

Article

A Fast Space-Time Adaptive Processing Algorithm Based on Sparse Bayesian Learning for Airborne Radar

Cheng Liu, Tong Wang *, Shuguang Zhang and Bing Ren

National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, China; m18392121020@163.com (C.L.); SG_Zhang12138@163.com (S.Z.); renb0327@163.com (B.R.)

* Correspondence: twang@mail.xidian.edu.cn

Abstract: Space-time adaptive processing (STAP) plays an essential role in clutter suppression and moving target detection in airborne radar systems. The main difficulty is that independent and identically distributed (i.i.d) training samples may not be sufficient to guarantee the performance in the heterogeneous clutter environment. Currently, most sparse recovery/representation (SR) techniques to reduce the requirement of training samples still suffer from high computational complexities. To remedy this problem, a fast group sparse Bayesian learning approach is proposed. Instead of employing all the dictionary atoms, the proposed algorithm identifies the support space of the data and then employs the support space in the sparse Bayesian learning (SBL) algorithm. Moreover, to extend the modified hierarchical model, which can only apply to real-valued signals, the real and imaginary components of the complex-valued signals are treated as two independent real-valued variables. The efficiency of the proposed algorithm is demonstrated both with the simulated and measured data.

Keywords: space-time adaptive processing; airborne radar; sparse recovery/representation; computational complexities; sparse Bayesian learning; group sparse



Citation: Liu, C.; Wang, T.; Zhang, S.; Ren, B. A Fast Space-Time Adaptive Processing Algorithm Based on Sparse Bayesian Learning for Airborne Radar. *Sensors* **2022**, *22*, 2664. <https://doi.org/10.3390/s22072664>

Academic Editor: Antonio Pauciuolo

Received: 21 February 2022

Accepted: 29 March 2022

Published: 30 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Space-time adaptive processing (STAP) has the capability to detect slow-moving targets that might otherwise be swallowed up by the strong sidelobe clutter. The performance of STAP filter is dependent on the accuracy of the clutter plus noise covariance matrix (CNCM) of the cell under test (CUT) [1]. According to the well-known Reed-Mallet-Brennan (RMB) rule [2], to achieve the steady performance, the number of the independent and identically distributed (i.i.d) secondary range cells used to estimate CNCM is no less than twice the system degrees of freedom (DOF). However, it is hard to obtain enough i.i.d samples in practice because of array geometry structures, nonhomogeneous clutter environment and so on [1]. Knowing how to improve the performance of STAP with limited samples has been a hot topic until now.

Reduced-dimension (RD) [3] and reduced-rank (RR) [4] algorithms are proposed to improve the performance of STAP with limited secondary samples. Even though they are easy to implement, their performance gets worse when the number of secondary samples is smaller than twice the rank of clutter [5].

Apart from RD and RR algorithms, some other algorithms [6–10] are proposed and succeed in suppressing clutter in theory. However, they face some disadvantages in practice. Sufficiently, (1) direct data domain (DDD) algorithms in [6] that achieve enough samples from the CUT suffer from aperture loss; and (2) knowledge-aided (KA) algorithms in [7–10] state that the accurate prior knowledge must be required in advance. Since the cost to achieve the accurate prior knowledge is expensive and the prior knowledge changes with time, KA algorithms are not widely used in practical applications.

Over the past twenty years, sparse recovery/representation (SR) algorithms have received continuous attention in STAP [11–15] because they have enormous potential to

estimate the clutter spectrum with limited samples. The sparse Bayesian learning (SBL)-type algorithms are robust and have drawn continuous attention in the last five years.

The SBL algorithm was proposed by Tipping in [16]. Wipf enhanced it in the single measurement vector (SMV) case in [17] and then the multiple measurement vectors (MMV) case in [18]. Due to the excellent performance of SBL algorithms, the fast marginal likelihood maximization (FMLM) algorithm [19], the Bayesian compressive sensing (BCS) algorithm [20], the multitask BCS (M-BCS) algorithm [21], and other Bayesian algorithms [22–24] were proposed by researchers in the following years. SBL was firstly introduced into STAP with MMV, defined as the M-SBL-STAP algorithm in [25], by Duan to estimate CNCM and Wang improved the fast convergence of the algorithm in [26]. SBL has also been used to solve some common problems in STAP in the last five years, such as off-grid in [27] and discrete interference in [28]. However, the Bayesian algorithms aforementioned have one or more of the following disadvantages: (a) high computational cost, (b) inaccurate estimation of the noise variance and (c) inapplicability to complex-valued signals.

In this paper, to improve the computational efficiency of the M-SBL-STAP algorithm, we extend the FMLM algorithm to the conventional M-SBL-STAP algorithm. The real and imaginary components of the complex-valued signals are treated as two independent real-valued variables in order to extend the modified hierarchical model. Simulation results with both simulated and Mountain-Top data demonstrate that the proposed algorithm can achieve high computational efficiency and good performance.

The main contributions of this paper can be listed as follows:

1. We extend the FMLM algorithm into M-SBL-STAP for the purpose of identifying the support space of the data, i.e., the atoms whose corresponding hyper-parameters are non-zero. After support space identification, the dimensions of the effective problems are drastically reduced due to sparsity, which can reduce computational complexities and alleviate memory requirements.
2. According to [18], we have no access to obtain the accurate value of the noise variance. Instead of estimating the noise variance, we extend the modified hierarchical model, introduced in Section 4, to the SBL framework.
3. Although the hierarchical models apply to the real-valued signals, they cannot be extended directly to the complex-valued signal according to [29,30]. The data needed to be dealt with in STAP are all complex-valued. To solve the problem, we transform sparse complex-valued signals into group sparse real-valued signals.

Notation: In this article, scalar quantities are denoted with italic typeface. Boldface small letters are reserved for vectors, and boldface capital letters are reserved matrices. The i -th entry of a vector \mathbf{x} is denoted by x_i , while \mathbf{A}_i and \mathbf{A}_{ij} denote the i -th row and (i, j) -th element of a matrix \mathbf{A} , respectively. The symbols $(\bullet)^T$ and $(\bullet)^H$ denote the matrix transpose and conjugate transpose, respectively. The symbols $\|\bullet\|_1$, $\|\bullet\|_2$ and $\|\bullet\|_F$ are reserved for ℓ_1 , ℓ_2 and Frobenius (ℓ_F) norms, respectively. $\|\bullet\|_0$ is reserved for ℓ_0 pseudo-norm. $\|\bullet\|_{2,0}$ stands for a mixed norm defined as the number of non-zero elements of the vector formed by the ℓ_2 norm of each row vector. The symbol $|\bullet|$ is reserved for the determinant. The notations \mathbf{I} , $\mathbf{0}$ and $\mathbf{1}$ represent the identity matrix, the all zero matrix and the all one matrix, respectively. The expectation of a random variable is denoted as $E(\bullet)$.

2. Background and Problem Formulation

2.1. STAP Signal Model for Airborne Radar

Consider an airborne pulsed-Doppler radar system with a side-looking uniform linear array (ULA) consisted of N elements. A coherent burst of M pulses is transmitted at a constant pulse repetition frequency (PRF) in a coherent processing interval (CPI). The complex sample at the CUT from the n -th element and the m -th pulse is denoted as x_{mn} , and the data for the CUT can be written as a $MN \times 1$ vector \mathbf{x} , termed a space-time snapshot.

$$\mathbf{x} = [x_{11}, x_{12}, \dots, x_{1N}, x_{21}, \dots, x_{MN}]^T \quad (1)$$

Radar systems need to ascertain whether the targets are present in the CUT; thus, target detection is formulated into a binary hypothesis problem: the hypothesis H_0 represents target absence and the other hypothesis H_1 represents target presence.

$$\begin{cases} \mathbf{x} = \mathbf{x}_c + \mathbf{n} & H_0 \\ \mathbf{x} = \alpha_t \mathbf{S}_t + \mathbf{x}_c + \mathbf{n} & H_1 \end{cases} \quad (2)$$

where α_t is the target amplitude and $\mathbf{S}_t \in \mathbb{C}^{MN \times 1}$ is the space-time vector of the target. The vector $\mathbf{n} \in \mathbb{C}^{MN \times 1}$ is the thermal noise, which is uncorrelated both spatially and temporally. A general model for the space-time clutter $\mathbf{x}_c \in \mathbb{C}^{MN \times 1}$ is

$$\mathbf{x}_c = \sum_{k=1}^{N_c} \alpha_k \mathbf{v}(f_{d,k}, f_{s,k}) \quad (3)$$

$$\mathbf{v}_{st}(f_{d,k}, f_{s,k}) = \mathbf{v}_t(f_{d,k}) \otimes \mathbf{v}_s(f_{s,k}) \quad (4)$$

$$\mathbf{v}_t(f_{d,k}) = [1, \exp(j\pi f_{d,k}), \dots, \exp(j\pi(M-1)f_{d,k})]^T \quad (5)$$

$$\mathbf{v}_s(f_{s,k}) = [1, \exp(j\pi f_{s,k}), \dots, \exp(j\pi(N-1)f_{s,k})]^T \quad (6)$$

where α_k denotes the random amplitude of the echo from the corresponding clutter patch; N_c denotes the number of clutter patches in a clutter ring; \mathbf{v}_{st} , \mathbf{v}_t and \mathbf{v}_s denote space-time steering vector, temporal steering vector and spatial steering vector, respectively; and $f_{d,k}$ and $f_{s,k}$ denote the corresponding normalized temporal frequency and spatial frequency, respectively.

According to the linearly constrained minimum variance (LCMV) criterion, the optimal STAP weight vector is

$$\mathbf{w}_{\text{opt}} = \frac{\mathbf{R}_{c+n}^{-1} \mathbf{S}_t}{\mathbf{S}_t^H \mathbf{R}_{c+n}^{-1} \mathbf{S}_t} \quad (7)$$

where the CNM \mathbf{R}_{c+n} is expressed as

$$\mathbf{R}_{c+n} = \mathbf{R}_c + \mathbf{R}_n = \mathbb{E} \left\{ \mathbf{x}_c \mathbf{x}_c^H \right\} + \sigma^2 \mathbf{I} \quad (8)$$

2.2. SR-STAP Model and Principle

Discretize the space-time plane uniformly into $K = N_s N_d$ grids, where $N_s = \varphi_s N$ ($\varphi_s > 1$) denotes the number of normalized spatial frequency bins and $N_d = \varphi_d M$ ($\varphi_d > 1$) denotes the number of normalized Doppler frequency bins. Each grid corresponds to a space-time steering vector \mathbf{v}_k , ($k = 1, 2, \dots, K$). The dictionary $\mathbf{D} \in \mathbb{C}^{MN \times K}$ used in STAP is the collection of space-time steering vectors of all grids.

$$\mathbf{D} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K] \quad (9)$$

The signal model in STAP can be cast in the following form

$$\mathbf{X} = \mathbf{D} \mathbf{A} + \mathbf{n} \quad (10)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L]$; $\mathbf{A} \in \mathbb{C}^{K \times L}$ denotes sparse coefficient matrix; non-zero elements indicate the presence of clutter on the space-time profile; L denotes the number of the range gates in the data; and $\mathbf{n} \in \mathbb{C}^{MN \times L}$ denotes zero-mean Gaussian noise.

In sparse signal recovery algorithms with MMV, our goal is to represent the measurement \mathbf{X} , which is contaminated by noise as a linear combination of as few dictionary atoms as possible. Therefore, the objective function is expressed as

$$\mathbf{A}^* = \underset{\mathbf{A}}{\text{argmin}} \|\mathbf{A}\|_{2,0}, \text{ s.t. } \|\mathbf{X} - \mathbf{D} \mathbf{A}\|_F^2 \leq \varepsilon \quad (11)$$

where ε is the noise error allowance.

3. M-SBL-STAP Algorithm

3.1. Sparse Bayesian Learning Formulation

In the SBL framework, $\{\mathbf{x}_l\}_{l=1}^L$ are assumed as i.i.d training snapshots. The noise is submitted to white complex Gaussian distribution with unknown power σ^2 . The likelihood function of the measurement vectors can be denoted as

$$p(\mathbf{X}|\mathbf{A}, \sigma^2) = (\pi\sigma^2)^{-MNL} \exp\left[-\sigma^{-2} \sum_{l=1}^L \|\mathbf{x}_l - \mathbf{D}\mathbf{a}_l\|^2\right] \quad (12)$$

Since the training snapshots are i.i.d, each column in \mathbf{A} is independent and shares the same covariance matrix. Assign to each column in \mathbf{A} with a zero-mean Gaussian prior

$$\mathbf{a}_l \sim \mathcal{CN}(0, \mathbf{\Gamma}), l = 1, 2, \dots, L \quad (13)$$

where $0 \in \mathbb{C}^{K \times 1}$ is a zero vector and $\mathbf{\Gamma} = \text{diag}(\boldsymbol{\gamma})$. $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \dots, \gamma_K\}$ are unknown variance parameters corresponding to sparse coefficient matrix. The prior of \mathbf{A} can be expressed as

$$p(\mathbf{A}|\mathbf{\Gamma}) = \pi^{-KL} \mathbf{\Gamma}^{-L} \exp\left(-\sum_{l=1}^L \mathbf{a}_l^H \mathbf{\Gamma}^{-1} \mathbf{a}_l\right) \quad (14)$$

Combining likelihood with prior, the posterior density of \mathbf{A} can be easily expressed as

$$p(\mathbf{A}|\mathbf{X}, \mathbf{\Gamma}, \sigma^2) = \frac{p(\mathbf{X}|\mathbf{A}, \sigma^2) p(\mathbf{A}|\mathbf{\Gamma})}{\int p(\mathbf{X}|\mathbf{A}, \sigma^2) p(\mathbf{A}|\mathbf{\Gamma}) d\mathbf{A}} \quad (15)$$

modulated by the hyper-parameters $\boldsymbol{\gamma}$ and σ^2 . To find the hyper-parameters $\boldsymbol{\gamma}$ and σ^2 , which are enough accurate to estimate the CNCM, the most common method is the expectation maximization (EM) algorithm. The EM algorithm is divided into two parts: E-step and M-step. t represents the sequence number of the current iteration. At the E-step: treat \mathbf{A} as hidden data, and the posterior density can be described with hyper-parameters $\boldsymbol{\gamma}$ and σ^2 .

$$\begin{aligned} p(\mathbf{A}^{(t+1)}|\mathbf{X}, \mathbf{\Gamma}^{(t)}, \{\sigma^2\}^{(t)}) &= \frac{p(\mathbf{X}, \mathbf{A}^{(t+1)}|\mathbf{\Gamma}^{(t)}, \{\sigma^2\}^{(t)})}{p(\mathbf{X}|\mathbf{\Gamma}^{(t)}, \{\sigma^2\}^{(t)})} \\ &= \pi^{-KL} |\boldsymbol{\Sigma}^{(t+1)}|^{-L} \exp\left[-\sum_{l=1}^L \left(\mathbf{a}_l^{(t+1)} - \boldsymbol{\mu}_l^{(t+1)}\right)^H \left(\boldsymbol{\Sigma}^{(t+1)}\right)^{-1} \left(\mathbf{a}_l^{(t+1)} - \boldsymbol{\mu}_l^{(t+1)}\right)\right] \end{aligned} \quad (16)$$

with covariance and mean given by

$$\boldsymbol{\Sigma}^{(t+1)} = \left(\left\{\sigma^{-2}\right\}^{(t)} \mathbf{D}^H \mathbf{D} + \boldsymbol{\Lambda}^{(t)}\right)^{-1} \quad (17)$$

$$\boldsymbol{\mu}^{(t+1)} = \left\{\sigma^{-2}\right\}^{(t)} \boldsymbol{\Sigma}^{(t+1)} \mathbf{D}^H \mathbf{X} \quad (18)$$

where

$$\boldsymbol{\Lambda}^{(t)} = \left\{\mathbf{\Gamma}^{(t)}\right\}^{-1} = \text{diag}\left(\left[\frac{1}{\gamma_1^{(t)}}, \frac{1}{\gamma_2^{(t)}}, \dots, \frac{1}{\gamma_K^{(t)}}\right]\right) \quad (19)$$

$$\boldsymbol{\mu}^{(t+1)} = \left[\boldsymbol{\mu}_1^{(t+1)}, \boldsymbol{\mu}_2^{(t+1)}, \dots, \boldsymbol{\mu}_L^{(t+1)}\right] \quad (20)$$

At the M-step, we update the hyper-parameters by maximizing the expectation of $p(\mathbf{X}, \mathbf{A}^{(t+1)} | \Gamma^{(t)}, \{\sigma^2\}^{(t)})$.

$$\left[\gamma^{(t+1)}, \{\sigma^2\}^{(t+1)} \right] = \underset{\Gamma^{(t)}, \{\sigma^2\}^{(t)}}{\operatorname{argmax}} \mathbb{E} \left[\ln p(\mathbf{X}, \mathbf{A}^{(t+1)} | \Gamma^{(t)}, \{\sigma^2\}^{(t)}) \right] \quad (21)$$

where

$$\gamma_i^{(t+1)} = \frac{1}{L} \sum_{l=1}^L \left\| \mu_l^{(t+1)}(i) \right\|_2^2 + \Sigma_{ii}^{(t+1)} \quad (22)$$

$$\{\sigma^2\}^{(t+1)} = \frac{\frac{1}{L} \sum_{l=1}^L \left\| \mathbf{x}_l - \mathbf{D} \mu_l^{(t+1)} \right\|_2^2}{NM - K + \sum_{k=1}^K \frac{\Sigma_{kk}^{(t+1)}}{\gamma_k^{(t)}}} \quad (23)$$

The M-SBL-STAP algorithm is shown in Algorithm 1.

Algorithm 1: Pseudocode for the M-SBL-STAP algorithm.

Step 1: Input: the clutter data \mathbf{X} , the dictionary \mathbf{D}

Step 2: Initialization: initial the values of γ and σ^2 .

Step 3: E-step: update the posterior moments $\Sigma^{(t+1)}$ and $\mu^{(t+1)}$ using (17) and (18).

Step 4: M-step: update $\gamma^{(t+1)}$ and $\{\sigma^2\}^{(t+1)}$ using (22) and (23).

Step 5: Repeat step 3 and step 4 until a stopping criterion is satisfied.

Step 6: Estimate the CNCM \mathbf{R} by the formula

$$\mathbf{R} = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K \left(\left\| \mu_{l,k}^* \right\|^2 \right) \mathbf{v}_k \mathbf{v}_k^H + \beta \mathbf{I}$$

where β is a load factor and the symbol * represents the stopping criterion.

Step 7: Compute the space-time adaptive weight \mathbf{w} using (7).

Step 8: The output of the M-SBL-STAP algorithm is $\mathbf{Z} = \mathbf{w}^H \mathbf{X}$.

3.2. Problem Statement of the M-SBL-STAP Algorithm

At the E-step in each iteration, the inversion of a $K \times K$ matrix, which brings high computational complexities in the order of $O(K^3)$, is required to be calculated when update covariance Σ (17). K is the number of the atoms in the dictionary, and the value of K is usually large. If we avoid calculating the matrix inversion of a $K \times K$ matrix, the computational complexities can be reduced a lot.

At the M-step in each iteration, the noise variance σ^2 needs to be estimated. However, it has been demonstrated in [18] that σ^2 estimated by (23) can be extremely inaccurate when the dictionary is structured and $K \geq NM$. We can see in [21] that σ^2 is a nuisance parameter in the iterative procedure and an inappropriate value may contaminate the convergence performance of the algorithm.

We extend a modified hierarchical model to the SBL framework, which aims at integrating σ^2 out instead of estimating σ^2 . However, the modified model, which applies to the real-valued signals, cannot be directly extended to complex-valued signals.

In the next sections, we introduce the proposed algorithm to solve the above problems. The proposed algorithm is defined as M-FMLM-STAP algorithm.

4. The Proposed M-FMLM-STAP Algorithm

4.1. Modified Hierarchical Model

In [16], the scholars define the following hierarchical hyperpriors over Γ , as well as the noise variance σ^2 .

$$p(\Gamma | c, d) = \prod_{k=1}^K \operatorname{Gamma} \left(\frac{1}{\gamma_k} \middle| c, d \right) \quad (24)$$

$$p(\sigma^2 | a, b) = \text{Gamma}(\sigma^{-2} | a, b) \quad (25)$$

where $\text{Gamma}(a | a, b) = \Gamma(a)^{-1} b^a a^{a-1} e^{-ba}$ and the ‘gamma function’ $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$. It has been demonstrated in [16] that an appropriate choice of the shape parameter a and the scale parameter b encourages the sparsity of the coefficient matrix.

In the STAP framework, CNCM can be expressed as

$$\mathbf{R}_{c+n} = \mathbf{D}\mathbf{\Gamma}\mathbf{D} + \sigma^2 \mathbf{I} \quad (26)$$

We can translate STAP weight vector in the form

$$\begin{aligned} \mathbf{w}_{\text{opt}} &= \mu \mathbf{R}_{c+n}^{-1} \mathbf{S}_t \\ &= \lambda \left(\mathbf{D} \frac{\mathbf{\Gamma}}{\sigma^2} \mathbf{D} + \mathbf{I} \right)^{-1} \mathbf{S}_t \\ &= \lambda \tilde{\mathbf{R}}^{-1} \mathbf{S}_t \end{aligned} \quad (27)$$

where λ and μ are constants.

The above analysis shows that $\tilde{\mathbf{R}}$ is equivalent to \mathbf{R}_{c+n} in the performance of clutter suppression. Thus, including σ^2 in the prior of $\mathbf{\Gamma}$, each component of \mathbf{A} is defined as a zero-mean Gaussian distribution and the modified hierarchical model follows (28) and (29).

$$p(\mathbf{A} | \mathbf{\Gamma}', \sigma^2) = \prod_{l=1}^L \prod_{k=1}^K \mathcal{CN}(a_{l,k} | 0, \gamma_k) = \prod_{l=1}^L \prod_{k=1}^K \mathcal{CN}(a_{l,k} | 0, \sigma^2 \gamma'_k) \quad (28)$$

$$p(\sigma^2 | a, b) = \text{Gamma}(\sigma^{-2} | a, b) \quad (29)$$

where $\mathbf{\Gamma}' = \text{diag}(\mathbf{\gamma}') = \text{diag}(\gamma'_1, \gamma'_2, \dots, \gamma'_{2K}) = \frac{\mathbf{\Gamma}}{\sigma^2}$.

4.2. Application of the Modified Hierarchical Model to Complex-Valued Signals

However, although both of the original and the modified models apply to real-valued signals, they cannot be directly extended to complex-valued signals. To solve the above problem, the real and imaginary components of the complex-valued signals are treated as two independent real-valued variables.

$$\mathbf{y}_l = \begin{bmatrix} \text{Re}(\mathbf{x}_l) \\ \text{Im}(\mathbf{x}_l) \end{bmatrix} \quad (30)$$

The new signal model is expressed as

$$\mathbf{Y} = \mathbf{\Psi}\mathbf{B} + \mathbf{N} \quad (31)$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L] \in \mathbb{R}^{2MN \times L}$ is the new data model, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L] \in \mathbb{R}^{2K \times L}$ is the new sparse coefficient matrix and $\mathbf{N} \in \mathbb{R}^{2MN \times L}$ is the new noise matrix in new model. The new dictionary $\mathbf{\Psi} \in \mathbb{R}^{2MN \times 2K}$ is expressed as

$$\mathbf{\Psi} = \begin{bmatrix} \text{Re}(\mathbf{v}_1) & -\text{Im}(\mathbf{v}_1) & \cdots & \text{Re}(\mathbf{v}_K) & -\text{Im}(\mathbf{v}_K) \\ \text{Im}(\mathbf{v}_1) & \text{Re}(\mathbf{v}_1) & \cdots & \text{Im}(\mathbf{v}_K) & \text{Re}(\mathbf{v}_K) \end{bmatrix} \quad (32)$$

while the new covariance, mean and hyper-parameter in the modified model are expressed as

$$\mathbf{\Sigma} = \left(\sigma^{-2} \mathbf{\Psi}^T \mathbf{\Psi} + \mathbf{\Gamma}^{-1} \right)^{-1} \quad (33)$$

$$\boldsymbol{\mu} = \sigma^{-2} \mathbf{\Sigma} \mathbf{\Psi}^T \mathbf{y} \quad (34)$$

$$\mathbf{\Gamma} = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_K, \gamma_{K+1}, \gamma_{K+2}, \dots, \gamma_{2K}) \quad (35)$$

where $\gamma_{2i-1} = \gamma_{2i}$, $i = 1, 2, \dots, K$. The signal \mathbf{y}_l is group sparse because the entries in each group, saying $\gamma_{\{i\}} = \{\gamma_{2i-1}, \gamma_{2i}\}$, $i = 1, 2, \dots, K$, are either zero or nonzero simultaneously.

In order to avoid calculating σ^2 in (33) and (34), we extend the modified hierarchical model to the SBL framework. The new real-valued likelihood function can be expressed as

$$\begin{aligned} p(\mathbf{Y}|\mathbf{B}, \sigma^2) &= (2\pi\sigma^2)^{-\frac{2MNL}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{l=1}^L \|\mathbf{y}_l - \mathbf{\Psi}\mathbf{b}_l\|^2\right] \\ &= (2\pi\sigma^2)^{-MNL} \exp\left[-\frac{1}{2\sigma^2} \sum_{l=1}^L \|\mathbf{y}_l - \mathbf{\Psi}\mathbf{b}_l\|^2\right] \end{aligned} \quad (36)$$

and the new prior of \mathbf{B} can be expressed as

$$\begin{aligned} p(\mathbf{B}|\mathbf{\Gamma}', \sigma^2) &= (2\pi)^{-\frac{2KL}{2}} \mathbf{\Gamma}'^{-\frac{L}{2}} \exp\left(-\frac{1}{2} \sum_{l=1}^L \mathbf{b}_l^T \mathbf{\Gamma}'^{-1} \mathbf{b}_l\right) \\ &= (2\pi)^{-KL} (\sigma^2 \mathbf{\Gamma}')^{-\frac{L}{2}} \exp\left(-\frac{1}{2} \sum_{l=1}^L \mathbf{b}_l^T (\sigma^2 \mathbf{\Gamma}')^{-1} \mathbf{b}_l\right) \end{aligned} \quad (37)$$

Combining the new likelihood and prior, the new posterior density of \mathbf{B} can be expressed as

$$p(\mathbf{B}|\mathbf{Y}, \mathbf{\Gamma}', \sigma^2) = \frac{p(\mathbf{Y}|\mathbf{B}, \sigma^2) p(\mathbf{B}|\mathbf{\Gamma}', \sigma^2)}{\int p(\mathbf{Y}|\mathbf{B}, \sigma^2) p(\mathbf{B}|\mathbf{\Gamma}', \sigma^2) d\mathbf{B}} \quad (38)$$

Integrate σ^2 out, and the posterior density function of \mathbf{B} is

$$\begin{aligned} p(\mathbf{B}|\mathbf{Y}, \mathbf{\Gamma}') &= \int p(\mathbf{B}|\mathbf{Y}, \mathbf{\Gamma}', \sigma^2) p(\sigma^2|a, b) d\sigma^2 \\ &= \frac{\Gamma(a+LK) \left[1 + \frac{1}{2b} \sum_{l=1}^L (\mathbf{a}_l - \boldsymbol{\mu}_l)^H \boldsymbol{\Sigma}^{-1} (\mathbf{a}_l - \boldsymbol{\mu}_l)\right]^{-(a+LK)}}{\Gamma(a)(2\pi b)^{LK} |\boldsymbol{\Sigma}|^{\frac{L}{2}}} \end{aligned} \quad (39)$$

where

$$\boldsymbol{\Sigma} = \left(\mathbf{\Psi}^T \mathbf{\Psi} + \mathbf{\Gamma}'^{-1}\right)^{-1} \quad (40)$$

$$\boldsymbol{\mu}_l = \boldsymbol{\Sigma} \mathbf{\Psi}^T \mathbf{y}_l \quad (41)$$

From (39), we can draw the conclusion that the modified formulation induces a heavy-tailed Student-t distribution on the residual noise, which improves the robustness of the algorithm.

The logarithm of the marginal likelihood function is

$$\begin{aligned} \mathcal{L}(\mathbf{\Gamma}') &= \log p(\mathbf{Y}|\mathbf{\Gamma}') \\ &= \log \int p(\mathbf{Y}|\mathbf{B}, \sigma^2) p(\mathbf{B}|\mathbf{\Gamma}', \sigma^2) p(\sigma^2|a, b) d\mathbf{B} d\sigma^2 \\ &= -\frac{1}{2} \sum_{l=1}^L \left[(2MN + 2a) \log(\mathbf{y}_l^T \mathbf{C}^{-1} \mathbf{y}_l + 2b) + \log|\mathbf{C}| \right] + \text{constant} \end{aligned} \quad (42)$$

with

$$\mathbf{C} = \mathbf{I}_{2MN} + \mathbf{\Psi} \mathbf{\Gamma}' \mathbf{\Psi}^T \quad (43)$$

Since the signals \mathbf{y}_l , $l = 1 \dots, L$ is group-sparse, we define

$$\alpha_i = \gamma'_{2i-1} = \gamma'_{2i}, i = 1, 2, \dots, K \quad (44)$$

and then

$$\mathbf{\Gamma}' = \text{diag}(\boldsymbol{\alpha}) \otimes \mathbf{I}_2 \quad (45)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_K]$.

4.3. Maximization of $\mathcal{L}(\Gamma')$ to Estimate α

A most probable way to point estimate hyper-parameters α may be found via a type-II maximum likelihood procedure. Mathematically, maximize the marginal likelihood function or its logarithm (42) with respect to α . In the following substance, the FMLM algorithm is applied to maximize (42) to estimate α . Unlike the EM algorithm, the FMLM algorithm reduces the computational complexities by identifying the support space of data, i.e., the atoms in the dictionary whose corresponding values in α is non-zero.

For notational convenience, we ignore the symbol γ' . γ and Γ in the following substance all represent γ' and Γ' , respectively.

Define Ω as the set of the non-zero values in Γ and ψ as the support space of data.

$$\Omega = \text{diag}(\alpha_{\omega_1}, \alpha_{\omega_2}, \dots, \alpha_{\omega_J}) \otimes \mathbf{I}_2 \quad (46)$$

$$\psi = [\Phi_{\omega_1}, \Phi_{\omega_2}, \dots, \Phi_{\omega_J}] \quad (47)$$

where

$$\{\omega_1, \omega_2, \dots, \omega_J\} = \{i | 1 \leq i \leq K, \alpha_i \neq 0\} \quad (48)$$

$$\Phi_i = \begin{bmatrix} \text{Re}(\mathbf{v}_i) & -\text{Im}(\mathbf{v}_i) \\ \text{Im}(\mathbf{v}_i) & \text{Re}(\mathbf{v}_i) \end{bmatrix}, i = 1, 2, \dots, K \quad (49)$$

and J is the number of non-zeros in α .

At the beginning of the FMLM algorithm, we initialize $\alpha^{(0)} = 0$, namely, $\Omega^{(0)} = \emptyset$ and $\psi^{(0)} = \emptyset$. Then, we need to identify the support space of data in each iteration until that α converges to a steady point.

The matrix \mathbf{C} can be decomposed into two parts.

$$\begin{aligned} \mathbf{C} &= \mathbf{I}_{2MN} + \sum_{k \neq i, 1 \leq k \leq K} \alpha_k \Phi_k \Phi_k^T + \alpha_i \Phi_i \Phi_i^T \\ &= \mathbf{C}_{-i} + \alpha_i \Phi_i \Phi_i^T \end{aligned} \quad (50)$$

The first term $\mathbf{C}_{-i} = \mathbf{I}_{2MN} + \sum_{k \neq i, 1 \leq k \leq K} \alpha_k \Phi_k \Phi_k^T$ contains all terms that are independent of α_i , and the second term includes all the terms related to α_i .

Using the Woodbury Matrix Identity, the matrix inversion and matrix determinate lemmas are expressed as

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \mathbf{C}_{-i}^{-1} \Phi_i (\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i)^{-1} \Phi_i^T \mathbf{C}_{-i}^{-1} \quad (51)$$

$$|\mathbf{C}| = |\mathbf{C}_{-i}| |\alpha_i \mathbf{I}_2| |\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i| \quad (52)$$

Then, (42) can be expressed as

$$\begin{aligned} \mathcal{L}(\Gamma) &= -\frac{1}{2} \sum_{l=1}^L \left[(2MN + 2a) \log(\mathbf{y}_l^T \mathbf{C}_{-i}^{-1} \mathbf{y}_l + 2b) + \log|\mathbf{C}_{-i}| \right] \\ &\quad - \frac{1}{2} \sum_{l=1}^L \left[(2MN + 2a) \log \left(1 - \frac{\mathbf{y}_l^T \mathbf{C}_{-i}^{-1} \Phi_i (\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i)^{-1} \Phi_i^T \mathbf{C}_{-i}^{-1} \mathbf{y}_l}{\mathbf{y}_l^T \mathbf{C}_{-i}^{-1} \mathbf{y}_l + 2b} \right) \right] + \text{constant} \\ &\quad + \log|\alpha_i \mathbf{I}_2| + \log|\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i| \\ &= \mathcal{L}_{-i} + \mathcal{L}(\alpha_i) \end{aligned} \quad (53)$$

where \mathcal{L}_{-i} contains all terms that are independent of α_i and $\mathcal{L}(\alpha_i)$ includes all the terms related to α_i .

Define

$$\hat{\mathbf{S}}_i \triangleq \mathbf{\Phi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{\Phi}_i, \hat{\mathbf{q}}_{l,i} \triangleq \mathbf{\Phi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{y}_l, \hat{g}_{l,i} \triangleq \mathbf{y}_l^T \mathbf{C}_{-i}^{-1} \mathbf{y}_l + 2b \quad (54)$$

Then, $\mathcal{L}(\alpha_i)$ can be expressed as

$$\mathcal{L}(\alpha_i) = -\frac{1}{2} \sum_{l=1}^L \left[(2MN + 2a) \log \left(1 - \frac{\hat{\mathbf{q}}_{l,i}^T (\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i)^{-1} \hat{\mathbf{q}}_{l,i}}{\hat{g}_{l,i}} \right) + 2 \log |\alpha_i| + \log |\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i| \right] \quad (55)$$

The eigenvalue decomposition (EVD) of $\hat{\mathbf{S}}_i$

$$\hat{\mathbf{S}}_i = \mathbf{V}_i \mathbf{\Lambda}_i \mathbf{V}_i^T = \sum_{j=1}^2 \hat{s}_{i,j} \mathbf{v}_{i,j} \mathbf{v}_{i,j}^T \quad (56)$$

where $\hat{s}_{i,j}$ and $\mathbf{v}_{i,j}$ denote the j -th eigenvalue and eigenvector of $\hat{\mathbf{S}}_i$ respectively. $\mathbf{V}_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}]$ is the eigen-matrix of $\hat{\mathbf{S}}_i$.

Define $\mathbf{c}_{l,i} = \mathbf{V}_i^T \hat{\mathbf{q}}_{l,i} = \begin{bmatrix} c_{l,i,1} \\ c_{l,i,2} \end{bmatrix}$, and the formula (55) can then be expressed as

$$\mathcal{L}(\alpha_i) = -\frac{1}{2} \sum_{l=1}^L \left[(2MN + 2a) \log \left(1 - \frac{\frac{\alpha_i^2 c_{l,i,1}^2}{1 + \alpha_i \hat{s}_{i,1}} + \frac{\alpha_i^2 c_{l,i,2}^2}{1 + \alpha_i \hat{s}_{i,2}}}{\hat{g}_{l,i}} \right) + 2 \log(\alpha_i) + \sum_{j=1}^2 \log \left(\frac{1 + \alpha_i \hat{s}_{i,j}}{\alpha_i} \right) \right] \quad (57)$$

The next step involves maximizing (57) to estimate the hyper-parameters $\alpha_i, \forall i$, and we choose only one candidate from $\alpha_i, \forall i$ that can maximize (53).

Differentiate $\mathcal{L}(\alpha_i)$ with respect to α_i , and set the result to zero

$$\frac{\partial \mathcal{L}(\alpha_i)}{\partial \alpha_i} = -\frac{1}{2} \sum_{l=1}^L \left\{ \left(\frac{-(2MN + 2a) \left(\frac{c_{l,i,1}^2}{(1 + \alpha_i \hat{s}_{i,1})^2} + \frac{c_{l,i,2}^2}{(1 + \alpha_i \hat{s}_{i,2})^2} \right)}{\hat{g}_{l,i} - \frac{\alpha_i c_{l,i,1}^2}{1 + \alpha_i \hat{s}_{i,1}} - \frac{\alpha_i c_{l,i,2}^2}{1 + \alpha_i \hat{s}_{i,2}}} \right) + \frac{\hat{s}_{i,1}}{1 + \alpha_i \hat{s}_{i,1}} + \frac{\hat{s}_{i,2}}{1 + \alpha_i \hat{s}_{i,2}} \right\} \quad (58)$$

which is equivalent to solve the following polynomial function

$$f(\alpha_i) = d_3 \alpha_i^3 + d_2 \alpha_i^2 + d_1 \alpha_i + d_0 = 0 \quad (59)$$

where

$$\begin{aligned} d_0 &= \sum_{l=1}^L \left\{ \hat{g}_{l,i} (\hat{s}_{i,1} + \hat{s}_{i,2}) - (2MN + 2a) (c_{l,i,1}^2 + c_{l,i,2}^2) \right\} \\ d_1 &= \sum_{l=1}^L \left\{ 4 \hat{g}_{l,i} \hat{s}_{i,1} \hat{s}_{i,2} + \hat{g}_{l,i} (\hat{s}_{i,1}^2 + \hat{s}_{i,2}^2) - (c_{l,i,1}^2 + c_{l,i,2}^2) (\hat{s}_{i,1} + \hat{s}_{i,2}) - 2(2MN + 2a) (c_{l,i,1}^2 \hat{s}_{i,2} + c_{l,i,2}^2 \hat{s}_{i,1}) \right\} \\ d_2 &= \sum_{l=1}^L \left\{ 3 \hat{g}_{l,i} \hat{s}_{i,1}^2 \hat{s}_{i,2} + 3 \hat{g}_{l,i} \hat{s}_{i,1} \hat{s}_{i,2}^2 - 3 c_{l,i,1}^2 \hat{s}_{i,1} \hat{s}_{i,2} - 3 c_{l,i,2}^2 \hat{s}_{i,1} \hat{s}_{i,2} - c_{l,i,1}^2 \hat{s}_{i,2}^2 - c_{l,i,2}^2 \hat{s}_{i,1}^2 - (2MN + 2a + 1) c_{l,i,1}^2 \hat{s}_{i,2}^2 - (2MN + 2a + 1) c_{l,i,2}^2 \hat{s}_{i,1}^2 \right\} \\ d_3 &= \sum_{l=1}^L \left\{ 2 \hat{g}_{l,i} \hat{s}_{i,1}^2 \hat{s}_{i,2}^2 - 2 c_{l,i,1}^2 \hat{s}_{i,1} \hat{s}_{i,2}^2 - 2 c_{l,i,2}^2 \hat{s}_{i,1}^2 \hat{s}_{i,2} \right\} \end{aligned} \quad (60)$$

It has at most 3 roots found by standard root-finding algorithms. Considering that $\alpha_i \geq 0$ and the corresponding $\mathbf{\Phi}_i$ is not in the support space when $\alpha_i = 0$, the set of the positive roots is

$$\Theta = \{\alpha_i | \alpha_i \in \mathbb{R}^+, f(\alpha_i) = 0\} \quad (61)$$

The optimal α_i^* is given by

$$\alpha_i^* = \begin{cases} 0 & \text{if } \max_{\alpha_i \in \Theta} \mathcal{L}(\alpha_i) < 0 \\ \operatorname{argmax}_{\alpha_i \in \Theta} \mathcal{L}(\alpha_i) & \text{else} \end{cases} \quad (62)$$

If $\alpha_i^* = 0$, it means the set $\{\alpha_i, \Phi_i\}$ has no contribution to $\mathcal{L}(\Gamma)$, that is to say, $\mathcal{L}(\alpha_i^*) = 0$. If $\max_{\alpha_i \in \Theta} \mathcal{L}(\alpha_i) < 0$, we need to set $\alpha_i^* = 0$ because $\mathcal{L}(0) = 0 > \max_{\alpha_i \in \Theta} \mathcal{L}(\alpha_i)$.

In the $(t+1)$ -th iteration, we choose only one $\alpha_j, 1 \leq j \leq K$ that can maximize (53), while fix the other $\{\alpha_i | 1 \leq i \leq K, i \neq j\}$. The sequence number j is expressed as

$$j = \operatorname{argmax}_{i, 1 \leq i \leq K} \mathcal{L}(\Gamma) = \operatorname{argmax}_{i, 1 \leq i \leq K} \mathcal{L}_{-i} + \mathcal{L}(\alpha_i^*) \quad (63)$$

In order to avoid calculating $\mathcal{L}_{-i}, \forall i$, we define

$$\Delta \mathcal{L}(i) \triangleq \mathcal{L}_{-i} + \mathcal{L}(\alpha_i^*) - \mathcal{L}(\Gamma^{(t)}) = \mathcal{L}_{-i} + \mathcal{L}(\alpha_i^*) - \mathcal{L}_{-i} - \mathcal{L}(\alpha_i^{(t)}) = \mathcal{L}(\alpha_i^*) - \mathcal{L}(\alpha_i^{(t)}) \quad (64)$$

Since $\mathcal{L}(\Gamma^{(t)})$ in the $(t+1)$ -th iteration is a fixed constant, the sequence number j can also be expressed as

$$j = \operatorname{argmax}_{i, 1 \leq i \leq K} \Delta \mathcal{L}(i) = \operatorname{argmax}_{i, 1 \leq i \leq K} \mathcal{L}(\alpha_i^*) - \mathcal{L}(\alpha_i^{(t)}) \quad (65)$$

The next step is to change the value of α_j while fixing the other $\{\alpha_i | 1 \leq i \leq K, i \neq j\}$.

$$\alpha_j^{(t+1)} = \alpha_j^* \quad (66)$$

$$\alpha_i^{(t+1)} = \alpha_i^{(t)}, 1 \leq i \leq K, i \neq j \quad (67)$$

If $\alpha_j^{(t+1)} > 0$ and $\alpha_j^{(t)} = 0$, add Φ_j to $\Psi^{(t)}$ (i.e., Φ_j is not in $\Psi^{(t)}$, $\Psi^{(t+1)} = [\Psi^{(t)}, \Phi_j]$, $\Omega^{(t+1)} = \begin{bmatrix} \Omega^{(t)} \\ \alpha_j^{(t+1)} \otimes \mathbf{I}_2 \end{bmatrix}$); if $\alpha_j^{(t+1)} > 0$ and $\alpha_j^{(t)} > 0$ (i.e., Φ_j is already in $\Psi^{(t)}$, $\Psi^{(t+1)} = \Psi^{(t)}$), replace $\alpha_j^{(t)}$ with $\alpha_j^{(t+1)}$ in $\Omega^{(t)}$; if $\alpha_j^{(t+1)} = 0$ and $\alpha_j^{(t)} > 0$, delete Φ_j from $\Psi^{(t)}$ and delete $\alpha_j^{(t)}$ from $\Omega^{(t)}$; and if $\alpha_j^{(t+1)} = 0$ and $\alpha_j^{(t)} = 0$ (i.e., $\Gamma^{(t+1)} = \Gamma^{(t)}$), stop iteration because the procedure converges to steady state. Finally, Ψ^* is the exact support space of the data and Ω^* is the set of the non-zero values in the exact Γ where the symbol $*$ represents the stopping criterion.

4.4. Fast Computation of $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$

In each iteration, we need to calculate the matrix inversion of all $\mathbf{C}_{-i}, \forall i$ when updating $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$. In order to reduce computation complexities, we need a fast means to update $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$.

Define

$$\mathbf{S}_i \triangleq \Phi_i^T \mathbf{C}^{-1} \Phi_i, \mathbf{q}_{l,i} \triangleq \Phi_i^T \mathbf{C}^{-1} \mathbf{y}_l, g_l \triangleq \mathbf{y}_l^T \mathbf{C}^{-1} \mathbf{y}_l + 2b \quad (68)$$

If we can calculate $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$ with $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$, the computational complexities can be reduced a lot because we only calculate the matrix inversion of \mathbf{C} .

Substituting (51) into (68), we can arrive at the following formula:

$$\begin{aligned}
 \mathbf{S}_i &= \Phi_i^T \mathbf{C}^{-1} \Phi_i \\
 &= \Phi_i^T \left(\mathbf{C}_{-i}^{-1} - \mathbf{C}_{-i}^{-1} \Phi_i \left(\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i \right)^{-1} \Phi_i^T \mathbf{C}_{-i}^{-1} \right) \Phi_i \\
 &= \hat{\mathbf{S}}_i - \hat{\mathbf{S}}_i \left(\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i \right)^{-1} \hat{\mathbf{S}}_i \\
 &= \mathbf{V}_i \begin{bmatrix} \frac{\hat{S}_{i,1}}{1 + \alpha_i \hat{S}_{i,1}} & \\ & \frac{\hat{S}_{i,2}}{1 + \alpha_i \hat{S}_{i,2}} \end{bmatrix} \mathbf{V}_i^T
 \end{aligned} \tag{69}$$

We can draw the conclusion that the eigen-matrix of \mathbf{S}_i is the same as that of $\hat{\mathbf{S}}_i$ from (56) and (69). The EVD of \mathbf{S}_i can be also expressed as

$$\mathbf{S}_i = \mathbf{V}_i \begin{bmatrix} S_{i,1} & \\ & S_{i,2} \end{bmatrix} \mathbf{V}_i^T \tag{70}$$

where $S_{i,j}$ denotes the j -th eigenvalue of \mathbf{S}_i .

$$S_{i,j} = \frac{\hat{S}_{i,j}}{1 + \alpha_i \hat{S}_{i,j}}, j = 1, 2 \tag{71}$$

Thus, $\hat{\mathbf{S}}_i$ can be obtained via the EVD of \mathbf{S}_i

$$\hat{\mathbf{S}}_i = \mathbf{V}_i \begin{bmatrix} \frac{S_{i,1}}{1 - \alpha_i S_{i,1}} & \\ & \frac{S_{i,2}}{1 - \alpha_i S_{i,2}} \end{bmatrix} \mathbf{V}_i^T \tag{72}$$

$\hat{\mathbf{q}}_{l,i}$ and $\hat{g}_{l,i}$ can be also computed with $\mathbf{q}_{l,i}$ and g_l , respectively, via the EVD of \mathbf{S}_i .

$$\begin{aligned}
 \mathbf{q}_{l,i} &= \Phi_i^T \mathbf{C}^{-1} \mathbf{y}_l \\
 &= \Phi_i^T \left(\mathbf{C}_{-i}^{-1} - \mathbf{C}_{-i}^{-1} \Phi_i \left(\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i \right)^{-1} \Phi_i^T \mathbf{C}_{-i}^{-1} \right) \mathbf{y}_l \\
 &= \hat{\mathbf{q}}_{l,i} - \hat{\mathbf{S}}_i \left(\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i \right)^{-1} \hat{\mathbf{q}}_{l,i} \\
 &= \left(\mathbf{I}_2 - \hat{\mathbf{S}}_i \left(\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i \right)^{-1} \right) \hat{\mathbf{q}}_{l,i}
 \end{aligned} \tag{73}$$

Thus,

$$\hat{\mathbf{q}}_{l,i} = \left(\mathbf{I}_2 - \hat{\mathbf{S}}_i \left(\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i \right)^{-1} \right)^{-1} \mathbf{q}_{l,i} = \mathbf{V}_i \begin{bmatrix} \frac{1}{1 - \alpha_i S_{i,1}} & \\ & \frac{1}{1 - \alpha_i S_{i,2}} \end{bmatrix} \mathbf{V}_i^T \mathbf{q}_{l,i} \tag{74}$$

Similarly,

$$\begin{aligned}
 g_l &= \mathbf{y}_l^T \left(\mathbf{C}_{-i}^{-1} - \mathbf{C}_{-i}^{-1} \Phi_i \left(\alpha_i^{-1} \mathbf{I}_2 + \Phi_i^T \mathbf{C}_{-i}^{-1} \Phi_i \right)^{-1} \Phi_i^T \mathbf{C}_{-i}^{-1} \right) \mathbf{y}_l + 2b \\
 &= \hat{g}_{l,i} - \hat{\mathbf{q}}_{l,i}^T \left(\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i \right)^{-1} \hat{\mathbf{q}}_{l,i} \\
 &= \hat{g}_{l,i} - \hat{\mathbf{q}}_{l,i}^T \mathbf{V}_i \begin{bmatrix} \alpha_i (1 - \alpha_i S_{i,1}) & \\ & \alpha_i (1 - \alpha_i S_{i,2}) \end{bmatrix} \mathbf{V}_i^T \hat{\mathbf{q}}_{l,i}
 \end{aligned} \tag{75}$$

Using (74), we can obtain

$$\hat{g}_{l,i} = g_l + \mathbf{q}_{l,i}^T \mathbf{V}_i \begin{bmatrix} \frac{\alpha_i}{1 - \alpha_i S_{i,1}} & \\ & \frac{\alpha_i}{1 - \alpha_i S_{i,2}} \end{bmatrix} \mathbf{V}_i^T \mathbf{q}_{l,i} \tag{76}$$

Therefore, $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$ can be obtained from $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$ using (72), (74) and (76).

Compared with (68), there is another approach that requires fewer computational complexities to calculate $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$. The formula (43) is equivalent to

$$\mathbf{C} = \mathbf{I}_{2MN} + \boldsymbol{\Psi}\boldsymbol{\Omega}\boldsymbol{\Psi}^T \quad (77)$$

With matrix inversion lemmas and (77), it is fast and more convenient to update $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$ with the following formulae compared with (68).

$$\begin{aligned} \mathbf{S}_i &= \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i - \boldsymbol{\Phi}_i^T \boldsymbol{\Psi} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \boldsymbol{\Phi}_i \\ \mathbf{q}_{l,i} &= \boldsymbol{\Phi}_i^T \mathbf{y}_l - \boldsymbol{\Phi}_i^T \boldsymbol{\Psi} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \mathbf{y}_l \\ g_l &= \mathbf{y}_l^T \mathbf{y}_l - \mathbf{y}_l^T \boldsymbol{\Psi} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \mathbf{y}_l + 2b \end{aligned} \quad (78)$$

where $\boldsymbol{\Sigma}$ herein represents the covariance of the non-zeros in $\boldsymbol{\Gamma}$.

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \{\boldsymbol{\Omega}\}^{-1} \right)^{-1} \quad (79)$$

Additionally, the mean of the non-zeros in $\boldsymbol{\Gamma}$ herein is expressed as

$$\mu_l = \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \mathbf{y}_l, l = 1, \dots, L \quad (80)$$

The computational complexities are measured in terms of the number of multiplications. Assuming that the dimension of $\boldsymbol{\Psi}$ is $2MN \times r$, r is not a fixed value in each iteration but satisfies the condition $r \ll MN < K$ because the measurements are sparse. When we calculate \mathbf{S}_i with (68), the computation complexities are in order of $O((2MN)^3 + 8(MN)^2 + 8MN)$. With (78), the computation complexities of calculating \mathbf{S}_i are in order of $O(r^3 + 16MN + 8MNr + 2r^2)$. It is apparent that the latter operation is faster and more convenient.

Since the measurements are sparse, the dimension of the matrix $\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \{\boldsymbol{\Omega}\}^{-1}$ in (79) is far smaller than the dimension of $\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \boldsymbol{\Gamma}'^{-1}$ in (41). The proposed M-FMLM-STAP algorithm identifies the support space of data to reduce the effective problem dimensions drastically due to sparsity. Therefore, the proposed algorithm has tremendous potential for real-time operation.

The proposed M-FMLM-STAP algorithm is shown in Algorithm 2. The detailed update formulae are shown in the Appendix A.

The main symbols aforementioned are listed in Table 1.

Table 1. The main symbols.

Symbols	Parameters	Symbols	Parameters
\mathbf{X}	The original data	\mathbf{Y}	The new data
\mathbf{D}	The original dictionary	$\boldsymbol{\Psi}$	The new dictionary
\mathbf{A}	The original coefficient matrix	\mathbf{B}	The new coefficient matrix
$\boldsymbol{\Gamma}$	The original hyper-parameter	$\boldsymbol{\Gamma}'$	The new hyper-parameter
$\boldsymbol{\Sigma}$	The covariance of coefficient	μ	The mean of coefficient
$\boldsymbol{\Omega}$	The set of the non-zero values in $\boldsymbol{\Gamma}'$	$\boldsymbol{\Psi}$	The support space of data
$\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$	See (54)	$\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$	See (68)
$\boldsymbol{\Phi}_i$	See (49)		

Algorithm 2: Pseudocode for M-FMLM-STAP algorithm.

Step 1: Input: the original data \mathbf{X} , the original dictionary \mathbf{D} and $a = b = 0$.

Step 2: $\mathbf{y} = [\text{Re}(\mathbf{X}); \text{Im}(\mathbf{X})]$ and Ψ .

Step 3: Initialize: $\mathbf{C} = \mathbf{C}_{-i} = \mathbf{I}$ $\alpha = 0$ and $\psi = \emptyset$.

Step 4: while not converged do

 Choose only one candidate $j \in \{1, 2, \dots, K\}$ and find optimal α_j using (65).

 If $\alpha_j^{(t+1)} > 0$ and $\alpha_j^{(t)} = 0$, then

$$\psi^{(t+1)} = [\psi^{(t)}, \Phi_j], \text{ and } \Omega^{(t+1)} = \begin{bmatrix} \Omega^{(t)} & \\ & \alpha_j^{(t+1)} \otimes \mathbf{I}_2 \end{bmatrix}.$$

 Otherwise, if $\alpha_j^{(t+1)} > 0$ and $\alpha_j^{(t)} > 0$, then

$$\psi^{(t+1)} = \psi^{(t)}, \text{ and replace } \alpha_j^{(t)} \text{ with } \alpha_j^{(t+1)}.$$

 Otherwise, if $\alpha_j^{(t+1)} > 0$ and $\alpha_j^{(t)} > 0$, then

$$\text{delete } \Phi_j \text{ from } \psi^{(t)}, \text{ and delete } \alpha_j^{(t)} \text{ from } \Omega^{(t)}.$$

 end

 Update $\Sigma, \mu, \mathbf{S}_i, \mathbf{q}_{l,i}$ and g_l referring to Appendix A.

 end while

Step 5: Estimate the CNCM \mathbf{R} by

$$\mathbf{R} = \frac{1}{L} \sum_{l=1}^L \sum_{i=1}^J \left(\|\mu_{l,2i-1}^*\|^2 + \|\mu_{l,2i}^*\|^2 \right) \mathbf{v}_{\omega_i} \mathbf{v}_{\omega_i}^H + \beta \mathbf{I}$$

 where the vector \mathbf{v}_{ω_i} is the ω_i -th column of \mathbf{D} , J is the number of non-zeros in α^*

 and β is a load factor. The symbol * represents the stopping criterion.

Step 6: Compute the space-time adaptive weight \mathbf{w} using (7).

Step 7: The output of M-FMLM-STAP is $\mathbf{Z} = \mathbf{w}^H \mathbf{X}$.

5. Complexity Analysis and Convergence Analysis

5.1. Complexity Analysis

Here, we compare the computational complexities of the proposed M-FMLM-STAP algorithm and the M-SBL-STAP algorithm for a single iteration. The computational complexities are measured in terms of the number of multiplications.

First, we analyze the computational complexities of the M-SBL-STAP algorithm. It is apparent that the main computational complexities of the M-SBL-STAP algorithm are related to the formulae (17) and (18). Noting that Λ in (17) is a diagonal matrix, the computational complexities of (17) are in the order of $o(K^3 + K^2MN)$. The computational complexities of (18) are in the order of $o(K^2MN + KMN L)$. Thus, the computational complexities of the M-SBL-STAP algorithm are in the order of $o(K^3 + 2K^2MN + KMN L)$.

Second, we analyze the computational complexities of the M-FMLM-STAP algorithm. Noting that the dimension of \mathbf{S}_i is 2×2 , the computational complexities of the EVD of \mathbf{S}_i are small enough to be ignored, that is to say, the computational complexities to calculate $\{\hat{\mathbf{S}}_i, \hat{\mathbf{q}}_{l,i}, \hat{g}_{l,i}\}$ with $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l\}$ can be ignored. Meanwhile, the computational complexities to solve the polynomial function (59), $\forall i, 1 \leq i \leq K$ and to find the sequence number (65) are in direct proportion to K , which are also small enough to be ignored. Thus, the main computational complexities of the M-FMLM-STAP algorithm are used to update $\{\mathbf{S}_i, \mathbf{q}_{l,i}, g_l, \Sigma, \mu\}$. Assuming that the dimension of ψ is $2MN \times r$, r is not a fixed value in all iterations. However, the condition $r \ll MN < K$ is satisfied because the measurements are sparse. The computational complexities of the term $\Phi_i^T - \Phi_i^T \psi \Sigma \psi^T$ are in the order of $o(8MNr + 2r^2)$. Thus, $\forall i$ and $\forall l$, the computational complexities of (78), are in the order of $o((8MNr + 2r^2)K + 8MNK + 4MN LK + (4MNr + r^2 + 4MN)L)$. The computational complexities of (79) and (80) are in the order of $o(2MNr^2 + r + r^3)$ and $o(2MNr^2 + 2MNrL)$, respectively. To sum up, ignore the low-order computational complexities, and the computational complexities of the M-FMLM-STAP algorithm for a single iteration are in the order of $o(8MNrK + 4MN LK + 6MNrL)$.

Figure 1 illustrates the complexity requirements of two algorithms for a single iteration. It shows the computational complexities to the number of pluses M for a single iteration. Suppose that $\varphi_S = \varphi_d = 4$, $N = 10$ and $L = 6$. Although the value r is unknown and unfixed in each iteration, we can suppose that it is twice as much to the rank of clutter in the current iteration. We can draw the conclusion that the computational complexities of the M-SBL-STAP algorithm are far more than that of the M-FMLM-STAP algorithm for a single iteration.

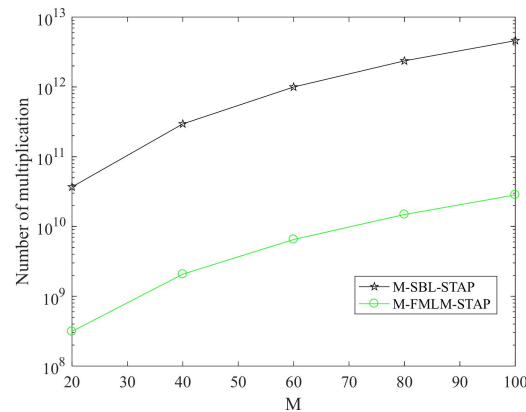


Figure 1. Computational complexity versus the number of pluses for a single iteration.

5.2. Convergence Analysis

It is obvious that the logarithm of the marginal likelihood function $\mathcal{L}(\Gamma)$ has upper bound [31]. From (62), in the t -th iteration, $\mathcal{L}(\alpha_i^*)$ is the maximal value of $\mathcal{L}(\alpha_i)$, that is to say, $\mathcal{L}(\alpha_i^*) \geq \mathcal{L}(\alpha_i^{(t)})$. Therefore, the condition $\mathcal{L}(\Gamma^{(t+1)}) \geq \mathcal{L}(\Gamma^{(t)})$, $\forall t$ is promised in each iteration referring to (65). Thus, the convergence performance of our proposed algorithm is guaranteed.

6. Performance Assessment

In this section, we firstly verify the performance of the loading sample matrix inversion (LSMI) algorithm, the multiple orthogonal matching pursuit (M-OMP-STAP) algorithm [32], the M-SBL-STAP algorithm and the proposed M-FMLM-STAP algorithm with the simulated data of side-looking ULA in the ideal case and the array errors case. The first two algorithms are common approaches in STAP. We then assess the performance of the M-SBL-STAP algorithm and the proposed M-FMLM-STAP algorithm with the Mountain-Top data. We utilize the improvement factor (IF) and signal to interference plus noise ratio (SINR) loss as two measurements of performance, which are expressed as

$$\text{IF} = \frac{|\mathbf{w}^H \mathbf{s}|^2 \text{tr}(\mathbf{R})}{\mathbf{w}^H \mathbf{R} \mathbf{w} \mathbf{s}^H \mathbf{s}} \quad (81)$$

$$L_{\text{SINR}} = \frac{\sigma^2 |\mathbf{w}^H \mathbf{s}|^2}{MN \mathbf{w}^H \mathbf{R} \mathbf{w}} \quad (82)$$

where \mathbf{R} is the exact CNCM of the CUT.

6.1. Simulated Data

The parameters of a side-looking phased array radar are listed in Table 2. The 600th range gate is chosen to be CUT. According to the parameters of the radar system, the slope of the clutter ridge is 1. We set the resolution scales $\varphi_S = 5$ and $\varphi_d = 5$. Total of six training samples are utilized for two algorithms and the parameters $a = b = 0$.

Table 2. Radar system parameters.

Symbols	Parameters	Value
λ	Wavelength	0.3 m
d	Distance between elements	0.15 m
ϑ	Platform velocity	150 m/s
H	Platform height	9000 m
M	Number of pulses	8
N	Number of channels	8
f_{PRF}	Pulse repetition frequency	2000 Hz
f_s	Range sampling frequency	2.5 MHz
θ	Perspective angle	90°
CNR	Clutter to noise ratio	30 dB

In Figure 2, there are five clutter power spectrum figures in the framework of the side-looking ULA in the ideal case. Figure 2a–e show the clutter power spectrums estimated by the exact CNCM, the LSMI algorithm, the M-OMP-STAP algorithm, the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm, respectively. We note that the clutter spectrums obtained by the M-SBL-STAP algorithm and the M-FMLM-STAP algorithms are closer to the exact spectrum both on the location and power than the other algorithms in the ideal case. As shown in Figure 3, the IF factor curve achieved by the M-SBL-STAP algorithm is a litter better than that achieved by the M-FMLM-STAP algorithm because the latter algorithm is sensitive to noise fluctuation. However, the latter algorithm needs much fewer computational complexities, and the loss on performance can be offset by the decrease on the computational complexities. Using a computer equipped with two Inter(R) Xeon(R) Gold 6140 CPU @2.30 GHz 2.29 GHz processors, the former spends more than 40 s and the latter spends less than 4 s at average. The larger the value of K , the bigger gap of computational complexities between the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm.

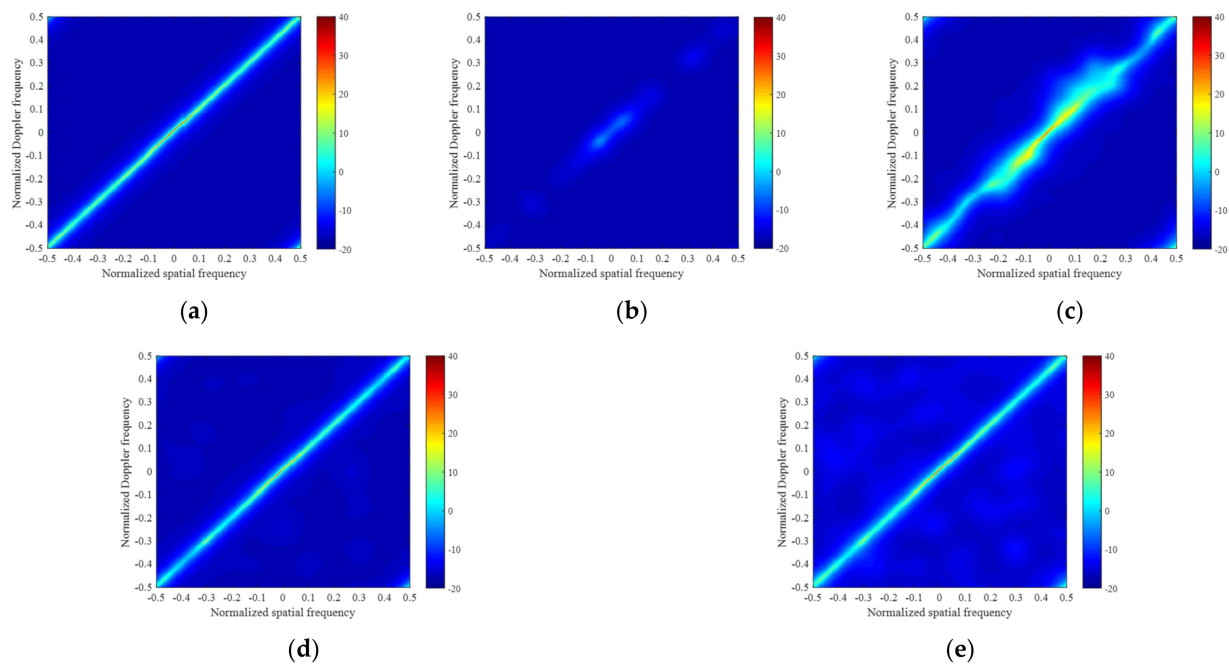


Figure 2. Angle-Doppler clutter power spectrums in the ideal case: (a) exact spectrum calculated by the ideal CNCM; (b) sparse spectrum estimated by LSMI; (c) sparse spectrum estimated by M-OMP-STAP; (d) sparse spectrum estimated by M-SBL-STAP; and (e) sparse spectrum estimated by M-FMLM-STAP.

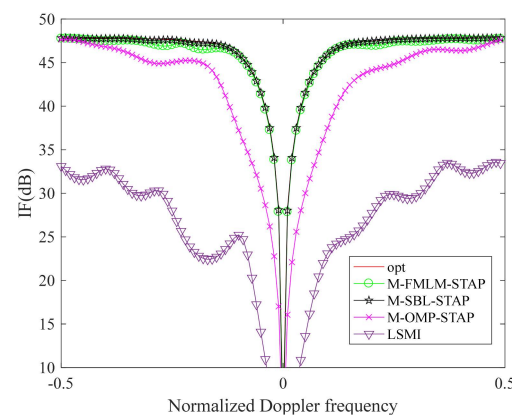


Figure 3. IF curves in the ideal case.

In Figure 4, we apply four approaches in the non-ideal case with amplitude Gaussian error (standard deviation 0.03) and phase random error (standard deviation 2°). The error is the same at all direction. Figure 4a–e show the clutter power spectrums estimated by the exact CNCM, the LSMI algorithm, the M-OMP-STAP algorithm, the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm, respectively. We note that the clutter spectrums obtained by the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm are much closer to the exact spectrum, both in terms of the location and power in the non-ideal case. As shown in Figure 5, the notch of the IF factor curve achieved by the M-SBL-STAP algorithm is close to that achieved by the M-FMLM-STAP algorithm, which means the performance of two algorithms is about the same in the non-ideal case. However, the running time of the latter is much less than that of the former.

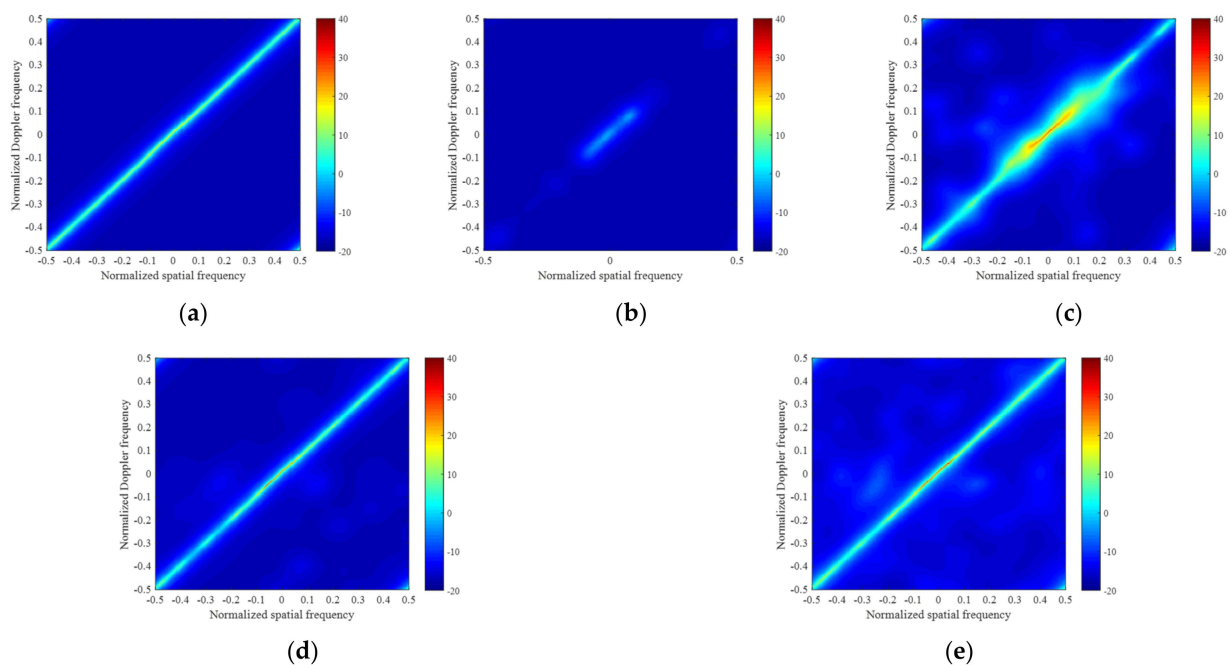


Figure 4. Angle-Doppler clutter power spectrums in the non-ideal case with array errors: (a) exact spectrum calculated by the exact CNCM; (b) sparse spectrum estimated by LSMI; (c) sparse spectrum estimated by M-OMP-STAP; (d) sparse spectrum estimated by M-SBL-STAP; and (e) sparse spectrum estimated by M-FMLM-STAP.

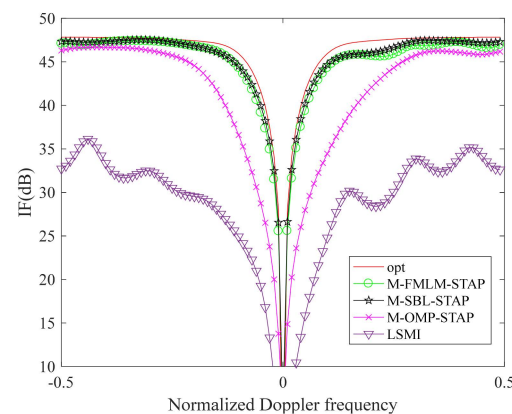


Figure 5. IF curves in the non-ideal case.

In Figure 6, the average SINR loss, defined as the mean of the SINR loss values in the entire normalized Doppler frequency range to the number of training samples of the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm, is presented. The curve of the average SINR loss achieved by the M-SBL-STAP algorithm is a little better than that achieved by the M-FMLM-STAP algorithm within 0.5 dB, which can be ignored in practice. However, the gap on computational complexities of two algorithms will be larger as the DOF of the radar system increases. All presented results are averaged over 100 independent Monte Carlo runs.

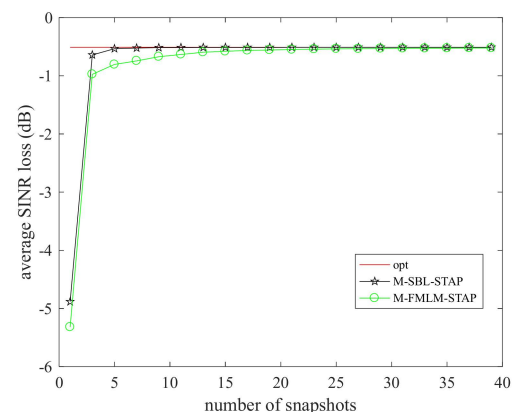


Figure 6. Average SINR loss to the number of training samples.

6.2. Measured Data

We apply the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm to the public available Mountain-Top set, i.e., t38pre01v1 CPI6 [33]. In this data file, the number of array elements and coherent pulses are 14 and 16, respectively. There are 403 range cells, and the clutter locates around 245° relative to the true North. The target is located in 147th range cell, and the azimuth angle is 275° to the true North. The normalized target Doppler frequency is 0.25. The estimated clutter Capon spectrum utilizing all 403 range cells is provided in Figure 7.

Figure 8 depicts the STAP output power of the M-SBL-STAP algorithm and the M-FMLM-STAP algorithm in the range cells from 130 to 165, and 10 range cells out of 20 range cells located next to the CUT are selected as training samples. The target locates at 147th range cell and can be detected by two algorithms. Obviously, the detection performance of the proposed M-FMLM-STAP algorithm is close to that of M-SBL-STAP algorithm. However, the operation time of the former algorithm is much less than that of the latter algorithm. Therefore, the proposed algorithm is applicable in practice.

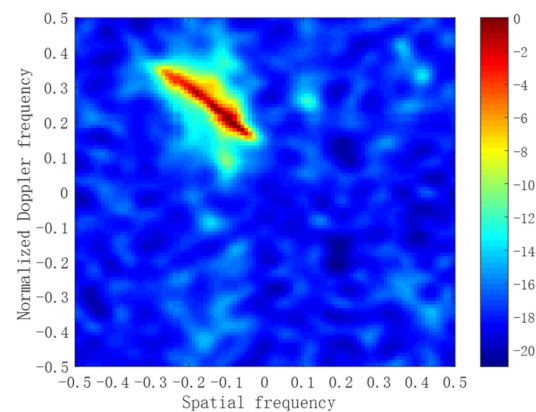


Figure 7. Estimated clutter Capon spectrum with all samples.

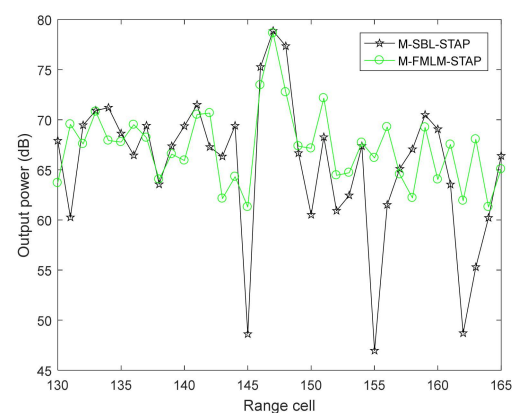


Figure 8. STAP output power to the range cell of two algorithms.

7. Conclusions

In this paper, we have extended the real-valued multitask compressive sensing technique to suppress complex-valued heterogeneous clutter for airborne radar. Unlike the conventional M-SBL-STAP algorithm, the proposed algorithm avoids the inversion of a $K \times K$ matrix at each iteration to guarantee real-time operations. We integrate the noise σ^2 out instead of estimating σ^2 to overcome the problem that we have no access to obtain the accurate value of σ^2 . The complex-valued multitask clutter data are translated into group real-valued sparse signals in the article to suppress clutter because the modified hierarchical model is not suitable to complex-valued signals. At the end, simulation results demonstrate the high computational efficiency and the great performance of the proposed algorithm.

Author Contributions: Conceptualization, C.L. and T.W.; methodology, C.L. and T.W.; software, C.L.; validation, C.L., S.Z. and B.R.; writing—original draft preparation, C.L.; writing—review and editing, S.Z.; visualization, C.L.; supervision, T.W.; project administration, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China, grant number 2021YFA1000400.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A.

For notational convenience, we ignore the symbol (t) in the t -th iteration and use \sim to represent the $(t+1)$ -th iteration.

Appendix A.1. Adding a New Basis Group Function ($\tilde{\alpha}_i > 0$ & $\alpha_i = 0$)

When $\alpha_i = 0$, $\mathbf{C} = \mathbf{C}_{-i}$.

$$\Delta\mathcal{L}(i) = -\frac{1}{2} \sum_{l=1}^L \left\{ (2NM + 2a) \log \left(1 - \frac{\mathbf{q}_{l,i}^T (\tilde{\alpha}_i^{-1} \mathbf{I}_2 + \mathbf{S}_i)^{-1} \mathbf{q}_{l,i}}{\hat{g}_l} \right) + \log |\tilde{\alpha}_i \mathbf{I}_2| + \log |\tilde{\alpha}_i^{-1} \mathbf{I}_2 + \mathbf{S}_i| \right\} \quad (\text{A1})$$

$$\tilde{\Sigma} = \begin{bmatrix} \Sigma + \Sigma \Psi^T \Phi_i \mathbf{Z} \Phi_i^T \Psi \Sigma & -\Sigma \Psi^T \Phi_i \mathbf{Z} \\ -(\Sigma \Psi^T \Phi_i \mathbf{Z})^T & \mathbf{Z} \end{bmatrix} \quad (\text{A2})$$

$$\tilde{\mu}_l = \begin{bmatrix} \mu_l - \Sigma \Psi^T \Phi_i \mathbf{Z} \mathbf{q}_{l,i} \\ \mathbf{Z} \mathbf{q}_{l,i} \end{bmatrix} \quad (\text{A3})$$

$$\tilde{\mathbf{S}}_i = \mathbf{S}_i - \mathbf{S}_i \mathbf{Z} \mathbf{S}_i = \mathbf{S}_i - \Phi_i^T \mathbf{E}_i \mathbf{Z} \mathbf{E}_i^T \Phi_i \quad (\text{A4})$$

$$\tilde{\mathbf{q}}_{l,i} = \mathbf{q}_{l,i} - \mathbf{S}_i \mathbf{Z} \mathbf{q}_{l,i} = \mathbf{q}_{l,i} - \Phi_i^T \mathbf{E}_i \mathbf{Z} \mathbf{q}_{l,i} \quad (\text{A5})$$

$$\tilde{g}_l = g_l - \mathbf{q}_{l,i}^T \mathbf{Z} \mathbf{q}_{l,i} = g_l - \mathbf{y}_l^T \mathbf{E}_i \mathbf{Z} \mathbf{E}_i^T \mathbf{y}_l \quad (\text{A6})$$

where $\mathbf{Z} = (\tilde{\alpha}_i^{-1} \mathbf{I}_2 + \mathbf{S}_i)^{-1}$ and $\mathbf{E}_i = \Phi_i - \Psi \Sigma \Psi^T \Phi_i$.

Appendix A.2. Re-Estimating a Basis Group Function ($\tilde{\alpha}_i > 0$ & $\alpha_i > 0$)

i represents the index of the i -th group in the whole dictionary matrix, and g represents the position index of the above-mentioned i -th group in the used basis matrix Ψ . $\tilde{\alpha}_i$ is the hyper-parameter re-estimated, $\Delta\alpha_i = \tilde{\alpha}_i - \alpha_i$, $\Sigma_g = \Sigma(:, 2g-1:2g)$, and $\Sigma_{gg} = \Sigma(2g-1:2g, 2g-1:2g)$.

$$\Delta\mathcal{L}(i) = -\frac{1}{2} \sum_{l=1}^L \left\{ (2NM + 2a) \log \left(1 - \frac{\hat{\mathbf{q}}_{l,i}^T (\tilde{\alpha}_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i)^{-1} \hat{\mathbf{q}}_{l,i}}{\hat{g}_{l,i}} \right) + \log |\tilde{\alpha}_i \mathbf{I}_2| + \log |\tilde{\alpha}_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i| - (2NM + 2a) \log \left(1 - \frac{\hat{\mathbf{q}}_{l,i}^T (\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i)^{-1} \hat{\mathbf{q}}_{l,i}}{\hat{g}_{l,i}} \right) - \log |\alpha_i \mathbf{I}_2| - \log |\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i| \right\} \quad (\text{A7})$$

$$\tilde{\Sigma} = \Sigma - \Sigma_g \mathbf{G} \Sigma_g^T \quad (\text{A8})$$

$$\tilde{\mu} = \mu - \Sigma_g \mathbf{G} \mu_g \quad (\text{A9})$$

$$\tilde{\mathbf{S}}_i = \mathbf{S}_i + \Phi_i^T \Psi \Sigma_g \mathbf{G} \Sigma_g^T \Psi^T \Phi_i \quad (\text{A10})$$

$$\tilde{\mathbf{q}}_{l,i} = \mathbf{q}_{l,i} + \Phi_i^T \Psi \Sigma_g \mathbf{G} \Sigma_g^T \Psi^T \mathbf{y}_l \quad (\text{A11})$$

$$\tilde{g}_l = g_l + \mathbf{y}_l^T \Psi \Sigma_g \mathbf{G} \Sigma_g^T \Psi^T \mathbf{y}_l \quad (\text{A12})$$

where $\mathbf{G} = \left(-\frac{\alpha_i \tilde{\alpha}_i}{\Delta\alpha_i} \mathbf{I}_2 + \Sigma_{gg} \right)^{-1}$ and $\mu_g = \Sigma_g^T \Psi^T \mathbf{y}$ denotes the g -th 2×1 sub-vector of μ .

Appendix A.3. Deleting a Basis Group Function ($\tilde{\alpha}_i = 0$ & $\alpha_i > 0$)

i represents the index of the i -th group in the whole dictionary matrix, and g represents the position index of the above-mentioned i -th group, which is removed in the used basis matrix ψ .

$$\Delta\mathcal{L}(i) = \frac{1}{2} \sum_{l=1}^L \left[(2NM + 2a) \log \left(1 - \frac{\tilde{\mathbf{q}}_{l,i}^T (\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i)^{-1} \tilde{\mathbf{q}}_{l,i}}{\hat{g}_{l,i}} \right) + \log |\alpha_i \mathbf{I}_2| + \log |\alpha_i^{-1} \mathbf{I}_2 + \hat{\mathbf{S}}_i| \right] \quad (\text{A13})$$

$$\tilde{\Sigma} = \Sigma - \Sigma_g \Sigma_{gg}^{-1} \Sigma_g^T \quad (\text{A14})$$

$$\tilde{\mu} = \mu - \Sigma_g \Sigma_{gg}^{-1} \mu_g \quad (\text{A15})$$

$$\tilde{\mathbf{S}}_i = \mathbf{S}_i + \Phi_i^T \psi \Sigma_g \Sigma_{gg}^{-1} \Sigma_g^T \psi^T \Phi_i \quad (\text{A16})$$

$$\tilde{\mathbf{q}}_{l,i} = \mathbf{q}_{l,i} + \Phi_i^T \psi \Sigma_g \Sigma_{gg}^{-1} \Sigma_g^T \psi^T \mathbf{y}_l \quad (\text{A17})$$

$$\tilde{g}_l = g_l + \mathbf{y}_l^T \psi \Sigma_g \Sigma_{gg}^{-1} \Sigma_g^T \psi^T \mathbf{y}_l \quad (\text{A18})$$

References

- Ward, J. *Space-Time Adaptive Processing for Airborne Radar*; Technical Report; MIT Lincoln Laboratory: Lexington, KY, USA, 1998.
- Reed, I.S.; Mallet, J.D.; Brennan, L.E. Rapid convergence rate in adaptive arrays. *IEEE Trans. Aerosp. Electron. Syst.* **1974**, *AES-10*, 853–863. [\[CrossRef\]](#)
- Dipietro, R.C. Extended factored space-time processing for airborne radar systems. In Proceedings of the 26th Asilomar Conference on Signals, Systems and Computing, Pacific Grove, CA, USA, 26–28 October 1992; pp. 425–430.
- Haimovich, A. The eigencanceler: Adaptive radar by eigenanalysis methods. *IEEE Trans. Aerosp. Electron. Syst.* **1996**, *32*, 532–542. [\[CrossRef\]](#)
- Melvin, W.L. A STAP overview. *IEEE Aerosp. Electron. Syst. Mag.* **2004**, *19*, 19–35. [\[CrossRef\]](#)
- Sarkar, T.K.; Park, S.; Koh, J. A deterministic least squares approach to space time adaptive processing. *Digit. Signal Process.* **1996**, *6*, 185–194. [\[CrossRef\]](#)
- Capraro, C.T.; Capraro, G.T.; Bradaric, I.; Weiner, D.D.; Wicks, M.C.; Baldygo, W.J. Implementing digital terrain data in knowledge-aided space-time adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1080–1097. [\[CrossRef\]](#)
- Capraro, C.T.; Capraro, G.T.; Weiner, D.D.; Wicks, M.C.; Baldygo, W.J. Improved STAP performance using knowledge-aided secondary data selection. In Proceedings of the 2004 IEEE Radar Conference, Philadelphia, PA, USA, 29 April 2004.
- Melvin, W.; Wicks, M.; Antonik, P.; Salama, Y.; Li, P.; Schuman, H. Knowledge-based space-time adaptive processing for airborne early warning radar. *IEEE Aerosp. Electron. Syst. Mag.* **1998**, *13*, 37–42. [\[CrossRef\]](#)
- Bergin, J.S.; Teixeira, C.M.; Techau, P.M. Improved clutter mitigation performance using knowledge-aided space-time adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 997–1009. [\[CrossRef\]](#)
- Donoho, D.L.; Elad, M.; Temlyakov, V.N. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory* **2006**, *52*, 6–18. [\[CrossRef\]](#)
- Sun, K.; Zhang, H.; Li, G.; Meng, H.D.; Wang, X.Q. A novel STAP algorithm using sparse recovery technique. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Cape Town, South Africa, 12–17 July 2009; Volume 1, pp. 3761–3764.
- Yang, Z.C.; Li, X.; Wang, H.Q.; Jiang, W.D. On clutter sparsity analysis in space-time adaptive processing airborne radar. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1214–1218. [\[CrossRef\]](#)
- Yang, Z.C.; de Lamare, R.C.; Li, X. L1-regularized STAP algorithm with a generalized sidelobe canceler architecture for airborne radar. *IEEE Trans. Signal Process.* **2012**, *60*, 674–686. [\[CrossRef\]](#)
- Sen, S. Low-rank matrix decomposition and spatio-temporal sparse recovery for STAP radar. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 1510–1523. [\[CrossRef\]](#)
- Tipping, M.E. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn.* **2001**, *1*, 211–244.
- Wipf, D.P.; Rao, B.D. Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.* **2004**, *52*, 2153–2164. [\[CrossRef\]](#)
- Wipf, D.P.; Rao, B.D. An empirical Bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Trans. Signal Process.* **2007**, *55*, 3704–3716. [\[CrossRef\]](#)
- Tipping, M.E.; Faul, A.C. Fast marginal likelihood maximization for sparse Bayesian models. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003; pp. 3–6.
- Ji, S.H.; Xue, Y.; Carin, L. Bayesian compressive sensing. *IEEE Trans. Signal Process.* **2008**, *56*, 2346–2356. [\[CrossRef\]](#)
- Ji, S.H.; Dunson, D.; Carin, L. Multi-task compressive sensing. *IEEE Trans. Signal Process.* **2009**, *57*, 92–106. [\[CrossRef\]](#)

22. Wu, Q.S.; Zhang, Y.M.; Amin, M.G.; Himed, B. Complex multitask Bayesian compressive sensing. In Proceedings of the 2014 IEEE International Conference on Acoustic, Speech and Signal Processing, Florence, Italy, 4–9 May 2014.
23. Serra, J.G.; Testa, M.; Katsaggelos, A.K. Bayesian K-SVD using fast variational inference. *IEEE Trans. Image Process.* **2017**, *26*, 3344–3359. [[CrossRef](#)]
24. Ma, Z.Q.; Dai, W.; Liu, Y.M.; Wang, X.Q. Group sparse Bayesian learning via exact and fast marginal likelihood maximization. *IEEE Trans. Signal Process.* **2017**, *65*, 2741–2753. [[CrossRef](#)]
25. Duan, K.Q.; Wang, Z.T.; Xie, W.C.; Chen, H.; Wang, Y.L. Sparsity-based STAP algorithm with multiple measurement vectors via sparse Bayesian learning strategy for airborne radar. *IET Signal Process.* **2017**, *11*, 544–553. [[CrossRef](#)]
26. Wang, Z.T.; Xie, W.C.; Duan, K.Q. Clutter suppression algorithm base on fast converging sparse Bayesian learning for airborne radar. *Signal Process.* **2017**, *130*, 159–168. [[CrossRef](#)]
27. Yuan, H.D.; Xu, H.; Duan, K.Q.; Xie, W.C.; Liu, W.J.; Wang, Y.L. Sparse Bayesian learning-based space-time adaptive processing with off-grid self-calibration for airborne radar. *IEEE Access* **2018**, *6*, 47296–47307. [[CrossRef](#)]
28. Yang, X.P.; Sun, Y.Z.; Yang, J.; Long, T.; Sarkar, T.K. Discrete Interference suppression method based on robust sparse Bayesian learning for STAP. *IEEE Access* **2019**, *10*, 26740–26751. [[CrossRef](#)]
29. Bai, Z.L.; Shi, L.M.; Sun, J.W.; Christensen, M.G. Complex sparse signal recovery with adaptive Laplace priors. *arXiv* **2006**, arXiv:2006.16720v1.
30. Babacan, S.D.; Molina, R.; Katsaggelos, A.K. Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **2010**, *19*, 53–63. [[CrossRef](#)]
31. Wipf, D.; Nagarajan, S. A new view of automatic relevance determination. *Adv. Neural Inf. Process. Syst.* **2008**, *20*, 1–9.
32. Tropp, J.A. Algorithms for simultaneous sparse approximation, part I: Greedy pursuit. *Signal Process.* **2006**, *86*, 572–588. [[CrossRef](#)]
33. Titi, G.W.; Marshall, D.F. The ARPA/NAVY mountaintop program: Adaptive signal processing for airborne early warning radar. In Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, USA, 9 May 1996.