporal sparse recovery for STAP radar. IEEE J Sel Top Signal Process, 2015, 9: 1510–1523
22 Fletcher R. Conjugate gradient methods for indefinite systems. In: Numerical Analysis. Berlin: Springer, 1976. 73–89
23 Joly P, Meurant G. Complex conjugate gradient methods. Numer Math, 1993, 4: 379–406
24 Mihalyffy L. An alternative representation of the generalized inverse of partitioned matrices. Linear Algebra Appl, 1971, 4: 95–100
25 Saad Y. Iterative Methods for Sparse Linear Systems. Boston: PWS-Kent, 1995
26 Bank R E, Chan T F. An analysis of the composite step biconjugate gradient algorithm for solving nonsymmetric systems. Numer Math, 1993, 66: 295–319
27 Rao B D, Engan K, Cotter S F, et al. Subset selection in noise based on diversity measure minimization. IEEE Trans Signal Process, 2003, 51: 760–770
28 Peng Y, Fei Y, Feng Y. Sparse array synthesis with regularized FOCUSS algorithm. In: Prceedings of International Symposium of the IEEE Antennas and Propagation Society, 2013. 1406–1407
29 Golub G H, Van-Loan C F. Matrix Computations. 3rd ed. Boltimore and London: The Johns Hopkins University Press, 1996
30 Duan K Q, Xie W C, Wang Y L. Nonstationary clutter suppression for airborne conformal array radar. Sci China Inf Sci, 2011, 54: 2170–2177
31 Wu R B, Jia Q Q, Li H. A novel STAP method for the detection of fast air moving targets from high speed platform. Sci China Inf Sci, 2012, 55: 1259–1269
32 Peckham C D, Haimovich A M, Ayoub T F, et al. Reduced-rank STAP performance analysis. IEEE Trans Aerosp Electron Syst, 2000, 36: 664–676