---

**Article:**
Fa, R., de Lamare, R.C. and Wang, L. (2010) Reduced-Rank STAP Schemes for Airborne Radar Based on Switched Joint Interpolation, Decimation and Filtering Algorithm. IEEE Transactions on Signal Processing. 5447728. pp. 4182-4194. ISSN 1053-587X

https://doi.org/10.1109/TSP.2010.2048212

---

**Published paper**

Fa, R, de Lamare, R.C, Wang, L(2010)
Reduced-Rank STAP Schemes for Airborne Radar Based on Switched Joint Interpolation, Decimation and Filtering Algorithm

IEEE TRANSACTIONS ON SIGNAL PROCESSING
58 (8) 4182-4194
http://dx.doi.org/10.1109/TSP.2010.2048212

# Reduced-Rank STAP Schemes for Airborne Radar Based on Switched Joint Interpolation, Decimation and Filtering Algorithm

Rui Fa, Rodrigo C. de Lamare and Lei Wang

*Abstract*— **In this paper, we propose a reduced-rank space-time adaptive processing (STAP) technique for airborne phased array radar applications. The proposed STAP method performs dimensionality reduction by using a reduced-rank switched joint interpolation, decimation and filtering algorithm (RR-SJIDF). In this scheme, a multiple-processing-branch (MPB) framework, which contains a set of jointly optimized interpolation, decimation and filtering units, is proposed to adaptively process the observations and suppress jammers and clutter. The output is switched to the branch with the best performance according to the minimum variance criterion. In order to design the decimation unit, we present an optimal decimation scheme and a low-complexity decimation scheme. We also develop two adaptive implementations for the proposed scheme, one based on a recursive least squares (RLS) algorithm and the other on a constrained conjugate gradient (CCG) algorithm. The proposed adaptive algorithms are tested with simulated radar data. The simulation results show that the proposed RR-SJIDF STAP schemes with both the RLS and the CCG algorithms converge at a very fast speed and provide a considerable SINR improvement over the state-of-the-art reduced-rank schemes.**

*Index Terms*— **Space-time adaptive processing (STAP), reduced-rank techniques, airborne phased array radar.**

## I. Introduction

SPACE-time adaptive processing (STAP) techniques have been motivated as a key enabling technology for advanced airborne radar applications following the landmark publication by Brennan and Reed [1]. A great deal of attention has been given to STAP algorithms and much of the work has been done in the past three decades [2]–[15]. It is fully understood that STAP techniques can improve slow-moving target detection through better mainlobe clutter suppression, provide better detection in combined clutter and jamming environments, and offer a significant increase in output signal-to-interference-plus-noise-ratio (SINR). However, due to its large computational complexity cost by the matrix inversion operation, the optimum STAP processor is prohibitive for practical implementation. Furthermore, an even more challenging issue is raised by full-rank STAP techniques when the number of elements $M$ in the filter is large. It is well-known that $K \geq 2M$ independent and identically distributed (i.i.d) training samples are required for the filter to achieve the steady performance [16]. Thus, in dynamic scenarios the full-rank STAP with large $M$ usually fail or provide poor performance in tracking target signals contaminated by interference and noise.

Reduced-rank adaptive signal processing has been considered as a key technique for dealing with large systems in the last decade. The basic idea of the reduced-rank algorithms is to reduce the number of adaptive coefficients by projecting the received vectors onto a lower dimensional subspace which consists of a set of basis vectors. The adaptation of the low-order filter within the lower dimensional subspace results in significant computational savings, faster convergence speed and better tracking performance. The first statistical reduced-rank method was based on a principal-components (PC) decomposition of the target-free covariance matrix [4]. Another class of eigen-decomposition methods was based on the cross-spectral metric (CSM) [8]. Both the PC and the CSM algorithms require a high computational cost due to the eigen-decomposition. A family of the Krylov subspace methods has been investigated thoroughly in the recent years. This class of reduced-rank algorithms, including the multistage Wiener filter (MSWF) [12], [18] and the auxiliary-vector filters (AVF) [19]–[21], projects the observation data onto a lower-dimensional Krylov subspace. These methods are very complex to implement in practice and suffer from numerical problems despite their improved convergence and tracking performance. The joint domain localized (JDL) approach, which is a beamspace reduced-dimension algorithm, was proposed by Wang and Cai [22] and investigated in both homogeneous and nonhomogeneous environments in [23], [24], respectively. Recently, reduced-rank adaptive processing algorithms based on joint iterative optimization of adaptive filters [25], [26] and based on an adaptive diversity-combined decimation and interpolation scheme [27], [28] were proposed, respectively. In our prior work [26], a joint iterative optimization of adaptive filters STAP scheme using the linearly constrained minimum variance (LCMV) was considered and applied to airborne radar applications, resulting in a significant improvement both in convergence speed and SINR performance as compared with the existing reduced-rank STAP algorithms.

The goal of this paper is to devise cost-effective STAP algorithms that have substantially faster convergence performance than existing methods. This enables the radar system with a significantly better probability of detection ($P_D$) with limited training. We develop a reduced-rank STAP design based on a switched joint interpolation, decimation and filtering (RR-SJIDF) algorithm for airborne radar systems. In this scheme, the number of elements for adaptive processing is substantially

This work is funded by the Ministry of Defence (MoD), UK. Project MoD, Contract No. RT/COM/S/021.The authors are with the Communications Research Group, Department of Electronics, University of York, YO10 5DD, United Kingdom. Email: {rf533, rcdl500, lw517}@ohm.york.ac.uk

reduced, resulting in considerable computational savings and very fast convergence performance for radar applications. The proposed approach obtains the subspace of interest via a multiple processing branch (MPB) framework which consists of a set of simple interpolation, decimation and filtering operations. Unlike the previous work in [27], multiple interpolators and reduced-rank filters are employed in the MPB framework and are designed with the LCMV criterion. For each branch, the interpolator and the reduced-rank filter can be jointly optimized by minimizing a cost function subject to linear constraints. We describe an optimal decimation scheme and a low-complexity decimation scheme for the proposed structure. We also derive two adaptive implementations using the recursive least squares (RLS) and the constrained conjugate gradient (CCG) algorithms for the proposed scheme and evaluate their computational complexity. The numerical results show that the proposed RR-SJIDF STAP schemes with both the RLS and the CCG algorithms converge at a very fast speed and provide a considerable SINR improvement with significantly low complexity compared with the existing reduced-dimension and reduced-rank algorithms, namely, the JDL, the MSWF and the AVF algorithms.

The main contributions of our paper are listed as follows:

i) A reduced-rank STAP scheme based on SJIDF algorithm for airborne radar platform is proposed.
ii) In the proposed scheme, a MPB framework is introduced. For each branch, the interpolator and reduced-rank filters are jointly optimized by minimizing the modified minimum variance (MV) cost function with a set of constraints.
iii) Two efficient adaptive implementations using the RLS and the CCG algorithms are developed for the proposed STAP scheme and a detailed study of their computational complexity requirements is provided.
iv) Algorithms for automatically adjusting the rank of the proposed SJIDF scheme are developed.
v) A study and comparative analysis of reduced-rank STAP techniques for radar systems is carried out.

This paper is organized as follows. Section II states the signal model, the optimum full-rank STAP algorithm and the fundamentals of reduced-rank signal processing. Section III presents the proposed reduced-rank STAP scheme, describes the proposed joint iterative optimization of the interpolation, decimation and filtering tasks, and details the proposed decimation schemes. In Section IV, we develop two adaptive implementations using the RLS and the CCG algorithms and algorithms for automatically adjusting the rank of the proposed scheme. In Section V, we discuss the convergence properties of the optimization of the proposed scheme. The performance assessment of the proposed reduced-rank STAP scheme is provided in Section VI using simulated radar data. Finally, conclusions are given in Section VII.

## II. Signal Model, Radar Signal Processing and Problem Statement

The system under consideration is a pulsed Doppler radar residing on an airborne platform. The radar antenna is a uniformly spaced linear array antenna consisting of $N$ elements.

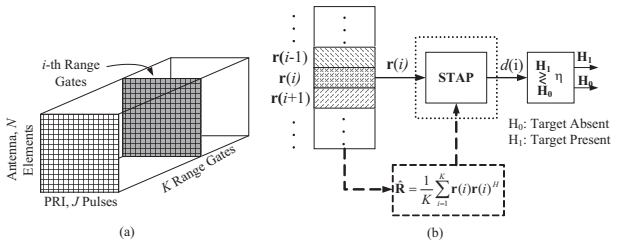


Fig. 1. (a) The Radar CPI datacube. (b) The STAP schematic.

Radar returns are collected in a coherent processing interval (CPI), which is referred to as the 3-D radar datacube shown in Fig. 1(a), where $K$ denotes the number of samples collected to cover the range interval. The data is then processed at one range of interest, which corresponds to a slice of the CPI datacube. This slice is a $J \times N$ matrix which consists of $N \times 1$ spatial snapshots for $J$ pulses at the range of interest. It is convenient to stack the matrix column-wise to form the $M \times 1, M = JN$ vector $\mathbf{r}(i)$, termed the $i$-th range gate space-time snapshot, $1 \leq i \leq K$ [1].

### A. Signal Model

The objective of a radar is to ascertain whether targets are present in the data. Thus, given a space-time snapshot, radar detection is a binary hypothesis problem, where hypothesis $\mathbf{H}_0$ corresponds to target absence and hypothesis $\mathbf{H}_1$ corresponds to target presence. The radar space-time snapshot is then expressed for each of the two hypotheses in the following form,

$$\begin{aligned} \mathbf{H}_0 &: \mathbf{r}(i) = \mathbf{v}(i), \\ \mathbf{H}_1 &: \mathbf{r}(i) = a\mathbf{s} + \mathbf{v}(i), \end{aligned} \tag{1}$$

where $a$ is a zero-mean complex Gaussian random variable with variance $\sigma_s^2$, $\mathbf{v}(i)$ denotes the input interference-plus-noise vector which consists of clutter $\mathbf{r}_c(i)$, jamming $\mathbf{r}_j(i)$ and the white noise $\mathbf{r}_n(i)$. These three components are assumed to be mutually uncorrelated. Thus, the $M \times M$ covariance matrix $\mathbf{R}$ of the undesired clutter-plus-jammer-plus-noise component can be modelled as

$$\mathbf{R} = \mathbb{E}\{\mathbf{v}(i)\mathbf{v}^H(i)\} = \mathbf{R}_c + \mathbf{R}_j + \mathbf{R}_n, \tag{2}$$

where $H$ represents Hermitian transpose and $\mathbb{E}$ denotes expectation. According to [6], the noise covariance noise matrix $\mathbf{R}_n = \mathbb{E}\{\mathbf{r}_n(i)\mathbf{r}_n^H(i)\}$ can be written as a scaled identity matrix $\sigma_n^2\mathbf{I}_M$, where $\sigma_n^2$ is the noise power. The clutter signal can be modeled as the superposition of a large number of independent clutter patches with evenly distributed in azimuth about the receiver. Thus, the clutter covariance matrix can be expressed as

$$\begin{aligned} \mathbf{R}_c &= \mathbb{E}\{\mathbf{r}_c(i)\mathbf{r}_c^H(i)\} \\ &= \sum_{k=1}^{N_r}\sum_{l=1}^{N_c} \xi_{kl}^c \left[\mathbf{b}(\vartheta_{kl}^c)\mathbf{b}(\vartheta_{kl}^c)^H\right] \otimes \left[\mathbf{a}(\varpi_{kl}^c)\mathbf{a}(\varpi_{kl}^c)^H\right], \end{aligned} \tag{3}$$

where $N_r$ denotes the number of range ambiguities and $N_c$ denotes the number of the clutter patches. $\xi_{kl}^c$ is the power

of reflected signal by the $kl$-th clutter patch. The notation $\otimes$ denotes Kronecker product. $\mathbf{b}(\vartheta_{kl}^c)$ and $\mathbf{a}(\varpi_{kl}^c)$, respectively, denote the spatial steering vector with the spatial frequency $\vartheta_{kl}^c$ and the temporal steering vector with the normalized Doppler frequency $\varpi_{kl}^c$ for the $kl$-th clutter patch, which can be expressed as follows

$$\mathbf{b}(\vartheta) = \begin{bmatrix} 1 \\ e^{-j2\pi\vartheta} \\ e^{-j2\pi 2\vartheta} \\ \vdots \\ e^{-j2\pi(N-1)\vartheta} \end{bmatrix}, \ \mathbf{a}(\varpi) = \begin{bmatrix} 1 \\ e^{-j2\pi\varpi} \\ e^{-j2\pi 2\varpi} \\ \vdots \\ e^{-j2\pi(K-1)\varpi} \end{bmatrix}, \quad (4)$$

where $\vartheta = \frac{d}{\lambda}\cos(\phi)\sin(\theta)$ and $\varpi = f_d/f_r$, where $\lambda$ is wavelength; $d$ is interelement spacing which is normally set to half wavelength; $\phi$ and $\theta$ are elevation and azimuth, respectively; $f_d$ and $f_r$ are Doppler frequency and pulse repetition frequency, respectively. The jamming covariance matrix $\mathbf{R}_j = \mathbb{E}\{\mathbf{r}_j(i)\mathbf{r}_j^H(i)\}$ can be written as

$$\sum_{q=1}^{N_j} \xi_q^j \left[\mathbf{b}(\vartheta_q^j)\mathbf{b}(\vartheta_q^j)^H\right] \otimes \mathbf{I}_K,$$

where $\xi_q^j$ is the power of the $q$-th jammer. $\mathbf{b}(\vartheta_q^j)$ is the spatial steering vector with the spatial frequency $\vartheta_q^j$ of the $q$-th jammer and $N_j$ is the number of jammers. The vector $\mathbf{s}$, which is the $M \times 1$ normalized space-time steering vector in the space-time look-direction, can be defined as:

$$\mathbf{s} = \sqrt{\xi_t}\mathbf{b}(\vartheta_t) \otimes \mathbf{a}(\varpi_t), \quad (5)$$

where $\mathbf{a}(\varpi_t)$ is the $K \times 1$ normalized temporal steering vector at the target Doppler frequency $\varpi_t$ and $\mathbf{b}(\vartheta_t)$ is the $N \times 1$ normalized spatial steering vector in the direction provided by the target spatial frequency $\vartheta_t$ and $\xi_t$ denotes the power of the target.

### *B. Optimum Radar Signal Processing*

To detect the presence of targets, each range bin is processed by an adaptive 2D beamformer (to achieve maximum output SINR) followed by a hypothesis test to determine the target presence or absence. Here, we assume that the secondary data $\{\mathbf{r}(i)\}_{i=1}^K$ are i.i.d training samples. The optimum full-rank STAP [1] obtained by an unconstrained optimization of the SINR is given as follows:

$$\boldsymbol{\omega}_{opt} = k\mathbf{R}^{-1}\mathbf{s}, \quad (6)$$

where $k$ is an arbitrary nonzero complex number. By solving the LCMV problem as [37]

$$\boldsymbol{\omega}_{opt} = \arg\min_{\boldsymbol{\omega}(i)} \boldsymbol{\omega}^H(i)\mathbf{R}\boldsymbol{\omega}(i) \quad \text{s. t.} \quad \mathbf{s}^H\boldsymbol{\omega}(i) = 1, \quad (7)$$

the optimal constrained weight vector for maximizing the output SINR, while maintaining a normalized response in the target spatial-Doppler look-direction was originally given in [29] by

$$\boldsymbol{\omega}_{opt} = \frac{\mathbf{R}^{-1}\mathbf{s}}{\mathbf{s}^H\mathbf{R}^{-1}\mathbf{s}}. \quad (8)$$

The solution in (8) is also known as the minimum variance distortionless response (MVDR) solution.

### *C. Reduced-Rank Signal Processing*

The basic idea of reduced-rank algorithms is to reduce the number of adaptive coefficients by projecting the received vectors onto a lower dimensional subspace as illuminated in the figure. Let $\mathbf{S}_D$ denote the $M \times D$ projection matrix with column vectors which are an $M \times 1$ basis for a $D$-dimensional subspace, where $D < M$. Thus, the received signal $\mathbf{r}(i)$ is transformed into its reduced-rank version $\bar{\mathbf{r}}(i)$ given by

$$\bar{\mathbf{r}}(i) = \mathbf{S}_D^H\mathbf{r}(i). \quad (9)$$

The reduced-rank signal is processed by an adaptive reduced-rank filter $\bar{\boldsymbol{\omega}}(i) \in \mathcal{C}^{D\times 1}$. Subsequently, the decision is made based on the filter output $y(i) = \bar{\boldsymbol{\omega}}^H(i)\bar{\mathbf{r}}(i)$. By solving the optimization problem as below

$$\bar{\boldsymbol{\omega}}_{opt} = \arg\min_{\bar{\boldsymbol{\omega}}(i)} \bar{\boldsymbol{\omega}}^H(i)\bar{\mathbf{R}}\bar{\boldsymbol{\omega}}(i), \quad \text{subject to} \quad \bar{\boldsymbol{\omega}}^H(i)\bar{\mathbf{s}} = 1, \quad (10)$$

the optimum MVDR solution for the reduced-rank weight vector $\bar{\boldsymbol{\omega}}_{opt}$ is obtained [26]

$$\bar{\boldsymbol{\omega}}_{opt} = \frac{\bar{\mathbf{R}}^{-1}\bar{\mathbf{s}}}{\bar{\mathbf{s}}^H\bar{\mathbf{R}}^{-1}\bar{\mathbf{s}}}, \quad (11)$$

where $\bar{\mathbf{R}} = \mathbf{S}_D^H\mathbf{R}\mathbf{S}_D$ denotes the reduced-rank covariance matrix and $\bar{\mathbf{s}} = \mathbf{S}_D^H\mathbf{s}$ denotes the reduced-rank steering vector.

The challenge left to us is how to efficiently design and optimize the projection matrix $\mathbf{S}_D$. The PC method which is also known as the eigencanceller method [4] suggested to form the projection matrix using the eigenvectors of the covariance matrix $\mathbf{R}$ corresponding to the eigenvalues with significant magnitude. The CSM method, a counterpart of the PC method belonging to the eigen-decomposition algorithm family, outperforms the PC method because it employs the projection matrix which contains the eigenvectors which contribute the most towards maximizing the SINR [17]. A family of closely related reduced-rank adaptive filters, such as the MSWF [18] and the AVF [19], employs a set of basis vectors as the projection matrix which spans the same subspace, known as the Krylov subspace. The Krylov subspace is generated by taking the powers of the covariance matrix of observations on a cross-correlation (or steering) vector. Despite the improved convergence and tracking performance achieved with these methods, the remaining problems are their high complexity and the existence of numerical problems for implementation. The joint domain localized (JDL) approach, which is a beamspace reduced-dimension algorithm, was proposed by Wang and Cai [22] and investigated in both homogeneous and nonhomogeneous environments in [23], [24], respectively. Recently, reduced-rank filtering algorithms based on joint iterative optimization of adaptive filters [25], [26] and based on an adaptive diversity-combined decimation and interpolation scheme [27], [28] were proposed, respectively.

## III. Proposed RR-SJIDF Scheme

In this section, we detail the proposed adaptive reduced-rank filtering scheme based on the switched joint interpolation, decimation and filtering (RR-SJIDF). The reduced-rank adaptive filtering scheme based on combined decimation

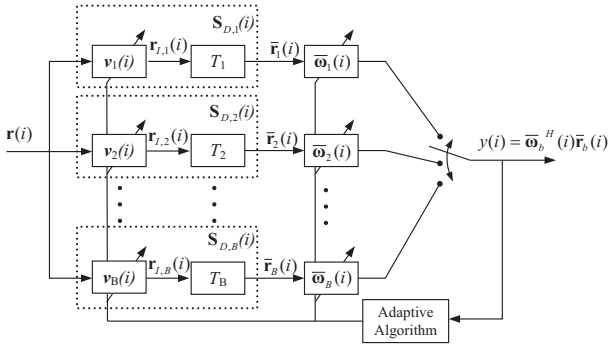


Fig. 2. Proposed Adaptive Reduced-Rank Filtering Scheme (RR-SJIDF).

and interpolation filtering was presented in [27], [28]. In this work, we develop a reduced-rank STAP algorithm based on the SJIDF scheme for airborne radar applications, whose schematic is shown in Fig. 2. The motivation for designing a projection matrix based on interpolation and decimation comes from two observations. The first is that rank reduction can be performed by constructing new samples with interpolators and eliminating (decimating) samples that are not useful in the STAP design. The second comes from the structure of the projection matrix, whose columns are a set of vectors formed by the interpolators and the decimators.

### A. Overview of the RR-SJIDF Scheme

Here, we explain how the proposed RR-SJIDF scheme works and its main building blocks. In this scheme, the number of elements for adaptive processing is substantially reduced, resulting in considerable computational savings and very fast convergence performance for the radar applications. The proposed approach obtains the subspace of interest via a multiple processing branch (MPB) framework. The $M \times 1$ received vector $\mathbf{r}(i) = [r_0(i), r_1(i), \cdots, r_{M-1}(i)]^T$ is processed by a MPB framework with $B$ branches, where each spatio-temporal processing branch contains an interpolator filter, a decimation unit and a reduced-rank filter. In the $b$-th branch $b \in [1, B]$, the received vector $\mathbf{r}(i)$ is filtered by the interpolator filter $\bar{\boldsymbol{\upsilon}}_b(i) = [\upsilon_{0,b}(i), \upsilon_{1,b}(i), \cdots, \upsilon_{I-1,b}(i)]^T$ with filter length $I$, yielding the interpolated received vector $\mathbf{r}'_b(i)$ with $M$ samples, which is expressed by

$$\mathbf{r}'_b(i) = \mathbf{V}_b(i)\mathbf{r}(i), \tag{12}$$

where the $M \times M$ Toeplitz convolution matrix $\mathbf{V}_b(i)$ is given by

$$\mathbf{V}_b(i) = \begin{bmatrix} \upsilon_{0,b}(i) & 0 & \dots & 0 \\ \vdots & \upsilon_{0,b}(i) & \dots & 0 \\ \upsilon_{I-1,b}(i) & \vdots & \dots & 0 \\ 0 & \upsilon_{I-1,b}(i) & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \upsilon_{0,b}(i) \end{bmatrix}. \tag{13}$$

In order to facilitate the description of the scheme, let us express the vector $\mathbf{r}'_b(i)$ in an alternative way which will be useful in the following through the equivalence:

$$\mathbf{r}'_b(i) = \mathbf{V}_b(i)\mathbf{r}(i) = \boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i), \tag{14}$$

where the $M \times I$ matrix $\boldsymbol{\mathcal{R}}_0(i)$ with the samples of $\mathbf{r}(i)$ has a Hankel structure [30] and is described by

$$\boldsymbol{\mathcal{R}}_0(i) = \begin{bmatrix} r_0(i) & r_1(i) & \dots & r_{I-1}(i) \\ r_1(i) & r_2(i) & \dots & r_I(i) \\ \vdots & \vdots & \dots & \vdots \\ r_{M-I}(i) & r_{M-I+1}(i) & \dots & r_{M-1}(i) \\ r_{M-I+1}(i) & r_{M-I+2}(i) & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r_{M-2}(i) & r_{M-1}(i) & 0 & 0 \\ r_{M-1}(i) & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

The dimensionality reduction is performed by a decimation unit with $D \times M$ decimation matrices $\mathbf{T}_b$ that projects $\mathbf{r}_I(i)$ onto $D \times 1$ vectors $\bar{\mathbf{r}}_b(i)$ with $b = 1, \dots, B$, where $D = M/L$ is the rank and $L$ is the decimation factor. The $D \times 1$ vector $\bar{\mathbf{r}}_b(i)$ for branch $b$ is expressed by

$$\bar{\mathbf{r}}_b(i) = \underbrace{\mathbf{T}_b\mathbf{V}_b(i)}_{\mathbf{S}_{D,b}(i)}\mathbf{r}(i) = \mathbf{T}_b\mathbf{r}'_b(i) = \mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i), \tag{16}$$

where $\mathbf{S}_{D,b}(i)$ is the equivalent projection matrix and the vector $\bar{\mathbf{r}}_b(i)$ for branch $b$ is used in the minimization of the output power for branch $b$, which is given by

$$|y_b(i)|^2 = |\bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{r}}_b(i)|^2.$$

The output at the end of the MPB framework $y(i)$ is selected according to:

$$y(i) = y_{b_s}(i) \quad \text{when} \quad b_s = \arg\min_{1 \le b \le B} |y_b(i)|^2, \tag{17}$$

where $B$ is a parameter to be set by the designer.Essential to the derivation of the joint iterative optimization that follows is to express the output of the RR-SJIDF STAP $y_b(i) = \bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{r}}_b(i)$ as a function of $\bar{\boldsymbol{\upsilon}}_b(i)$, the decimation matrix $\mathbf{T}_b$ and $\bar{\boldsymbol{\omega}}_b^H(i)$ as follows:

$$\begin{aligned} y_b(i) &= \bar{\boldsymbol{\omega}}_b^H(i)\mathbf{S}_{D,b}(i)\mathbf{r}(i) \\ &= \bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i) = \bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{r}}_{\bar{\omega},b}(i) \\ &= [\bar{\boldsymbol{\upsilon}}_b^H(i)\boldsymbol{\mathcal{R}}_0^H(i)\mathbf{T}_b^H\bar{\boldsymbol{\omega}}_b(i)]^* = [\bar{\boldsymbol{\upsilon}}_b^H(i)\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)]^*. \end{aligned} \tag{18}$$

where $\bar{\mathbf{r}}_{\bar{\omega},b}(i) = \mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i)$ denotes the reduced-rank signal with respect to $\bar{\boldsymbol{\omega}}_b(i)$ and $\bar{\mathbf{r}}_{\bar{\upsilon},b}(i) = \boldsymbol{\mathcal{R}}_0^H(i)\mathbf{T}_b^H\bar{\boldsymbol{\omega}}_b(i)$ denotes the reduced-rank signal with respect to $\bar{\boldsymbol{\upsilon}}_b(i)$, $(\cdot)^*$ denotes the conjugate operation. The expression (18) indicates that the dimensionality reduction carried out by the proposed scheme depends on finding appropriate $\bar{\boldsymbol{\upsilon}}_b(i)$, $\bar{\boldsymbol{\omega}}_b(i)$ and $\mathbf{T}_b$. In the following subsections we will derive the joint optimizations of $\bar{\boldsymbol{\upsilon}}_b(i)$ and $\bar{\boldsymbol{\omega}}_b(i)$ and design the decimation unit $\mathbf{T}_b$.

### B. Optimization of the Filters

In this part, we describe the proposed joint and iterative optimization algorithm that adjusts the parameters of the interpolator filter $\bar{\boldsymbol{v}}_b(i)$ and the reduced-rank filter $\bar{\boldsymbol{\omega}}_b(i)$ with the given decimation pattern $\mathbf{T}_b$. According to the LCMV criterion, the optimization problem is given by

$$\begin{aligned} &\min \mathbb{E}\left[\left|\bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{v}}_b(i)\right|^2\right] \\ &\text{subject to } \bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{v}}_b(i)=1,\ b=[1,\cdots,B], \end{aligned} \tag{19}$$

where $\boldsymbol{\mathcal{S}}_0$ is $M\times I$ steering matrix with a Hankel structure, which has the same form as $\boldsymbol{\mathcal{R}}_0(i)$

$$\boldsymbol{\mathcal{S}}_0=\begin{bmatrix} s_0 & s_1 & \cdots & s_{I-1} \\ s_1 & s_2 & \cdots & s_I \\ \vdots & \vdots & \cdots & \vdots \\ s_{M-I} & s_{M-I+1} & \cdots & s_{M-1} \\ s_{M-I+1} & s_{M-I+2} & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_{M-2} & s_{M-1} & 0 & 0 \\ s_{M-1} & 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

The constrained cost function in (19) can be transformed into unconstrained one by introducing a Lagrangian multiplier, which is given as

$$\begin{aligned} \mathcal{L}(\bar{\boldsymbol{\omega}}_b(i),\bar{\boldsymbol{v}}_b(i)) &= \mathbb{E}\left[\left|\bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{v}}_b(i)\right|^2\right] \\ &\quad + 2\Re\left\{\lambda\left[\bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{v}}_b(i)-1\right]\right\}, \end{aligned} \tag{21}$$

where $\lambda$ is the Lagrangian multiplier. By fixing $\bar{\boldsymbol{\omega}}(i)$ and $\bar{\boldsymbol{v}}(i)$, respectively, (21) can be rewritten into two equations as

$$\mathcal{L}(\bar{\boldsymbol{v}}_b(i)) = \mathbb{E}\left[\left|\bar{\boldsymbol{v}}_b^H(i)\bar{\mathbf{r}}_{\bar{v},b}(i)\right|^2\right] + 2\Re\left\{\lambda_{\bar{v},b}\left[\bar{\boldsymbol{v}}_b^H(i)\bar{\mathbf{s}}_{\bar{v},b}(i)-1\right]\right\}$$
$$\mathcal{L}(\bar{\boldsymbol{\omega}}_b(i)) = \mathbb{E}\left[\left|\bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{r}}_{\bar{\omega},b}(i)\right|^2\right] + 2\Re\left\{\lambda_{\bar{\omega},b}\left[\bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)-1\right]\right\}$$

where $\bar{\mathbf{s}}_{\bar{v},b}(i)=\mathbf{T}_b^H(i)\boldsymbol{\mathcal{S}}_0^H\bar{\boldsymbol{\omega}}_b(i)$ and $\bar{\mathbf{s}}_{\bar{\omega},b}(i)=\mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{v}}_b(i)$ denote the reduced-rank steering vectors with respect to $\bar{\boldsymbol{v}}(i)$ and $\bar{\boldsymbol{\omega}}(i)$, respectively. $\lambda_{\bar{v},b}$ and $\lambda_{\bar{\omega},b}$ are the Lagrangian multipliers for $\bar{\boldsymbol{v}}(i)$ and $\bar{\boldsymbol{\omega}}(i)$, respectively. By minimizing $\mathcal{L}(\bar{\boldsymbol{v}}_b(i))$ and solving for $\lambda_{\bar{v},b}$, we get

$$\bar{\boldsymbol{v}}_b(i)=\frac{\bar{\mathbf{R}}_{\bar{v},b}^{-1}\bar{\mathbf{s}}_{\bar{v},b}(i)}{\bar{\mathbf{s}}_{\bar{v},b}^H(i)\bar{\mathbf{R}}_{\bar{v},b}^{-1}\bar{\mathbf{s}}_{\bar{v},b}(i)}, \tag{22}$$

where $\bar{\mathbf{R}}_{\bar{v},b}=\mathbb{E}\left[\bar{\mathbf{r}}_{\bar{v},b}(i)\bar{\mathbf{r}}_{\bar{v},b}^H(i)\right]$. By minimizing $\mathcal{L}(\bar{\boldsymbol{\omega}}_b(i))$ and solving for $\lambda_{\bar{\omega},b}$, we get

$$\bar{\boldsymbol{\omega}}_b(i)=\frac{\bar{\mathbf{R}}_{\bar{\omega},b}^{-1}\bar{\mathbf{s}}_{\bar{\omega},b}(i)}{\bar{\mathbf{s}}_{\bar{\omega},b}^H(i)\bar{\mathbf{R}}_{\bar{\omega},b}^{-1}\bar{\mathbf{s}}_{\bar{\omega},b}(i)}, \tag{23}$$

where $\bar{\mathbf{R}}_{\bar{\omega},b}=\mathbb{E}\left[\bar{\mathbf{r}}_{\bar{\omega},b}(i)\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)\right]$. Note that the joint iterative optimization of the interpolation filters $\{\bar{\boldsymbol{v}}_b(i)|b=1,...,B\}$ and the reduced-rank filters $\{\bar{\boldsymbol{\omega}}_b(i)|b=1,...,B\}$ are performed separately in all the processing branches.

### C. Design of the Decimation Unit

Here, we consider two strategies for the design of the decimation unit $\mathbf{T}_b(i)$. We constrain the design of $\mathbf{T}_b(i)$ so that the elements of the matrix only take the value 0 or 1. This corresponds to the decimation unit simply keeping or discarding the samples. The first strategy exhaustively explores all possible decimation patterns which select $D$ samples out of $M$ samples, this is therefore the optimal approach. In this case, the scheme can be viewed as a combinatorial problem and the total number of patterns $B$, equal to

$$B=M\cdot(M-1)\cdots(M-D+1)=\begin{pmatrix} M \\ D \end{pmatrix}. \tag{24}$$

However, the optimal decimation scheme described above is too complex for practical use since it needs $D$ permutations of M samples for each snapshot and carries out an exhaustive search over all possible patterns. Therefore, an alternative decimation scheme with low-complexity that renders itself to practical use is of great interest. To this end, we consider the second decimation scheme which we call pre-stored decimation unit (PSDU). The PSDU scheme employs a structure formed in the following way

$$\mathbf{T}_b=[\boldsymbol{\phi}_{b,1}\quad\boldsymbol{\phi}_{b,2}\quad\ldots\quad\boldsymbol{\phi}_{b,D}], \tag{25}$$

where the $M\times 1$ vector $\boldsymbol{\phi}_{b,d}$ denotes the $d$th basis vector of the $b$th decimation unit, $d=1,...,D$, $b=1,...,B$, and is composed of a single 1 and $(M-1)$ 0s, according to the following

$$\boldsymbol{\phi}_{b,d}=[\underbrace{0,\ \ldots\ ,\ 0}_{z_{b,d}},\ 1,\ \underbrace{0,\ \ldots\ ,\ 0}_{M-z_{b,d}-1}], \tag{26}$$

where $z_{b,d}$ is the number of zeros before the only element equal to one. We set the value of $z_{b,d}$ in a deterministic way which can be expressed as

$$z_{b,d}=\frac{M}{D}\times(d-1)+(b-1). \tag{27}$$

It should be remarked that other designs have been investigated and this structure has been adopted due to an excellent trade-off between performance and complexity.

## IV. Adaptive Algorithms

Adaptive implementations of the LCMV beamformer were subsequently reported with the RLS and the CG algorithms [16], [31]–[33]. Here, we develop the RLS and the CCG algorithms that adjust the parameters of the interpolation filters and the reduced-rank filters for the MPB structure based on the minimization of the CMV cost function. Furthermore, we compare the complexity of the proposed RR-SJIDF algorithms with other existing algorithms, namely, the full-rank RLS filter, the JDL, the MSWF and the AVF algorithms, in terms of multiplications and additions per snapshot.

### A. Recursive Least Squares (RLS) algorithm

Here, we describe an RLS algorithm that adaptively adjusts the coefficients of the interpolation filters $\{\bar{\boldsymbol{\upsilon}}_b(i)|b=1,...,B\}$ and the reduced-rank filters $\{\bar{\boldsymbol{\omega}}_b(i)|b=1,...,B\}$ based on the least squares (LS) cost functions, which are shown as below:

$$\begin{aligned}\mathcal{L}_{LS}(\bar{\boldsymbol{\upsilon}}_b(i)) &= \sum_{n=1}^{i}\alpha^{i-n}\left|\bar{\boldsymbol{\upsilon}}_b^H(n)\bar{\mathbf{r}}_{\bar{\upsilon},b}(n)\right|^2 \\ &\quad + 2\Re\left\{\lambda_{\bar{\upsilon},b}\left[\bar{\boldsymbol{\upsilon}}_b^H(i)\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)-1\right]\right\}, \\ \mathcal{L}_{LS}(\bar{\boldsymbol{\omega}}_b(i)) &= \sum_{n=1}^{i}\alpha^{i-n}\left|\bar{\boldsymbol{\omega}}_b^H(n)\bar{\mathbf{r}}_{\bar{\omega},b}(n)\right|^2 \\ &\quad + 2\Re\left\{\lambda_{\bar{\omega},b}\left[\bar{\boldsymbol{\omega}}_b^H(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)-1\right]\right\},\end{aligned} \tag{28}$$

where $\alpha$ is the forgetting factor. By computing the gradients of $\mathcal{L}_{LS}(\bar{\boldsymbol{\upsilon}}_b(i))$ and $\mathcal{L}_{LS}(\bar{\boldsymbol{\omega}}_b(i))$, and equating them to zero and solving for $\lambda_{\bar{\upsilon},b}$ and $\lambda_{\bar{\omega},b}$, respectively, we obtain

$$\begin{aligned}\bar{\boldsymbol{\upsilon}}_b(i) &= \frac{\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}^{-1}(i)\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)}{\bar{\mathbf{s}}_{\bar{\upsilon},b}^H(i)\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}^{-1}(i)\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)}, \\ \bar{\boldsymbol{\omega}}_b(i) &= \frac{\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}^{-1}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)}{\bar{\mathbf{s}}_{\bar{\omega},b}^H(i)\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}^{-1}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)},\end{aligned} \tag{29}$$

where $\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}(i)=\sum_{n=1}^{i}\alpha^{i-n}\bar{\mathbf{r}}_{\bar{\upsilon},b}(n)\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(n)$ and $\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)=\sum_{n=1}^{i}\alpha^{i-n}\bar{\mathbf{r}}_{\bar{\omega},b}(n)\bar{\mathbf{r}}_{\bar{\omega},b}^H(n)$ denote the time averaged correlation matrices with respect to $\bar{\boldsymbol{\omega}}_b(i)$ and $\bar{\boldsymbol{\upsilon}}_b(i)$, respectively. By employing the matrix inversion lemma, and defining $\mathbf{P}_{\bar{\upsilon},b}(i)=\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}^{-1}(i)$ and $\mathbf{P}_{\bar{\omega},b}(i)=\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}^{-1}(i)$,respectively, and the gain vectors $\bar{\mathbf{k}}_{\bar{\upsilon},b}(i)$ and $\bar{\mathbf{k}}_{\bar{\omega},b}(i)$ are expressed, respectively, as follows

$$\begin{aligned}\bar{\mathbf{k}}_{\bar{\upsilon},b}(i) &= \frac{\mathbf{P}_{\bar{\upsilon},b}(i-1)\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)}{\alpha+\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i)\mathbf{P}_{\bar{\upsilon},b}(i-1)\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)}, \\ \bar{\mathbf{k}}_{\bar{\omega},b}(i) &= \frac{\mathbf{P}_{\bar{\omega},b}(i-1)\bar{\mathbf{r}}_{\bar{\omega},b}(i)}{\alpha+\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)\mathbf{P}_{\bar{\omega},b}(i-1)\bar{\mathbf{r}}_{\bar{\omega},b}(i)},\end{aligned} \tag{30}$$

and thus we can rewrite $\mathbf{P}_{\bar{\upsilon},b}(i)$ and $\mathbf{P}_{\bar{\omega},b}(i)$ recursively as

$$\begin{aligned}\mathbf{P}_{\bar{\upsilon},b}(i) &= \alpha^{-1}\mathbf{P}_{\bar{\upsilon},b}(i-1)-\alpha^{-1}\bar{\mathbf{k}}_{\bar{\upsilon},b}(i)\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i)\mathbf{P}_{\bar{\upsilon},b}(i-1), \\ \mathbf{P}_{\bar{\omega},b}(i) &= \alpha^{-1}\mathbf{P}_{\bar{\omega},b}(i-1)-\alpha^{-1}\bar{\mathbf{k}}_{\bar{\omega},b}(i)\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)\mathbf{P}_{\bar{\omega},b}(i-1),\end{aligned} \tag{31}$$

where $\mathbf{P}_{\bar{\upsilon},b}(0)$ and $\mathbf{P}_{\bar{\omega},b}(0)$ are initialized to $\delta^{-1}\mathbf{I}$, where $\delta$ is a small positive constant and $\mathbf{I}$ is the identity matrix. It is worth remarking that $\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)$, $\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i)$,$\bar{\mathbf{s}}_{\bar{\omega},b}^H(i)$ and $\bar{\mathbf{s}}_{\bar{\upsilon},b}^H(i)$ have to be updated as soon as $\bar{\boldsymbol{\upsilon}}_b(i)$ and $\bar{\boldsymbol{\omega}}_b(i)$ are updated since they are dependent on $\bar{\boldsymbol{\omega}}_b(i)$ and $\bar{\boldsymbol{\upsilon}}_b(i)$, respectively. The output at the end of the MPB framework $y(i)$ is selected according to:

$$y(i)=y_{b_s}(i)\quad\text{when}\quad b_s=\arg\min_{1\le b\le B}|y_b(i)|^2, \tag{32}$$

where

$$y_b(i)=\bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i). \tag{33}$$

The algorithm is summaried in Table I.

TABLE I

THE SJIDF SCHEME USING THE RLS ALGORITHM

**Initialisation: for each branch** $b=1,\cdots,B$

$\mathbf{P}_{\bar{\upsilon},b}(0)=\delta^{-1}\mathbf{I}$ and $\mathbf{P}_{\bar{\omega},b}(0)=\delta^{-1}\mathbf{I}$,

$\bar{\boldsymbol{\omega}}_b(0)=[1,0,\cdots,0]^T$ and $\bar{\boldsymbol{\upsilon}}_b(0)=[1,0,\cdots,0]^T$,

$\bar{\mathbf{s}}_{\bar{\upsilon},b}(1)=\mathbf{T}_b^H\boldsymbol{\mathcal{S}}_0^H\bar{\boldsymbol{\omega}}_b(0)$,

$\bar{\mathbf{s}}_{\bar{\omega},b}(1)=\mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{\upsilon}}_b(0)$,

**Recursion: for each branch** $b=1,\cdots,B$

**and each time instant** $i=1,\cdots,K$

**STEP 1: updating** $\bar{\boldsymbol{\upsilon}}_b(i)$

$\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)=\mathbf{T}_b^H\boldsymbol{\mathcal{R}}_0^H\bar{\boldsymbol{\omega}}_b(i-1)$,

$\bar{\mathbf{k}}_{\bar{\upsilon},b}(i)=\frac{\mathbf{P}_{\bar{\upsilon},b}(i-1)\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)}{\alpha+\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i)\mathbf{P}_{\bar{\upsilon},b}(i-1)\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)}$,

$\mathbf{P}_{\bar{\upsilon},b}(i)=\alpha^{-1}\mathbf{P}_{\bar{\upsilon},b}(i-1)-\alpha^{-1}\bar{\mathbf{k}}_{\bar{\upsilon},b}(i)\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i)\mathbf{P}_{\bar{\upsilon},b}(i-1)$,

$\bar{\boldsymbol{\upsilon}}_b(i)=\frac{\mathbf{P}_{\bar{\upsilon},b}(i)\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)}{\bar{\mathbf{s}}_{\bar{\upsilon},b}^H(i)\mathbf{P}_{\bar{\upsilon},b}(i)\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)}$,

$\bar{\mathbf{s}}_{\bar{\omega},b}(i)=\mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{\upsilon}}_b(i)$,

**STEP 2: updating** $\bar{\boldsymbol{\omega}}_b(i)$

$\bar{\mathbf{r}}_{\bar{\omega},b}(i)=\mathbf{T}_b\boldsymbol{\mathcal{R}}_0\bar{\boldsymbol{\upsilon}}_b(i)$,

$\bar{\mathbf{k}}_{\bar{\omega},b}(i)=\frac{\mathbf{P}_{\bar{\omega},b}(i-1)\bar{\mathbf{r}}_{\bar{\omega},b}(i)}{\alpha+\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)\mathbf{P}_{\bar{\omega},b}(i-1)\bar{\mathbf{r}}_{\bar{\omega},b}(i)}$,

$\mathbf{P}_{\bar{\omega},b}(i)=\alpha^{-1}\mathbf{P}_{\bar{\omega},b}(i-1)-\alpha^{-1}\bar{\mathbf{k}}_{\bar{\omega},b}(i)\bar{\mathbf{r}}_{\bar{\omega},b}^H(i)\mathbf{P}_{\bar{\omega},b}(i-1)$,

$\bar{\boldsymbol{\omega}}_b(i)=\frac{\mathbf{P}_{\bar{\omega},b}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)}{\bar{\mathbf{s}}_{\bar{\omega},b}(i)^H\mathbf{P}_{\bar{\omega},b}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)}$,

$\bar{\mathbf{s}}_{\bar{\upsilon},b}(i+1)=\mathbf{T}_b^H\boldsymbol{\mathcal{S}}_0^H\bar{\boldsymbol{\omega}}_b(i)$,

**STEP 3: Calculating the output of** $b$**-th branch**

$y_b(i)=\bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{\upsilon}}_b(i)$,

**Output:**

$y(i)=y_{b_s}(i)$ when $b_s=\arg\min_{1\le b\le B}|y_b(i)|^2$.

### B. Constrained Conjugate Gradient (CCG) Algorithm

In this subsection, we develop a CCG algorithm to implement the proposed RR-SJIDF STAP. According to (22) and (23) which were derived in the previous section based on CMV criterion, let us define two intermediate vectors, CG-based weight vectors, $\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)=\bar{\mathbf{R}}_{\bar{\upsilon},b}^{-1}\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)$ and $\tilde{\bar{\boldsymbol{\omega}}}_b(i)=\bar{\mathbf{R}}_{\bar{\omega},b}^{-1}\bar{\mathbf{s}}_{\bar{\omega},b}(i)$, respectively, to solve the equations and save the computations. Thus, we may obtain $\bar{\boldsymbol{\upsilon}}_b(i)=\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)/(\bar{\mathbf{s}}_{\bar{\upsilon},b}^H(i)\tilde{\bar{\boldsymbol{\upsilon}}}_b(i))$ and $\bar{\boldsymbol{\omega}}_b(i)=\tilde{\bar{\boldsymbol{\omega}}}_b(i)/(\bar{\mathbf{s}}_{\bar{\omega},b}^H(i)\tilde{\bar{\boldsymbol{\omega}}}_b(i))$. The solutions to $\bar{\mathbf{R}}_{\bar{\upsilon},b}\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)=\bar{\mathbf{s}}_{\bar{\upsilon},b}(i)$ and $\bar{\mathbf{R}}_{\bar{\omega},b}\tilde{\bar{\boldsymbol{\omega}}}_b(i)=\bar{\mathbf{s}}_{\bar{\omega},b}(i)$, $\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)$ and $\tilde{\bar{\boldsymbol{\omega}}}_b(i)$, respectively, are given by solving two optimization problems as follows [33]–[35]

$$\begin{aligned}\Phi(\tilde{\bar{\boldsymbol{\upsilon}}}_b) &= \tilde{\bar{\boldsymbol{\upsilon}}}_b^H(i)\bar{\mathbf{R}}_{\bar{\upsilon},b}\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)+2\Re\left\{\bar{\mathbf{s}}_{\bar{\upsilon},b}^H(i)\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)\right\}, \\ \tilde{\bar{\boldsymbol{\upsilon}}}_b(i) &= \underset{\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)\in\mathcal{C}^{I\times 1}}{\arg\min}\ \Phi(\tilde{\bar{\boldsymbol{\upsilon}}}_b),\end{aligned} \tag{34}$$

and

$$\begin{aligned}\Phi(\tilde{\bar{\boldsymbol{\omega}}}_b) &= \tilde{\bar{\boldsymbol{\omega}}}_b^H(i)\bar{\mathbf{R}}_{\bar{\omega},b}\tilde{\bar{\boldsymbol{\omega}}}_b(i)+2\Re\left\{\bar{\mathbf{s}}_{\bar{\omega},b}^H(i)\tilde{\bar{\boldsymbol{\omega}}}_b(i)\right\}, \\ \tilde{\bar{\boldsymbol{\omega}}}_b(i) &= \underset{\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)\in\mathcal{C}^{D\times 1}}{\arg\min}\ \Phi(\tilde{\bar{\boldsymbol{\omega}}}_b),\end{aligned} \tag{35}$$

where $\Phi(\tilde{\bar{\boldsymbol{\upsilon}}}_b)$ and $\Phi(\tilde{\bar{\boldsymbol{\omega}}}_b)$ are cost functions with respect to $\tilde{\bar{\boldsymbol{\upsilon}}}_b(i)$ and $\tilde{\bar{\boldsymbol{\omega}}}_b(i)$, respectively. The correlation matrices $\bar{\mathbf{R}}_{\bar{\upsilon},b}$ and $\bar{\mathbf{R}}_{\bar{\omega},b}$, respectively, are estimated by

$$\begin{aligned}\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}(i) &= \lambda_f\hat{\bar{\mathbf{R}}}_{\bar{\upsilon},b}(i)+\bar{\mathbf{r}}_{\bar{\upsilon},b}(i)\bar{\mathbf{r}}_{\bar{\upsilon},b}^H(i), \\ \hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i) &= \lambda_f\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)+\bar{\mathbf{r}}_{\bar{\omega},b}(i)\bar{\mathbf{r}}_{\bar{\omega},b}^H(i),\end{aligned} \tag{36}$$

where $\lambda_f$ is the forgetting factor. Let us define $\mathbf{g}_{\bar{v},b}(i)$ and $\mathbf{g}_{\bar{\omega},b}(i)$ as residual vectors which are expressed, respectively, as follows

$$\begin{aligned}\mathbf{g}_{\bar{v},b}(i) &= -\nabla_{\tilde{\boldsymbol{v}}_b}\Phi(\tilde{\boldsymbol{v}}_b)\\ &= \bar{\mathbf{s}}_{\bar{v},b}(i) - \hat{\bar{\mathbf{R}}}_{\bar{v},b}(i)\tilde{\boldsymbol{v}}_b(i),\end{aligned} \tag{37}$$

and

$$\begin{aligned}\mathbf{g}_{\bar{\omega},b}(i) &= -\nabla_{\tilde{\boldsymbol{\omega}}_b}\Phi(\tilde{\boldsymbol{\omega}}_b)\\ &= \bar{\mathbf{s}}_{\bar{\omega},b}(i) - \hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)\tilde{\boldsymbol{\omega}}_b(i).\end{aligned} \tag{38}$$

Thus, the CG-based weight vectors $\tilde{\boldsymbol{v}}_b(i)$ and $\tilde{\boldsymbol{\omega}}_b$ can be recursively written as [36]

$$\begin{aligned}\tilde{\boldsymbol{v}}_b(i) &= \tilde{\boldsymbol{v}}_b(i-1) + \alpha_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i),\\ \tilde{\boldsymbol{\omega}}_b(i) &= \tilde{\boldsymbol{\omega}}_b(i-1) + \alpha_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i),\end{aligned} \tag{39}$$

where $\alpha_{\bar{v},b}(i)$ and $\alpha_{\bar{\omega},b}(i)$ denote the step sizes. $\mathbf{p}_{\bar{v},b}(i)$ and $\mathbf{p}_{\bar{\omega},b}(i)$ denote the direction vectors. According to [36], $\alpha_{\bar{v},b}(i)$, $\alpha_{\bar{\omega},b}(i)$, $\mathbf{p}_{\bar{v},b}(i)$ and $\mathbf{p}_{\bar{\omega},b}(i)$ can, respectively, be given by

$$\begin{aligned}\alpha_{\bar{v},b}(i) &= \Big\{\lambda_f\left[\mathbf{p}^H_{\bar{v},b}(i)\mathbf{g}_{\bar{v},b}(i-1) - \mathbf{p}^H_{\bar{v},b}(i)\bar{\mathbf{s}}_{\bar{v},b}(i)\right]\\ &\quad - \eta_{\bar{v}}\mathbf{p}^H_{\bar{v},b}(i)\mathbf{g}_{\bar{v},b}(i-1)\Big\}\left[\mathbf{p}^H_{\bar{v},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i)\right]^{-1},\\ \alpha_{\bar{\omega},b}(i) &= \Big\{\lambda_f\left[\mathbf{p}^H_{\bar{\omega},b}(i)\mathbf{g}_{\bar{\omega},b}(i-1) - \mathbf{p}^H_{\bar{\omega},b}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)\right]\\ &\quad - \eta_{\bar{\omega}}\mathbf{p}^H_{\bar{\omega},b}(i)\mathbf{g}_{\bar{\omega},b}(i-1)\Big\}\left[\mathbf{p}^H_{\bar{\omega},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i)\right]^{-1},\\ \mathbf{p}_{\bar{v},b}(i) &= \mathbf{g}_{\bar{v},b}(i-1) + \beta_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i),\\ \mathbf{p}_{\bar{\omega},b}(i) &= \mathbf{g}_{\bar{\omega},b}(i-1) + \beta_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i),\end{aligned} \tag{40}$$

where $0 \le \eta_{\bar{v}}, \eta_{\bar{\omega}} \le 0.5$, $\beta_{\bar{v},b}(i)$ and $\beta_{\bar{\omega},b}(i)$ can be computed as

$$\begin{aligned}\beta_{\bar{v},b}(i) &= \frac{\mathbf{g}^H_{\bar{v},b}(i)\left[\mathbf{g}_{\bar{v},b}(i) + \lambda_f\bar{\mathbf{s}}_{\bar{v},b}(i) - \mathbf{g}_{\bar{v},b}(i-1)\right]}{\mathbf{g}^H_{\bar{v},b}(i-1)\mathbf{g}_{\bar{v},b}(i-1)},\\ \beta_{\bar{\omega},b}(i) &= \frac{\mathbf{g}^H_{\bar{\omega},b}(i)\left[\mathbf{g}_{\bar{\omega},b}(i) + \lambda_f\bar{\mathbf{s}}_{\bar{\omega},b}(i) - \mathbf{g}_{\bar{\omega},b}(i-1)\right]}{\mathbf{g}^H_{\bar{\omega},b}(i-1)\mathbf{g}_{\bar{\omega},b}(i-1)}.\end{aligned} \tag{41}$$

Thus, the interpolation filters $\bar{\boldsymbol{v}}_b(i)$ and the reduced-rank filters $\bar{\boldsymbol{\omega}}_b(i)$ can be written as $\bar{\boldsymbol{v}}_b(i) = \tilde{\boldsymbol{v}}_b(i)/(\bar{\mathbf{s}}^H_{\bar{v},b}(i)\tilde{\boldsymbol{v}}_b(i))$ and $\bar{\boldsymbol{\omega}}_b(i) = \tilde{\boldsymbol{\omega}}_b(i)/(\bar{\mathbf{s}}^H_{\bar{\omega},b}(i)\tilde{\boldsymbol{\omega}}_b(i))$ based on the CG-based weight vectors, respectively. The adaptive implementation of the proposed RR-SJIDF STAP using the CCG algorithm is summarised in Table II

TABLE II

THE SJIDF SCHEME USING THE CCG ALGORITHM

**Initialisation: for each branch** $b = 1, \cdots, B$

$\tilde{\boldsymbol{\omega}}_b(0) = [1, 0, \cdots, 0]^T$ and $\tilde{\boldsymbol{v}}_b(0) = [1, 0, \cdots, 0]^T$,

$\bar{\mathbf{s}}_{\bar{v},b}(1) = \mathbf{T}_b^H\boldsymbol{\mathcal{S}}_0^H\bar{\boldsymbol{\omega}}_b(0)$ and $\bar{\mathbf{s}}_{\bar{\omega},b}(1) = \mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{v}}_b(0)$,

$\mathbf{g}_{\bar{v},b}(0) = \bar{\mathbf{s}}_{\bar{v},b}(1)$ and $\mathbf{p}_{\bar{v},b}(1) = \mathbf{g}_{\bar{v},b}(0)$,

$\mathbf{g}_{\bar{\omega},b}(0) = \bar{\mathbf{s}}_{\bar{\omega},b}(1)$ and $\mathbf{p}_{\bar{\omega},b}(1) = \mathbf{g}_{\bar{\omega},b}(0)$,

$\hat{\mathbf{R}}_{\bar{v},b}(0) = \delta^{-1}\mathbf{I}$ and $\hat{\mathbf{R}}_{\bar{\omega},b}(0) = \delta^{-1}\mathbf{I}$,

$\bar{\boldsymbol{v}}_b(0) = \tilde{\boldsymbol{v}}_b(0)/(\bar{\mathbf{s}}^H_{\bar{v},b}(1)\tilde{\boldsymbol{v}}_b(0))$,

$\bar{\boldsymbol{\omega}}_b(0) = \tilde{\boldsymbol{\omega}}_b(0)/(\bar{\mathbf{s}}^H_{\bar{\omega},b}(1)\tilde{\boldsymbol{\omega}}_b(0))$,

**Recursion: for each branch** $b = 1, \cdots, B$

**and each time instant** $i = 1, \cdots, K$

**STEP 1: updating** $\bar{\boldsymbol{v}}_b(i)$

$\bar{\mathbf{r}}_{\bar{v},b}(i) = \mathbf{T}_b^H\boldsymbol{\mathcal{R}}_0^H\bar{\boldsymbol{\omega}}_b(i-1)$,

$\hat{\bar{\mathbf{R}}}_{\bar{v},b}(i) = \lambda_f\hat{\bar{\mathbf{R}}}_{\bar{v},b}(i) + \bar{\mathbf{r}}_{\bar{v},b}(i)\bar{\mathbf{r}}^H_{\bar{v},b}(i)$,

$\alpha_{\bar{v},b}(i) = \Big\{\lambda_f\left[\mathbf{p}^H_{\bar{v},b}(i)\mathbf{g}_{\bar{v},b}(i-1) - \mathbf{p}^H_{\bar{v},b}(i)\bar{\mathbf{s}}_{\bar{v},b}(i)\right] - \eta_{\bar{v}}\mathbf{p}^H_{\bar{v},b}(i)\mathbf{g}_{\bar{v},b}(i-1)\Big\}\left[\mathbf{p}^H_{\bar{v},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i)\right]^{-1}$,

$\mathbf{g}_{\bar{v},b}(i) = \lambda_f\mathbf{g}_{\bar{v},b}(i-1) - \alpha_{\bar{v},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i)$,

$\tilde{\boldsymbol{v}}_b(i) = \tilde{\boldsymbol{v}}_b(i-1) + \alpha_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i)$,

$\beta_{\bar{v},b}(i) = \frac{\mathbf{g}^H_{\bar{v},b}(i)\left[\mathbf{g}_{\bar{v},b}(i) + \lambda_f\bar{\mathbf{s}}_{\bar{v},b}(i) - \mathbf{g}_{\bar{v},b}(i-1)\right]}{\mathbf{g}^H_{\bar{v},b}(i-1)\mathbf{g}_{\bar{v},b}(i-1)}$,

$\mathbf{p}_{\bar{v},b}(i) = \mathbf{g}_{\bar{v},b}(i-1) + \beta_{\bar{v},b}(i)\mathbf{p}_{\bar{v},b}(i)$,

$\bar{\boldsymbol{v}}_b(i) = \tilde{\boldsymbol{v}}_b(i)/(\bar{\mathbf{s}}^H_{\bar{v},b}(i)\tilde{\boldsymbol{v}}_b(i))$,

$\bar{\mathbf{s}}_{\bar{\omega},b}(i) = \mathbf{T}_b\boldsymbol{\mathcal{S}}_0\bar{\boldsymbol{v}}_b(i)$,

**STEP 2: updating** $\bar{\boldsymbol{\omega}}_b(i)$

$\bar{\mathbf{r}}_{\bar{\omega},b}(i) = \mathbf{T}_b\boldsymbol{\mathcal{R}}_0\bar{\boldsymbol{v}}_b(i)$,

$\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i) = \lambda_f\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i) + \bar{\mathbf{r}}_{\bar{\omega},b}(i)\bar{\mathbf{r}}^H_{\bar{\omega},b}(i)$

$\alpha_{\bar{\omega},b}(i) = \Big\{\lambda_f\left[\mathbf{p}^H_{\bar{\omega},b}(i)\mathbf{g}_{\bar{\omega},b}(i-1) - \mathbf{p}^H_{\bar{\omega},b}(i)\bar{\mathbf{s}}_{\bar{\omega},b}(i)\right] - \eta_{\bar{\omega}}\mathbf{p}^H_{\bar{\omega},b}(i)\mathbf{g}_{\bar{\omega},b}(i-1)\Big\}\left[\mathbf{p}^H_{\bar{\omega},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i)\right]^{-1}$,

$\mathbf{g}_{\bar{\omega},b}(i) = \lambda_f\mathbf{g}_{\bar{\omega},b}(i-1) - \alpha_{\bar{\omega},b}(i)\hat{\bar{\mathbf{R}}}_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i)$,

$\tilde{\boldsymbol{\omega}}_b(i) = \tilde{\boldsymbol{\omega}}_b(i-1) + \alpha_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i)$,

$\beta_{\bar{\omega},b}(i) = \frac{\mathbf{g}^H_{\bar{\omega},b}(i)\left[\mathbf{g}_{\bar{\omega},b}(i) + \lambda_f\bar{\mathbf{s}}_{\bar{\omega},b}(i) - \mathbf{g}_{\bar{\omega},b}(i-1)\right]}{\mathbf{g}^H_{\bar{\omega},b}(i-1)\mathbf{g}_{\bar{\omega},b}(i-1)}$,

$\mathbf{p}_{\bar{\omega},b}(i) = \mathbf{g}_{\bar{\omega},b}(i-1) + \beta_{\bar{\omega},b}(i)\mathbf{p}_{\bar{\omega},b}(i)$,

$\bar{\boldsymbol{\omega}}_b(i) = \tilde{\boldsymbol{\omega}}_b(i)/(\bar{\mathbf{s}}^H_{\bar{\omega},b}(i)\tilde{\boldsymbol{\omega}}_b(i))$,

$\bar{\mathbf{s}}_{\bar{v},b}(i+1) = \mathbf{T}_b^H\boldsymbol{\mathcal{S}}_0^H\bar{\boldsymbol{\omega}}_b(i)$,

**STEP 3: Calculating the output of** $b$**-th branch**

$y_b(i) = \bar{\boldsymbol{\omega}}_b^H(i)\mathbf{T}_b\boldsymbol{\mathcal{R}}_0(i)\bar{\boldsymbol{v}}_b(i)$,

**Output:**

$y(i) = y_{b_s}(i)$ when $b_s = \arg\min_{1\le b\le B}|y_b(i)|^2$.

### C. Branch and Rank Selection

The performance of the algorithms described in the previous subsections highly depends on the parameters including the ranks $D$, $I$ and the number of branches $B$. In this subsection, we discuss the parameter settings to meet the best trade-off between the performance and the complexity. We have mentioned that in the previous section that the optimal number of branches is in (24), which is quite large. Within such range, we can claim that more branches, better performance for the proposed algorithm. However, considering the affordable complexity, we have to configure the algorithm with the number of branches as small as possible and meanwhile achieving competitive performance. As will be shown in the simulation results, the proposed algorithms with the number of branches $B$ equal to 4 or 5 have good trade-offs between the performance and the complexity. Since the performance of the proposed RR-SJIDF algorithm is also sensitive to the ranks $D$ and $I$, we present adaptation methods for automatically selecting the ranks of the algorithms based on the exponentially

weighted *a posteriori* LS type cost function described by

$$\mathcal{C}(\bar{\boldsymbol{\omega}}_b^{(D)}, \bar{\boldsymbol{v}}_b^{(I)}) = \sum_{l=1}^{i} \alpha^{i-l} \left| \bar{\boldsymbol{\omega}}_b^{H,(D)}(l) \mathbf{T}_b \boldsymbol{\mathcal{R}}_0(l) \bar{\boldsymbol{v}}_b^{(I)}(l) \right|^2 \quad (42)$$

where $\alpha$ is the forgetting factor, $\bar{\boldsymbol{\omega}}_b^{(D)}(i)$ is the reduced-rank filter with rank $D$ and $\bar{\boldsymbol{v}}_b^{(I)}(i)$ is the interpolator filter with rank $I$. For each time instant $i$ and a given decimation pattern $\mathbf{T}_b$ and $B$, we select the ranks $D$ and $I$ to minimize $\mathcal{C}(\bar{\boldsymbol{\omega}}_b^{(D)}, \bar{\boldsymbol{v}}_b^{(I)})$. The proposed rank adaptation algorithm that chooses the best ranks $D_{opt}$ and $I_{opt}$ for the filters $\bar{\boldsymbol{\omega}}_b(i)$ and $\bar{\boldsymbol{v}}_b(i)$, respectively, is given by

$$\{D_{opt}, I_{opt}\} = \underset{\substack{I_{min} \le I \le I_{max} \\ D_{min} \le D \le D_{max}}}{\arg\min} \mathcal{C}(\bar{\boldsymbol{\omega}}_b^{(D)}, \bar{\boldsymbol{v}}_b^{(I)}), \quad (43)$$

where $D_{min}$ and $D_{max}$, $I_{min}$ and $I_{max}$ are the minimum, maximum ranks allowed for the reduced-rank filters and interpolators, respectively. Note that a smaller rank may produce faster adaptation during the initial stages of estimation procedure and a slightly greater rank usually yields a better steady-state performance. Although the rank adaptation increases the computational complexity, two benefits can be achieved: one is that the ranks, which are crucial to the proposed algorithm, can be selected automatically, and the other is that the performance is much enhanced, which will be shown in the simulation results.

### D. Complexity Analysis

We detail the computational complexity in terms of additions and multiplications of the proposed schemes with the RLS and the CCG algorithms, and other existing algorithms, namely the full-rank RLS filter, the JDL, the MSWF-RLS and the AVF algorithms as shown in Table III. Note that the complexity of our proposed SJIDF scheme is dependent on the size of the interpolator and the reduced-rank filter($I$ and $D$) and the number of branches $B$, rather than the system size $M$. There is a tradeoff between complexity and performance when we set the parameters $I$, $D$ and $B$. We found that the proposed scheme with $B = 4$, $D = 4$ and $I = 16$ works well, as will be verified in the simulation results. The computational complexity of all algorithms is shown in Fig. 3, where we can find that the proposed schemes using both the RLS and the CCG algorithms have significantly lower complexity than other algorithms, expect the JDL algorithm. As will be seen in the simulation results, the JDL algorithm performs poorly in steady state and our proposed algorithms outperform the JDL algorithm in both convergence speed and steady-state performance.

## V. Analysis of the Optimization Problem

Let us now study the convergence properties of the proposed scheme. With respect to global convergence, a sufficient but not necessary condition is the convexity of the cost function, which is verified if its Hessian matrix is positive semi-definite. The method leads to an optimization problem with multiple solutions due to the discrete nature of $\mathbf{T}_b$ and the switching between branches. Therefore, the convergence of the

TABLE III

Comparison of the computational complexity.

| Algorithm | Number of operations per snapshot | |
|---|---|---|
| | Additions | Multiplications |
| Full-Rank-RLS | $6M^2 - 8M + 3$ | $6M^2 + 2M + 2$ |
| JDL-RLS | $DM + 4D^2 - D - 2$ | $DM + 5D^2 + 5D$ |
| MSWF-RLS | $(D+1)M^2 + 6D^2 - 8D + 2$ | $(D+1)M^2 + 2DM + 3D + 2$ |
| AVF | $D(M^2 + 3(M-1)^2) - 1 + D(5(M-1)+1) + 2M$ | $D(4M^2 + 4M + 1) + 4M + 2$ |
| SJIDF-RLS | $5D^2B + 5DIB + 4DB + 5I^2B + 3IB + 2B$ | $4D^2B + 5DIB - 3DB + 4I^2B - 3IB - 3B$ |
| SJIDF-CCG | $4DIB + 3D^2B + 12DB + 3I^2B + 12IB + 6B$ | $4DIB + D^2B + 5DB + I^2B + 5IB - 6B$ |

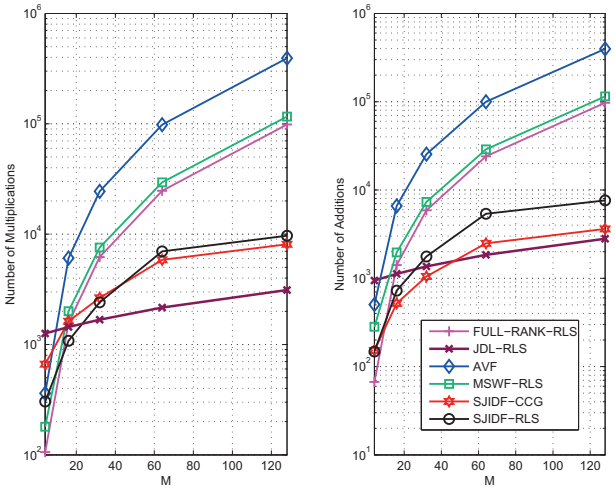


Fig. 3. The computational complexity analysis.

algorithms is not guaranteed to the global minimum since local minima may be encountered by the proposed RLS and CCG algorithms. It should be mentioned, however, that the proposed scheme is composed of several independent branches, and independent optimization problems, which are considerd to minimize the output energy with constraint for each single branch. Firstly, we consider an analysis of the optimization problem of single branch of joint interpolation, decimation and filtering method from the point of view of the cost function and constraints. We examine three cases of adaptation and discuss the nature of the optimization problem. Let us drop the time index $(i)$ and the branch index $b$ for simplicity, thus, the cost function in (21) can be rewritten as

$$\mathcal{L}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{\omega}}) = \mathbb{E}\left[ \left| \bar{\boldsymbol{\omega}}^H \mathbf{T} \boldsymbol{\mathcal{R}}_0 \bar{\boldsymbol{v}} \right|^2 \right] + 2\Re\left\{ \lambda \left[ \bar{\boldsymbol{\omega}}^H \mathbf{T} \boldsymbol{\mathcal{S}}_0 \bar{\boldsymbol{v}} - 1 \right] \right\}. \quad (44)$$

We will consider three cases of interest for our analysis as follows:

For case 1), we assume $\bar{\boldsymbol{\omega}}$ is fixed and $\bar{\boldsymbol{v}}$ is time-variant. The cost function in (44) can be rewritten as

$$\mathcal{L}(\bar{\boldsymbol{v}}) = \mathbb{E}\left[ \left| \bar{\boldsymbol{v}}^H \bar{\mathbf{r}}_{\bar{\boldsymbol{v}}} \right|^2 \right] + 2\Re\left\{ \lambda \left[ \bar{\boldsymbol{v}}^H \bar{\mathbf{s}}_{\bar{\boldsymbol{v}}} - 1 \right] \right\}, \quad (45)$$

where $\bar{\mathbf{r}}_{\bar{\boldsymbol{v}}} = \boldsymbol{\mathcal{R}}_0^H \mathbf{T}^H \bar{\boldsymbol{\omega}}$ and $\bar{\mathbf{s}}_{\bar{\boldsymbol{v}}} = \boldsymbol{\mathcal{S}}_0^H \mathbf{T}^H \bar{\boldsymbol{\omega}}$. The Hessian

matrix respect to $\bar{\boldsymbol{\upsilon}}$ is given by

$$\mathcal{H}_{\bar{\boldsymbol{\upsilon}}} = \frac{\partial\partial\mathcal{L}(\bar{\boldsymbol{\upsilon}})}{\partial\bar{\boldsymbol{\upsilon}}^H\partial\bar{\boldsymbol{\upsilon}}} = \mathbb{E}\{\bar{\mathbf{r}}_{\bar{\boldsymbol{\upsilon}}}\bar{\mathbf{r}}_{\bar{\boldsymbol{\upsilon}}}^H\} = \bar{\mathbf{R}}_{\bar{\boldsymbol{\upsilon}}}, \tag{46}$$

where $\bar{\mathbf{R}}_{\bar{\boldsymbol{\upsilon}}}$ is a positive semi-definite matrix, which means that $\mathcal{L}(\bar{\boldsymbol{\upsilon}}, \bar{\boldsymbol{\omega}})$ is a convex function of $\bar{\boldsymbol{\upsilon}}$ conditioned on the fixed $\bar{\boldsymbol{\omega}}$.

For case 2), we suppose $\bar{\boldsymbol{\omega}}$ is time-variant and $\bar{\boldsymbol{\upsilon}}$ is fixed. Using the same procedure of case 1), we may obtain the Hessian matrix respect to $\bar{\boldsymbol{\omega}}$ as

$$\mathcal{H}_{\bar{\boldsymbol{\omega}}} = \mathbb{E}\{\bar{\mathbf{r}}_{\bar{\boldsymbol{\omega}}}\bar{\mathbf{r}}_{\bar{\boldsymbol{\omega}}}^H\} = \bar{\mathbf{R}}_{\bar{\boldsymbol{\omega}}}, \tag{47}$$

where $\bar{\mathbf{r}}_{\bar{\boldsymbol{\omega}}} = \mathbf{T}\boldsymbol{\mathcal{R}}_0\bar{\boldsymbol{\upsilon}}$ and $\bar{\mathbf{R}}_{\bar{\boldsymbol{\omega}}}$ is a positive semi-definite matrix. In this case, $\mathcal{L}(\bar{\boldsymbol{\upsilon}}, \bar{\boldsymbol{\omega}})$ is a convex function of $\bar{\boldsymbol{\omega}}$ conditioned on the fixed $\bar{\boldsymbol{\upsilon}}$.

For case 3): we consider that both $\bar{\boldsymbol{\omega}}$ and $\bar{\boldsymbol{\upsilon}}$ are time-variant and the problem is to jointly optimize the two adaptive filters. The cost function in (44) is rewritten as

$$\mathcal{L}(\boldsymbol{\zeta}) = \mathbb{E}\left[\left|\boldsymbol{\zeta}^H\mathbf{A}\boldsymbol{\zeta}\right|^2\right] + 2\lambda\Re\left[\boldsymbol{\zeta}^H\mathbf{B}\boldsymbol{\zeta} - 1\right], \tag{48}$$

where $\boldsymbol{\zeta} = [\bar{\boldsymbol{\upsilon}}^T \;\; \bar{\boldsymbol{\omega}}^T]^T$ is $(I+D)\times 1$ vector, $\mathbf{A}_0(i)$ and $\mathbf{B}_0(i)$ are $(I+D)\times(I+D)$ matrices written by

$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{T}\boldsymbol{\mathcal{R}}_0 & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{T}\boldsymbol{\mathcal{S}}_0 & \mathbf{0} \end{bmatrix},$$

respectively. Thus, the Hessian matrix is given by

$$\begin{aligned} \mathcal{H}_{\boldsymbol{\zeta}} &= \frac{\partial\partial\mathcal{L}(\boldsymbol{\zeta})}{\partial\boldsymbol{\zeta}^H\partial\boldsymbol{\zeta}} \\ &= 2\mathbb{E}\left\{\mathbf{A}_0\boldsymbol{\zeta}\boldsymbol{\zeta}^H\mathbf{A}_0^H\right\} + 2\mathbb{E}\left\{\boldsymbol{\zeta}^H\mathbf{A}_0^H\boldsymbol{\zeta}\mathbf{A}_0\right\} + 2\lambda\mathbf{B}_0(i). \end{aligned} \tag{49}$$

In this case, the optimization problem depends on the parameters $\bar{\boldsymbol{\omega}}$, $\bar{\boldsymbol{\upsilon}}$ and $\lambda$, which suggests a nonconvex problem. However, convexity is a sufficient, but not necessary condition for the property that the cost function has no points of local minima. In our case, we conjecture that every point is possibly a point of global minima. To verify that, we carried out a number of studies and find that for a given decimation unit, the algorithms always converge to the same minima regardless of the initialization, provided $\bar{\boldsymbol{\omega}}$, $\bar{\boldsymbol{\upsilon}}$ are not all-zero quantities. An analysis of this problem remains an interesting open problem.

Based on the discussion above, a single branch *global* minima $\boldsymbol{\zeta}_b^{\star}$ can be provided by each branch. Thus, we can obtain a set of such minimas, which actually are local minimas relative to the overall optimization problem. Therefore the overall global minima can be obtained by

$$\boldsymbol{\zeta}_o^{\star} = \underset{\boldsymbol{\zeta}^{\star}\in\{\boldsymbol{\zeta}_b^{\star}|b=1,\cdots,B\}}{\arg\min} \mathcal{L}(\boldsymbol{\zeta}^{\star}). \tag{50}$$

Note that the overall global minima can be found when $B$ and the decimation units are properly selected.

TABLE IV
RADAR SYSTEM PARAMETERS

| Parameter | Value |
|---|---|
| Antenna array | Sideway-looking array (SLA) |
| Carrier frequency ($f_c$) | 450 MHz |
| Transmit pattern | Uniform |
| PRF ($f_r$) | 300 Hz |
| Platform velocity ($v$) | 50 m/s |
| Platform height ($h$) | 9000 m |
| Clutter-to-Noise ratio (CNR) | 40 dB |
| Jammer-to-Noise ratio (JNR) | 40 dB |
| Antenna setting I: | |
| Elements of sensors ($N$) | 10 |
| Number of Pulses ($J$) | 8 |
| Antenna setting II: | |
| Elements of sensors ($N$) | 8 |
| Number of Pulses ($J$) | 10 |

## VI. PERFORMANCE ASSESSMENT

In this section, we assess the proposed RR-SJIDF STAP algorithm using simulated radar data. The parameters of the simulated radar platform are shown in Table IV. For all simulations, we assume the presence of a mixture of two broadband jammers at $-45°$ and $60°$ with jammer-to-noise-ratio (JNR) equal to 40 dB. The clutter-to-noise-ratio (CNR) is fixed at 40 dB. All presented results are averages over 1000 independent Monte-Carlo runs.

### A. Setting of Parameters

In the first several experiments, we evaluate the SINR performance of our proposed RR-SJIDF scheme with different selections of $B$, $I$ and $D$. We investigate RR-SJIDF scheme with the RLS algorithm in two antenna settings with $M = 80$ for both. The first setting is to configure the number of elements $N = 10$ and the number of pulses $J = 8$, and the second is to configure $N = 8$ and $J = 10$. The evaluation of the SINR performance against the number of branches $B$ is shown in Fig. 4. We consider the RR-SJIDF-RLS algorithm with different values of $I$ and $D$ in both antenna settings. The results indicate that the RR-SJIDF-RLS algorithm using $B = 4$ can achieve approximately the same performance of that using more than 4 branches. Thus, in our case, to meet the best trade-off between the performance and the complexity, we normally choose $B = 4$ in our simulations. In Fig. 5, the SINR performance against the rank $D$ is shown. We can find that for the first antenna setting, the proposed scheme achieves the best performance with $D = 4$ when $I = 16$ and $B = 4$, while for the second antenna setting, the scheme achieves the best performance with $D = 5$ when $I = 13$ and $B = 4$. The results indicate an interesting fact that the selection of ranks $D$ and $I$ is highly related to the antenna setting, in other words, it is related to the structure of the received signal. That means the performance of the reduced-rank STAP algorithms can be improved if the structure of the received signal are well explored.

In the next experiment, we evaluate the SINR performance against the interpolator rank $I$ for the proposed RR-SJIDF-RLS algorithm with different $B$ and $D$, which are shown in

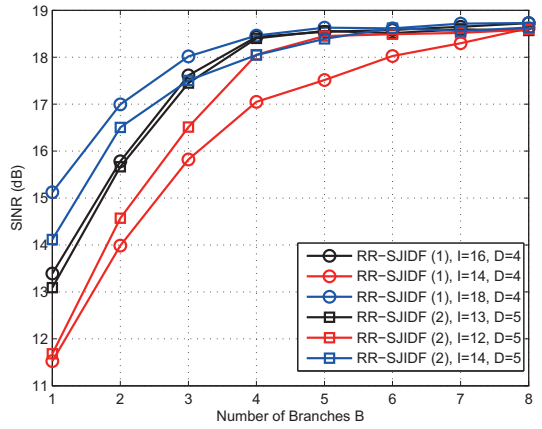


Fig. 4. SINR performance vs the number of branches $B$ with different values of $I$ and $D$, $M$ = 80, $\alpha$ = 0.9998, $K$ = 100 snapshots. (1) $N = 10$ and $J = 8$ antenna setting, (2) $N = 8$ and $J = 10$ antenna setting.

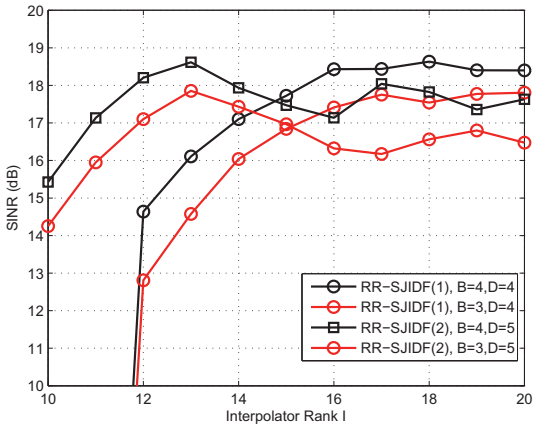


Fig. 6. SINR performance vs the interpolator rank $I$ with $M$ = 80, $\alpha$ = 0.9998, $K$ = 100 snapshots. (1) $N = 10$ and $J = 8$ antenna setting, (2) $N = 8$ and $J = 10$ antenna setting.

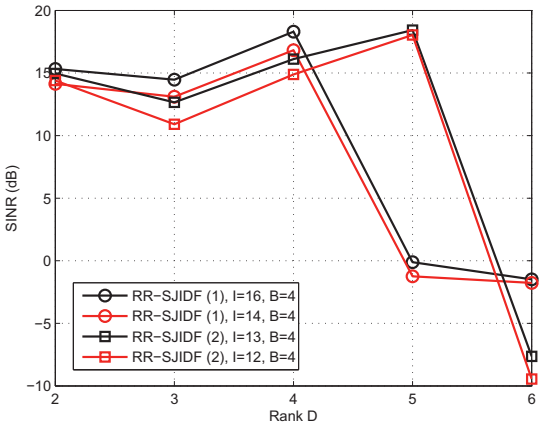


Fig. 5. SINR performance vs the rank $D$ with $M$ = 80, $\alpha$ = 0.9998, $K$ = 100 snapshots. (1) $N = 10$ and $J = 8$ antenna setting, (2) $N = 8$ and $J = 10$ antenna setting.

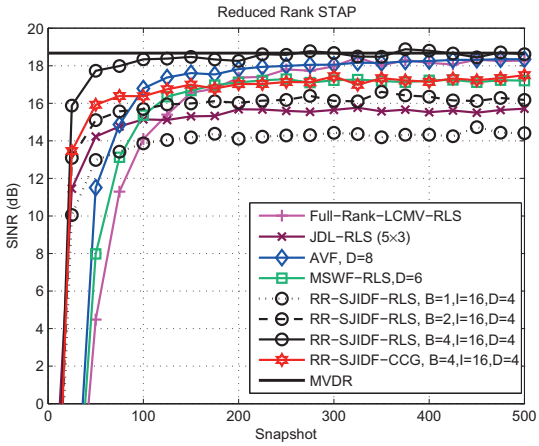


Fig. 7. SINR performance against snapshot with $M$ = 80, SNR = 0 dB, $\alpha$ = 0.9998. All algorithms are initialized to a scaled identity matrix $\delta^{-1}\mathbf{I}$, where $\delta$ is a small constant.

Fig. 6. The proposed scheme can improve the performance and converge fast if it is able to construct an appropriate subspace projection with proper coefficients in $\bar{\omega}_b(i)$ and $\bar{\upsilon}_b(i)$. Thus, for this reason and to keep a low complexity we adopt $I = 16$ and $D = 4$ for the first antenna setting and $I = 13$ and $D = 5$ for the second antenna setting since these values yield the best performance. In the folowing subsection, we will focus on the performance assessment of the proposed STAP scheme with $B = 4$, $I = 16$ and $D = 4$ for the antenna setting I.

### *B. Comparison with Existing Algorithms*

In this subsection, we compare both the SINR performance against the number of snapshots and the $P_D$ performance against the signal-to-noise-ratio (SNR) for the different designs of linear receiver using the full-rank filter with the RLS algorithm, the MSWF with the RLS algorithm, the AVF and our proposed technique, where the reduced-rank filter $\bar{\omega}(i)$ with $D$ coefficients provides an estimate to determine whether the target is present or not.

Firstly, as shown in Fig. 7, we evaluate the SINR against the number of snapshots $K$ performance of our proposed algorithm with different setting parameters and compare with the other schemes. The schemes are simulated over $K = 500$ snapshots and the SNR is set at 0 dB. The curves show an excellent performance by the proposed algorithm, which also converges much faster than other schemes. With the number of branches $B = 4$, the proposed scheme approaches the optimal MVDR performance after 50 snapshots. As one may expect, with an increase in the number of branches, the steady SINR performance improves.

In the second experiment, in Fig. 8, we present $P_D$ versus SNR performance for all schemes using 50 snapshots as the training data. The false alarm rate $P_{FA}$ is set to $10^{-6}$ and we suppose the target is injected in the boresight (0°) with Doppler frequency 100Hz. The figure illustrates that the pro-

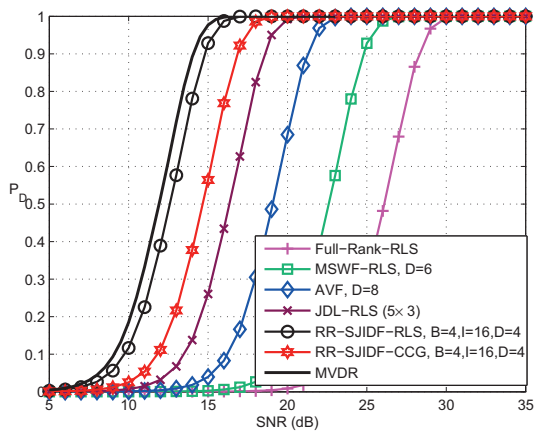


Fig. 8. Probability of detection performance vs SNR with $M$ = 80, $\alpha$ = 0.9998, $K$ = 50 snapshots, $P_{FA} = 10^{-6}$.

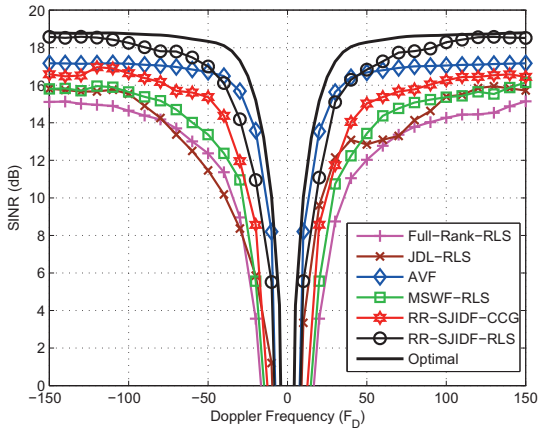


Fig. 9. SINR performance against Doppler frequency ($F_D$) with $M$ = 80, $\alpha$ = 0.9998, $K$ = 100 snapshots.

posed algorithm provides sub-optimal detection performance using very short support data, but remarkably, obtains a 90 percent detection rate, beating 50 percent for the AVF, 40 percent for the MSWF with the RLS and 30 percent for the full rank filter with the RLS at an SNR level of 15 dB.

We evaluate the SINR performance against the target Doppler frequency at the main bean look angle for our proposed algorithms and other existing algorithms, which are illustrated in Fig. 9. The potential Doppler frequency space form -150 to 150 Hz is examined and 100 snapshots are used to train the filter. The plots show that our proposed algorithms converge and approach the optimum in a short time, and form a deep null to cancel the mainbeam clutter. Note that the proposed RR-SJIDF-RLS algorithm outperforms other algorithms in the most of Doppler bins, but performs slightly worse than the AVF algorithm in the Doppler range of -50 to 50Hz.

## VII. Conclusions

In this paper, we proposed an RR-SJIDF STAP scheme for airborne radar systems. The proposed scheme performed dimensionality reduction by employing a MPB framework, which jointly optimizes interpolation, decimation and filtering units. The output was switched to the branch with the best performance according to the minimum variance criterion. In order to design the decimation unit, we considered the optimal decimation scheme and also a low-complexity pre-stored decimation units scheme. Furthermore, we developed an adaptive RLS algorithm for efficient implementation of the proposed scheme. Simulations results showed that the proposed RR-SJIDF STAP scheme converged at a very fast speed and provided a considerable SINR improvement, outperforming existing state-of-the-art reduced-rank schemes.

## References

[1] L. E. Brennan and I. S. Reed, "Theory of adaptive radar", *IEEE Trans. Aero. Elec. Syst.*, vol. AES-9, no. 2, pp. 237–252, 1973.
[2] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays", *IEEE Trans. Aero. Elec. Syst.*, vol. AES-10, no. 6, pp. 853–863, 1974.
[3] E. J. Kelly, "An adaptive detection algorithm", *IEEE Trans. Aero. Elec. Syst.*, vol. AES-22, no. 2, pp. 115–127, 1986.
[4] A. M. Haimovich and Y. Bar-Ness, "An eigenanalysis interference canceler", *IEEE Trans. Sig. Process.*, vol. 39, no. 1, pp. 76–84, 1991.
[5] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector", *IEEE Trans. Aero. Elec. Syst.*, vol. 28, no. 1, pp. 208–216, Jan 1992.
[6] J. Ward, "Space-time adaptive processing for airborne radar,", *Tech. Rep. 1015, MIT Lincoln lab., Lexington, MA*, Dec. 1994.
[7] A. Haimovich, "The eigencanceler: adaptive radar by eigenanalysis methods", *IEEE Trans. Aero. Elec. Syst.*, vol. 32, no. 2, pp. 532–542, 1996.
[8] J. S. Goldstein and I. S. Reed, "Reduced-rank adaptive filtering", *IEEE Trans. Sig. Process.*, vol. 45, no. 2, pp. 492–496, 1997.
[9] J. S. Goldstein and I. S. Reed, "Theory of partially adaptive radar", *IEEE Trans. Aero. Elec. Syst.*, vol. 33, no. 4, pp. 1309–1325, 1997.
[10] Y.-L. Gau and I.S. Reed, "An improved reduced-rank CFAR space-time adaptive radar detection algorithm", *IEEE Trans. Sig. Process.*, vol. 46, no. 8, pp. 2139–2146, Aug 1998.
[11] I. S. Reed, Y. L. Gau, and T. K. Truong, "CFAR detection and estimation for STAP radar", *IEEE Trans. Aero. Elec. Syst.*, vol. 34, no. 3, pp. 722–735, 1998.
[12] J. S. Goldstein, I. S. Reed, and P. A. Zulch, "Multistage partially adaptive STAP CFAR detection algorithm", *IEEE Trans. Aero. Elec. Syst.*, vol. 35, no. 2, pp. 645–661, 1999.
[13] J. R. Guerci, J. S. Goldstein, and I. S. Reed, "Optimal and adaptive reduced-rank STAP", *IEEE Trans. Aero. Elec. Syst.*, vol. 36, no. 2, pp. 647–663, 2000.
[14] R. Klemm, *Principle of space-time adaptive processing*, IEE Press, Bodmin, UK, 2002.
[15] W. L. Melvin, "A STAP overview", *IEEE Aero. .Elec. Syst. Mag.*, vol. 19, no. 1, pp. 19–35, 2004.
[16] S. Haykin, *Adaptive Filter Theory*, NJ: Prentice-Hall, 4th, ed2002.
[17] J. S. Goldstein and I. S. Reed, "Subspace selection for partially adaptive sensor array processing", *IEEE Trans. Aero. Elec. Syst.*, vol. 33, no. 2, pp. 539–544, 1997.
[18] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the wiener filter based on orthogonal projections", *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2943–2959, 1998.
[19] D. A. Pados and S. N. Batalama, "Joint space-time auxiliary-vector filtering for DS/CDMA systems with antenna arrays", *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1406–1415, 1999.
[20] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter", *IEEE Trans. Sig. Process.]*, vol. 49, no. 2, pp. 290–300, Feb 2001.
[21] D. A. Pados, G. N. Karystinos, S. N. Batalama, and J. D. Matyjas, "Short-data-record adaptive detection", *2007 IEEE Radar Conf.*, pp. 357–361, 17-20 April 2007.

[22] H. Wang, and L. Cai, "On adaptive spatial-temporal processing for airborne surveillance radar systems", *IEEE Trans. Aero. Elec. Syst.*, vol. 30, no. 3, 660670, 1994.

[23] R. S. Adve, T. B. Hale, and M. C. Wicks, "Practical joint domain localised adaptive processing in homogeneous and nonhomogeneous environments. Part 1: Homogeneous environments.", *IEE Proceedings Radar, Sonar and Navigation*, vol. 147, no. 2, 5765, 2000.

[24] R. S. Adve, T. B. Hale, and M. C. Wicks, "Practical joint domain localised adaptive processing in homogeneous and nonhomogeneous environments. Part 2: Nonhomogeneous environments.", *IEE Proceedings Radar, Sonar and Navigation*, vol. 147, no. 2, 6674, 2000.

[25] R. C. de Lamare and R. Sampaio-Neto, "Reduced-rank adaptive filtering based on joint iterative optimization of adaptive filters", *IEEE Sig. Proc. Lett.*, vol. 14, no. 12, pp. 980–983, 2007.

[26] R. Fa, R. C. de Lamare, and D. Zanatta-Filho, "Reduced-rank STAP algorithm for adaptive radar based on joint iterative optimization of adaptive filters", in *Conf. Record of the Fourty-Second Asilomar Conf. Sig. Syst. Comp.*, 2008.

[27] R. C. de Lamare and R. Sampaio-Neto, "Adaptive reduced-rank mmse parameter estimation based on an adaptive diversity-combined decimation and interpolation scheme", in *Proc. IEEE Int. Conf. Acous. Speech Sig. Process.*, 15–20 April 2007, vol. 3, pp. III–1317–III–1320.

[28] R. C. de Lamare, and R. Sampaio-Neto, "Adaptive reduced-rank processing based on joint and iterative interpolation, decimation, and filtering", *IEEE Trans. Sig. Process.* , vol.57, no.7, pp.2503-2514, July 2009

[29] S. Applebaum and D. Chapman, "Adaptive arrays with main beam constraints", *IEEE Trans. on Ant. Prop.*, vol. 24, no. 5, pp. 650–662, 1976.

[30] G. H. Golub and C. F. van Loan, *Matrix Computations*, Wiley, 2002.

[31] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering", *IEEE Trans. Sig. Process.*, vol. 44, no. 5, pp. 1168–1174, 1996.

[32] Jr. Apolinario, J. A., M. L. R. De Campos, and C. P. Bernal O, "The constrained conjugate gradient algorithm", *Signal Processing Letters*,vol. 7, no. 12, pp. 351–354, 2000.

[33] P. S. Chang and Jr. A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering", *IEEE Trans. Sig. Process.*, vol. 48, no. 2, pp. 409–418, Feb. 2000.

[34] M. E. Weippert, J. D. Hiemstra, J. S. Goldstein, and M. D. Zoltowski, "Insights from the relationship between the multistage wiener filter and the method of conjugate gradients", in *Proc. Sensor Array and Multichannel Signal Processing Workshop*, 4–6 Aug. 2002, pp. 388–392.

[35] L. L. Scharf, E. K. P. Chong, M. D. Zoltowski, J. S. Goldstein, and I. S. Reed, "Subspace expansion and the equivalence of conjugate direction and multistage wiener filters", *IEEE Trans. Sig. Process.*, vol. 56, no. 10, pp. 5013–5019, Oct. 2008.

[36] L. Wang and R. C. de Lamare, "Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming," Submitted to *IET Signal Processing* .

[37] H. L. Van Trees, *Optimum Array Processing*, Wiley, New York, 2002.