IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 5, 2019, accepted July 24, 2019, date of publication August 2, 2019, date of current version August 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933139

Reduced-Dimension Wiener Filter Based on Perpendicular Partition

TUANNING LIU, YUANPING ZHOU, AND RONGZHEN MIAO
College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China

Corresponding author: Tuanning Liu (liutuan123@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61831004.

ABSTRACT This paper addresses the reduced-dimension Wiener filter based on perpendicular partition. Two algorithms, i.e., cyclic per-weight (CPW) and batch CPW (BCPW) are presented, which can efficiently perform reduced-dimension adaptive beamforming. In particular, the CPW is suitable for the slow changing environments that have large snapshots to train the weight vector. The BCPW is designed for fast changing scenarios that have low training sequences and sometimes the number of snapshots is less than the number of sensors. In these two algorithms, the solutions of the weight vectors are circularly solved one by one, which only refer to scalar optimization problem then the matrix inversion is avoided. There is no intermediate transformation or orthogonal/non-orthogonal decomposition requirement in our algorithms, so their implementations are relatively simple. Convergence analysis and the computational complexities comparison are provided. Simulation results show the superiorities of the proposed algorithms. Specifically, the CPW has better convergence properties than previous schemes, and the BCPW has low computational complexity and good output SINR performance in low snapshots scenarios.

INDEX TERMS MMSE, Wiener filter, adaptive beamforming, cyclical optimization, perpendicular partition.

I. INTRODUCTION

The array beamforming technique has been extensively used in wireless communications, radar, sonar, microphone array, and so on [1]–[5]. There are numerous techniques to implement the adaptive beamforming [6], [7] such as the sample matrix inversion (SMI) [8], the least mean square (LMS) [4], [9], [10] and the recursive least squares (RLS) methods [11]–[13]. Specifically, the SMI provides a complete optimal solution, but it requires the matrix inversion operation which is computationally intense [14]. The LMS presents a simple solution of filtering, but it often suffers a slow convergence rate and instability. Normalized LMS (NLMS) algorithm employs the variable step sizes to improve the convergence speed, but it still depends on the constant step size which always degrades the performance of beamformer. The work of [4] developed another efficient variable step size mechanism, i.e., the shrinkage linear LMS (SL-LMS) algorithm. It exploits the relationship between the posteriori and priori error signals, so it obviously enhances the convergence rate and decreases the miss-adjustment. The RLS employs the SMW (Sherman Morrison-Woodbury) theorem to avoid the matrix inversion and solve the optimal weight vector by sliding window method [6], [15]. Compared to the LMS, the RLS has faster convergence rate and higher computation cost. In regard to the forgetting factor of the RLS, two recent works [11] and [12] proposed automatically adjusting forgetting factor scheme, which achieves a better tracking performance.

However, for massive arrays, all the above mentioned schemes always need a large number of snapshots to reach convergence state, which leads to performance deterioration especially in dynamic scenarios. Many Reduced-rank or reduced-dimension techniques have been proposed to enhance the convergence rate and reduce the computational complexity. The methods are really suitable for the scenarios with large arrays or low snapshots [16]. Conventional well-known reduced-rank methods are the principal components (PC) [17], [18] and the cross spectral (CS) [19], [20] methods, which require eigenvalue decomposition that always leads to a large computational burden. Besides, another adaptive reduced-rank approach is joint iterative optimization (JIO) scheme [5], [21]–[27], which utilizes the iterative adaptive method to obtain a low complexity and fast tracking adaptive beamformer. But the key of the method is how to get an optimal convergence factor.

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

In the past two decades, the family of Krylov subspace reduced-rank approaches have been studied intensively in the adaptive beamforming area. It mainly includes the auxiliary vector filtering (AVF) [28], the multistage Wiener filter (MSWF) [29]–[36] and the conjugate gradient [37], [38] algorithms. Among them, the MSWF usually has the lowest computational complexity for a similar MSE [39]. In particular, the MSWF utilizes the orthogonal decomposition to provide an equivalent realization for Wiener filter. According to the blocking matrix of MSWF, there are three main algorithms, i.e., the Goldstein Reed Scharf MSWF (GRS-MSWF) [29], the correlation subtractive structure MSWF (CSS-MSWF) [30] and the Household MSWF (HMSWF) [31] algorithms. The sizes of the blocking matrix in the GRS-MSWF and the HMSWF are decreased stage by stage, so they have a low computational load. However, the size of the blocking matrix of the CSS-MSWF remains invariable. Moreover, unlike the CSS-MSWF and the HMSWF, the GRS-MSWF needs to compute the blocking matrix in every stage. The HMSWF outperforms the CSS-HMSWF and the GRS-MSWF in numerical stability and computational efficiency. In addition, the work in [31] proposed an improved CSS-MSWF algorithm, entitled MSWF-CSSI, which employs the normalized blocking matrix to ensure the unit norm of cross-correlation vector. The key issue of the MSWF, as well as all rank-reduced approaches, is to determine the required number of rank accurately and efficiently. A new information (NI) rank selection method is proposed in [30], which has lower complexity and better performance than other comparable rank selection methods.

There are numerous beamforming optimization criteria in literatures, such as the minimum MSE (MMSE), the maximum signal-to-interference-plus-noise-ratio (Max-SINR), the minimum variance (MV) [40]–[43] and the constrained constant modulus (CCM) [12], [22], [25], [28], [44], [45], just to name a few. The MMSE criterion is one of the most promising date-dependent criteria because it doesn't require the direction of arrival (DOA) information. It relies on a desired signal to obtain the optimal beamformer and it can deal with multipath signals effectively.

The purpose of this paper is to address the issue of the reduced-dimension achievement for the Wiener filter. We design a perpendicular partition Wiener filter and propose two algorithms, i.e., cyclic per-weight (CPW) and batch CPW (BCPW). An array beamforming vector is divided into a number of single weights, which is called perpendicular partition. Through minimizing the mean square error (MSE) between the desired signal and the output signal, the update formula is obtained. Then the per-weight is optimized circularly and the optimal weight vector can be obtained. By theoretical analysis, we prove that the proposed algorithms converge to the optimal solution.

The proposed algorithms and their convergence proofs are different from the other existing reduced-dimension approaches, in our limited knowledge. There is no intermediate transformation or orthogonal/non-orthogonal decomposition or matrix inversion requirement in our methods. The comparison of computational complexities indicates that our algorithms have lower complexities than the RLS, the HMSWF-NI and the MSWF-CSSI-NI. In addition, simulation experiments show the CPW has fastest convergence speed in comparison with the SL-LMS and the RLS. And the BCPW has better output SINR performance than the HMSWF-NI and the MSWF-CSSI-NI in low snapshots situations.

The main contributions of this paper are as follows:

1) We design a perpendicular partition Wiener filter structure, which can efficiently achieve reduced-dimension adaptive beamforming.
2) The CPW is proposed to realize adaptive beamforming for large snapshots scenarios. It has faster convergence rate than the SL-LMS and the RLS, and its computational complexity is lower than that of the RLS.
3) Another method, i.e., the BCPW, is proposed to improve tracking performance in quickly changing scenarios. Compared with the HMSWF-NI and the MSWF-CSSI-NI, the BCPW enjoys not only low computational complexity but easier implementation. Simulation results show the BCPW has good output SINR performance in low snapshots situations.

The rest of this paper is arranged as follows. Section II introduces the system model, the Wiener filter and the HMSWF-NI briefly. Section III proposes the CPW and the BCPW, and analyzes their computational complexities. Section IV establishes the convergence of the proposed algorithms. Then Section V is devoted to the computer simulations, followed by the conclusion of this paper in Section VI.

Notations: $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^H$ indicate complex conjugate, vector or matrix transpose and Hermitian operation, respectively. Vectors are indicated by lowercase boldface letters and matrices are represented by uppercase boldface letters. $\mathbb{E}[\cdot]$, $\|\cdot\|$ and $|\cdot|$ denote the statistical expectation, the 2-norm of vectors and the absolute value, respectively.

II. PRELIMINARIES

A. SYSTEM MODEL

Suppose there is a uniform linear array (ULA), which is composed of M omnidirectional antennas with an inter-element space of $\lambda/2$ (λ is wavelength). It's worth noting that the ULA is selected just for illustration purposes and the scheme is also applicable to other array configurations. One desired signal and L interference signals are incident on this array. We assume that they are all far-field narrowband signals and uncorrelated with each other. The steering vectors of the signals are

$$\mathbf{a}(\theta_i) = \left[1, \mathrm{e}^{-j\pi \sin\theta_i}, \cdots, \mathrm{e}^{-j\pi(M-1)\sin\theta_i}\right]^T \in \mathbb{C}^M, \quad i = 0, 1, \ldots, L, \qquad (1)$$

where θ_0 is DOA of the desired signal and θ_i, $i = 1, 2, \ldots, L$, are DOAs of L interference signals respectively.

Assuming that the observed signal is $\mathbf{x}(k) \in \mathbb{C}^M$ at the kth time instant, we have

$$\mathbf{x}(k) = \mathbf{a}(\theta_0)d(k) + \sum_{i=1}^{L} \mathbf{a}(\theta_i)s_i(k) + \mathbf{n}(k), \tag{2}$$

where the desired signal is $d(k)$ and $s_i(k)$, $i = 1, 2, \ldots, L$, are L interferers; $\mathbf{n}(k) \in \mathbb{C}^M$ is the additive white Gaussian noise. The array beamforming vector is defined as $\mathbf{w} \in \mathbb{C}^M$. Then the array beamformer output signal at the kth time instant is

$$y(k) = \mathbf{w}^H \mathbf{x}(k). \tag{3}$$

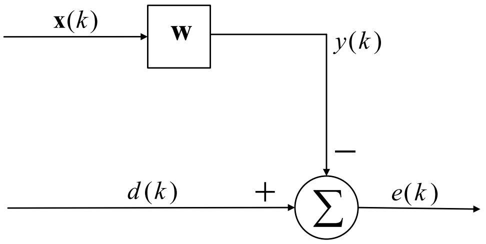

FIGURE 1. Wiener filter.

B. WIENER FILTER

The classical Wiener filter is shown in Fig. 1. Denote $e(k)$ as the error signal at the kth time instant of the filter, i.e.,

$$e(k) = d(k) - \mathbf{w}^H \mathbf{x}(k). \tag{4}$$

The MMSE beamformer is described as

$$\min_{\mathbf{w}}\{\mathbb{E}[|e(k)|^2]\} = \min_{\mathbf{w}}\{\mathbb{E}[|d(k) - \mathbf{w}^H \mathbf{x}(k)|^2]\}. \tag{5}$$

The solution of (5) refers to the well-known Wiener-Hopf equation [29], which is given by

$$\mathbf{Rw}{=}\mathbf{p}, \tag{6}$$

where $\mathbf{R} = \mathbb{E}[\mathbf{x}(k)\mathbf{x}^H(k)]$ is the auto-correlation matrix of $\mathbf{x}(k)$, and $\mathbf{p} = \mathbb{E}[\mathbf{x}(k)d^*(k)]$ is the cross-correlation vector between $\mathbf{x}(k)$ and $d(k)$. The noted Wiener solution $\mathbf{w}_{\text{MMSE}}$ is obtained as

$$\mathbf{w}_{\text{MMSE}} = \mathbf{R}^{-1}\mathbf{p}. \tag{7}$$

In the SMI [46], $\mathbf{R}$ and $\mathbf{p}$ are estimated by time average of K samples, that is,

$$\hat{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}(k)\mathbf{x}^H(k) = \frac{1}{K}\mathbf{X}\mathbf{X}^H \tag{8}$$

and

$$\hat{\mathbf{p}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}(k)d^*(k) = \frac{1}{K}\mathbf{X}\mathbf{d}^H, \tag{9}$$

where $\mathbf{X} \in \mathbb{C}^{M\times K}$ and $\mathbf{d} \in \mathbb{C}^K$ are obtained by stacking K time instant samples of $\mathbf{x}(k)$ and $d(k)$, i.e., $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(K)]$ and $\mathbf{d} = [d(1), d(2), \cdots, d(K)]^T$ respectively. Although the SMI is insensitive to eigenvalue spread of $\hat{\mathbf{R}}$, an ill-conditioned covariance matrix deteriorates the performance of the SMI. Diagonal load (DL) [47] technique is a widespread approach to enhance the robustness of the algorithm. A scaled identity matrix is added to $\hat{\mathbf{R}}$, in the DL method, which relieves the effects of mismatch errors, random perturbations, small sample support and so on. Unfortunately, there is a trade-off between the accuracy of the inversion and the allowable eigenvalue spread of $\hat{\mathbf{R}}$ for the SMI.

C. HMSWF-NI ALGORITHM

The MSWF provides a multistage orthogonal decomposition method for implementing the Wiener filter, which avoids matrix inversion and eigenvalue decomposition [29]. The main idea of the MSWF is maximizing the correlation between this projection output vector and previous stages output by choosing the additional projection vector. In detail, the MSWF decomposes the observed signal into two subspaces at each stage. One subspace is in the direction of the cross-correlation vector and another is orthogonal to this direction. The blocking matrix affects the computational complexity and the stability of the algorithm. Now the best performing method is the HMSWF algorithm [31] because Householder transformation ensures unitary blocking. The NI rank selection scheme not only has a small computation burden but also enjoys high selected-rank accuracy [30]. So the HMSWF based on the NI rank selection scheme, referred to as HMSWF-NI, is briefly described in this section.

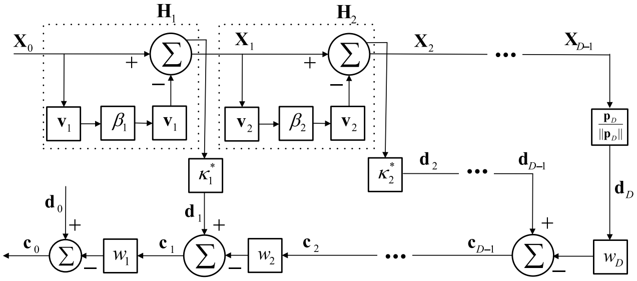

FIGURE 2. Householder multistage Wiener filter.

The diagram of the HMSWF is shown in Fig. 2, where $\mathbf{d}_i \in \mathbb{C}^K$ and $\mathbf{c}_i \in \mathbb{C}^K$ are the desired signal vector and the error vector at the ith stage by stacking K samples respectively. The HMSWF utilizes the Householder transform matrix $\mathbf{H}_i \in \mathbb{C}^{(M-i+1)\times(M-i+1)}$ to decompose the observed signal of ith stage $\mathbf{X}_i \in \mathbb{C}^{(M-i)\times K}$ into two co-orthogonal sections.

In forwards recursion of the HMSWF-NI, with Householder transform [15], [31], the following computations are carried out,

$$\mathbf{H}_i = \mathbf{I} - \beta_i \mathbf{b}_i \mathbf{b}_i^H,\ \mathbf{b}_i = \mathbf{p}_i - \kappa_i||\mathbf{p}_i||\mathbf{u}_i, \quad i = 1, 2, \ldots, D, \tag{10}$$

where $\mathbf{b}_i \in \mathbb{C}^{(M-i+1)}$ is the Household vector and D is the appropriate rank of the HMSWF determined by the NI scheme; $\mathbf{p}_i \in \mathbb{C}^{(M-i+1)}$ is calculated as $\hat{\mathbf{p}}_i = \frac{\mathbf{X}_{i-1}\mathbf{d}_{i-1}^H}{K}$; $\mathbf{u}_i = [1, 0, \cdots, 0]^T \in \mathbb{C}^{(M-i+1)}$; β_i and κ_i are given by

$$\kappa_i = \frac{\hat{p}_{i,1}}{|\hat{p}_{i,1}|}, \ \beta_i = \frac{-1}{\kappa_i^* ||\hat{\mathbf{p}}_i|| b_{i,1}}, \tag{11}$$

where $\hat{p}_{i,1}$ is the first element of $\hat{\mathbf{p}}_i$ and $b_{i,1}$ is the first element of $\mathbf{b}_i$.

In backwards recursion of the HMSWF-NI, only a scalar weight needs to be solved at each stage, i.e.,

$$w_i = \frac{||\hat{\mathbf{p}}_i||}{\zeta_i}, \quad \mathbf{c}_{i-1} = \mathbf{d}_{i-1} - w_i^* \mathbf{c}_i, \tag{12}$$

where $\zeta_i = \frac{\mathbf{c}_i^H \mathbf{c}_i}{K}$.

In the HMSWF, with the increasing of the dimensions of the subspace, the information between the desired signal and the observed signal is extracted. The main idea of the NI rank-selected criterion is that using the amount of this information to determine the appropriate rank. In other words, the proper rank D is measured by the level of the information between $\mathbf{x}(k)$ and $d(k)$. Compared to other rank-selected criteria, the NI scheme has less computational complexity and higher rank selection accuracy [30].

The level of the information between $\mathbf{x}(k)$ and $d(k)$ is denoted as ρ_i,

$$\rho_i = \sigma_{\mathbf{d}_i}^2 - ||\hat{\mathbf{p}}_i||^2 / \rho_{i-1}, \tag{13}$$

where $\sigma_{\mathbf{d}_i}^2 = \mathbb{E}[|d_i(k)|^2]$ is the variance of $d_i(k)$. The NI scheme is performed as

$$D = \max_i \{i | \rho_i > \eta\}, \tag{14}$$

where η is the threshold.

The detailed steps of the HMSWF-NI are summarized in **HMSWF-NI Algorithm**.

The HMSWF transforms the matrix inversion to a sequence of scalar inversion computations, which alleviates the computational burden. However, the orthogonal decomposition required by the algorithm involves the extra computational cost and implementation difficulties.

III. THE PROPOSED ALGORITHMS

In this section, we devise a perpendicular partition Wiener filter, and then we present two reduced-dimension adaptive beamforming algorithms, i.e., the CPW and the BCPW. The former is suitable for slow changing environments that have large snapshots to train weight vector; the latter is designed for fast changing scenarios which have low training sequences and sometimes the number of snapshots is less than the number of sensors.

In the beginning, an array beamforming vector is divided into a number of scalars, which is called perpendicular partition depicted in Fig. 3. Then we directly minimize the MSE between the desired signal and the output signal. The weight vector optimization problem is turned into per weight optimization problem, thus the matrix inversion is replaced by scalar reciprocal. The weight vector is circularly updated one by one, and it converges to the optimal solution finally.

HMSWF-NI Algorithm

```
Initialization: $\mathbf{d}_0 = \mathbf{d}$, $\mathbf{X}_0 = \mathbf{X}$.
Forward recursion:
for $i = 1, 2, \ldots, M-1$
    $\hat{\mathbf{p}}_i = \frac{\mathbf{X}_{i-1}\mathbf{d}_{i-1}^H}{K}$, $||\hat{\mathbf{p}}_i|| = \sqrt{\hat{\mathbf{p}}_i^H \hat{\mathbf{p}}_i}$, $\kappa_i = \frac{\hat{p}_{i,1}}{|\hat{p}_{i,1}|}$,
    $\mathbf{b}_i = \hat{\mathbf{p}}_i - \kappa_i ||\hat{\mathbf{p}}_i|| \mathbf{u}_i$, $\beta_i = \frac{-1}{\kappa_i^* ||\hat{\mathbf{p}}_i|| b_{i,1}}$,
    $\begin{bmatrix} \kappa_i \mathbf{d}_i^T \\ \mathbf{X}_i \end{bmatrix} = \mathbf{X}_{i-1} - \beta_i \mathbf{b}_i \mathbf{b}_i^H \mathbf{X}_{i-1}$.
    if $i = 1$
        $\rho_i = \sigma_{\mathbf{d}_1}^2$.
    else
        $\rho_i = \sigma_{\mathbf{d}_i}^2 - ||\hat{\mathbf{p}}_i||^2 / \rho_{i-1}$.
    end if
    if $\rho_i < \eta$
        $D = i - 1$,
        break.
    end if
end for
Backward recursion:
for $i = D, D-1, \ldots, 1$
    $\zeta_i = \frac{\mathbf{c}_i^H \mathbf{c}_i}{K}$, $w_i = \frac{||\hat{\mathbf{p}}_i||}{\zeta_i}$,
    $\mathbf{c}_{i-1} = \mathbf{d}_{i-1} - w_i^* \mathbf{c}_i$.
end for
```

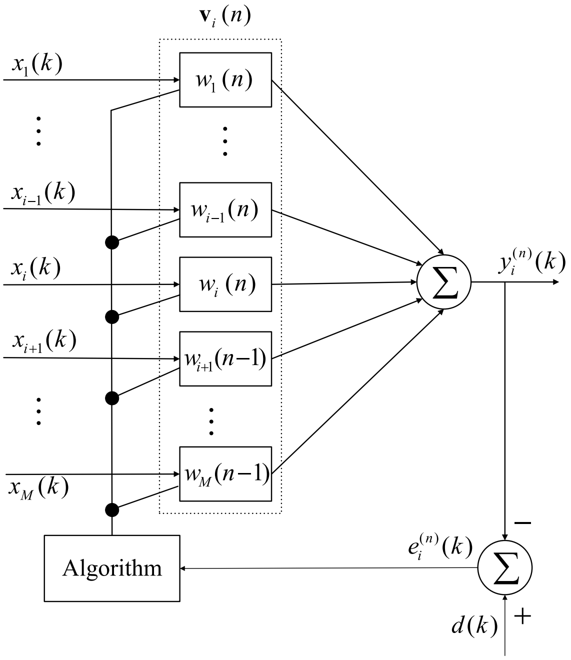

FIGURE 3. Perpendicular partition Wiener filter.

A. THE CPW ALGORITHM

The diagram of the perpendicular partition Wiener filter is shown in Fig. 3. The array beamforming vector is denoted as

$$\mathbf{w}(n) = [w_1(n), w_2(n), \cdots, w_M(n)]^T, \tag{15}$$

where n indicates the times of updates. Correspondingly,

$$\mathbf{x}_i(k) = [x_1(k), x_2(k), \cdots, x_M(k)]^T \tag{16}$$

is the observed signal of the kth time instant.

In order to express the case of each weight update, $\mathbf{v}_i(n) \in \mathbb{C}^M$ is defined as a vector indicating $\mathbf{w}$ at updating process. Specifically, $\mathbf{v}_i(n)$ denotes that the 1st-ith elements of $\mathbf{w}$ have been updated n times and the $(i+1)$th-Mth elements have been updated $n-1$ times, i.e.,

$$\mathbf{v}_i(n) = [w_1(n), \cdots, w_i(n), w_{i+1}(n-1), \cdots, w_M(n-1)]^T. \tag{17}$$

Specially, $\mathbf{v}_0(n)$ and $\mathbf{v}_M(n)$ are the beginning and the end of the nth update of $\mathbf{w}$ respectively; they can be expressed as

$$\mathbf{v}_0(n) = \mathbf{w}(n-1) \quad \text{and } \mathbf{v}_M(n) = \mathbf{w}(n). \tag{18}$$

Similarly, the output signal $y_i^{(n)}(k)$ and the error $e_i^{(n)}(k)$ in the system are respectively given by

$$y_i^{(n)}(k) = \mathbf{v}_i^H(n)\mathbf{x}(k) \text{ and } e_i^{(n)}(k) = d(k) - \mathbf{v}_i^H(n)\mathbf{x}(k). \tag{19}$$

In accordance with (18) and (19), we have

$$e_0^{(n)}(k) = e^{(n-1)}(k), \quad e_M^{(n)}(k) = e^{(n)}(k). \tag{20}$$

Then the optimization problem in (5) is replaced as

$$\min_{\mathbf{w}} \mathbb{E}[|e_i^{(n)}(k)|^2] = \min_{\mathbf{v}_i(n)} \mathbb{E}[|d(k) - \mathbf{v}_i^H(n)\mathbf{x}(k)|^2]. \tag{21}$$

Define $\Delta w_i(n)$ as an increment of w_i in the nth update, i.e.,

$$w_i(n) = w_i(n-1) + \Delta w_i(n). \tag{22}$$

Further, we have

$$\mathbf{v}_i^H(n)\mathbf{x}(k) = \mathbf{v}_{i-1}^H(n)\mathbf{x}(k) + \Delta w_i^*(n)x_i(k). \tag{23}$$

Therefore, according to (23), we can rewrite Eq. (21) as

$$\begin{aligned}&\min_{\mathbf{v}_i(n)} \mathbb{E}[|d(k) - \mathbf{v}_i^H(n)\mathbf{x}(k)|^2] \\ &\quad = \min_{\Delta w_i(n)} \mathbb{E}[|d(k) - \mathbf{v}_{i-1}^H(n)\mathbf{x}(k) - \Delta w_i^*(n)x_i(k)|^2]. \end{aligned} \tag{24}$$

Using the Lagrange multiplier method, the objective function is given by

$$\mathcal{L}(\Delta w_i(n)) = \mathbb{E}[|d(k) - \mathbf{v}_{i-1}^H(n)\mathbf{x}(k) - \Delta w_i^*(n)x_i(k)|^2]. \tag{25}$$

The gradient of $\mathcal{L}$ with respect to $\Delta w_i^*(n)$ is given by

$$\begin{aligned}&\mathcal{L}'_{\Delta w_i^*(n)} \\ &= \mathbb{E}\{x_i(k)[x_i(k)\Delta w_i^*(n) + \mathbf{v}_{i-1}^H(n)\mathbf{x}(k) - d(k)]^*\} \\ &= \mathbb{E}[|x_i(k)|^2]\Delta w_i(n) + \mathbb{E}\{x_i(k)[\mathbf{v}_{i-1}^H(n)\mathbf{x}(k) - d(k)]^*\}. \end{aligned} \tag{26}$$

Set $\mathcal{L}'_{\Delta w_i^*(n)} = 0$, we obtain

$$\begin{aligned}&\Delta w_i(n) \\ &= \{\mathbb{E}[|x_i(k)|^2\}^{-1}\{\mathbb{E}[x_i(k)d^*(k)] - \mathbb{E}[x_i(k)\mathbf{x}^H(k)\mathbf{v}_{i-1}(n)]\} \\ &= r_{ii}^{-1}[p_i - \mathbf{r}_i\mathbf{v}_{i-1}(n)], \quad i = 1, 2, \ldots, M, \end{aligned} \tag{27}$$

where $r_{ii} = \mathbb{E}[|x_i(k)|^2]$ is the variance of $\mathbf{x}_i(k)$; $p_i = \mathbb{E}[x_i(k)d^*(k)]$ is the cross-correlation value between $x_i(k)$ and $d(k)$; $\mathbf{r}_i = \mathbb{E}[x_i(k)\mathbf{x}^H(k)]$ is the cross-correlation vector between $x_i(k)$ and $\mathbf{x}(k)$. The update formula of the CPW is given by

$$w_i(n) = w_i(n-1) + \hat{r}_{ii}^{-1}[\hat{p}_i - \hat{\mathbf{r}}_i\mathbf{v}_{i-1}(n)], \quad i = 1, 2, \ldots, M. \tag{28}$$

The averages of K samples are applied to estimate r_{ii}, p_i and $\mathbf{r}_i$ for $i = 1, 2, \ldots, M$,

$$\hat{r}_{ii} = \frac{1}{K}\sum_{k=1}^{K} |x_i(k)|^2, \tag{29}$$

$$\hat{p}_i = \frac{1}{K}\sum_{k=1}^{K} x_i(k)d^*(k), \tag{30}$$

$$\hat{\mathbf{r}}_i = \frac{1}{K}\sum_{k=1}^{K} x_i(k)\mathbf{x}^H(k). \tag{31}$$

The detailed procedure of the CPW can be implemented in **Algorithm 1**.

Algorithm 1 Cyclic Per-Weight (CPW)

```
Initialization:
$\mathbf{w}(0) = [w_1(0), w_2(0), \cdots, w_M(0)]^T = \mathbf{v}_0(1) = \mathbf{0}$,
$\mathbf{w}(1) = \mathbf{u}_1 = [1, 0, \cdots, 0]^T \in \mathbb{C}^M$,
$\hat{r}_{ii} = \frac{1}{K}\sum_{k=1}^{K} |x_i(k)|^2$, slove $\hat{r}_{ii}^{-1}$,
$\hat{p}_i = \frac{1}{K}\sum_{k=1}^{K} x_i(k)d^*(k)$, $\hat{\mathbf{r}}_i = \frac{1}{K}\sum_{k=1}^{K} x_i(k)\mathbf{x}^H(k)$,
$for\ i = 1, 2, \ldots, M,\ set\ n = 1.$
Iteration procedure:
while $||\mathbf{w}(n) - \mathbf{w}(n-1)||^2 > \varsigma$ (small positive value) do
  for $i = 1, 2, \ldots, M$ do
    $w_i(n) = w_i(n-1) + \hat{r}_{ii}^{-1}[\hat{p}_i - \hat{\mathbf{r}}_i\mathbf{v}_{i-1}(n)]$,
    $\mathbf{v}_i(n) =$
    $[w_1(n), \cdots, w_i(n), w_{i+1}(n-1), \cdots, w_M(n-1)]$,
  end for
  $\mathbf{w}(n) = \mathbf{v}_M(n)$, $n = n + 1$.
end while
```

It is worth noting that the update formula (28) of the CPW only involves the scalar reciprocal computation, thus, $\mathbf{R}^{-1}$ is replaced with r_{ii}^{-1} in the CPW; r_{ii} is the ith diagonal element of $\mathbf{R}$. The CPW can be regarded as an alternative implementation of the reduced-dimension adaptive beamforming algorithm.

B. THE BCPW ALGORITHM

In the fast changing scenarios, low snapshots usually lead to an ill-conditioning $\hat{\mathbf{R}}$, and the eigenvalue spread can always reach a point where the computer cannot accurately invert the matrix. Especially, when $M > K$, $\hat{\mathbf{R}}$ is rank-deficient, the full-rank algorithms can't work. In the DL scheme, a scaled identity matrix is added to $\hat{\mathbf{R}}$ for compensating eigenvalue spread of $\hat{\mathbf{R}}$. Unfortunately, how to choose the loading level reliably and accurately is still a problem although there are many different schemes for computing the DL level automatically [48]. Our proposed perpendicular partition Wiener filter is a reduced-dimension model of the classical Wiener filter. In the CPW, we only need to invert $\hat{r}_{ii}$, which is the diagonal element of $\hat{\mathbf{R}}$. The processing of transforming $\hat{\mathbf{R}}^{-1}$ to $\hat{r}_{ii}^{-1}$ eliminates the correlations among the eigenvalues of $\hat{\mathbf{R}}$. Therefore, the CPW is suitable for low snapshots support situations. Considering the features of the schemes of rank-reduced, we present the BCPW to effectively achieve adaptive beamforming in low snapshots situations.

Set $\mathcal{L}'_{\Delta w_i^*(n)} = 0$, we can get another form of the solution,

$$\begin{aligned}\Delta w_i(n) &= \{\mathbb{E}[|x_i(k)|^2\}^{-1}\mathbb{E}\{x_i(k)[d(k) - \mathbf{v}_{i-1}^H(n)\mathbf{x}(k)]^*\}\\ &= \frac{\mathbb{E}[x_i(k)e_{i-1}^{(n)*}(k)]}{\mathbb{E}[|x_i(k)|^2]}, \quad i = 1, 2, \ldots, M.\end{aligned} \tag{32}$$

Then the updating formula of the BCPW is given by

$$w_i(n) = w_i(n-1) + \frac{\mathbb{E}[x_i(k)e_{i-1}^{(n)*}(k)]}{\mathbb{E}[|x_i(k)|^2\}}, \quad i = 1, 2, \ldots, M. \tag{33}$$

In the BCPW, we take the average of K snapshots as the estimates of the expectations in (33). Firstly, $\mathbf{X}$ is perpendicularly partitioned as

$$\mathbf{X}{=}[\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_M^T]^T, \tag{34}$$

where $\mathbf{x}_i \in \mathbb{C}^K$ is the ith partition of $\mathbf{X}$ and it is composed of K snapshots, i.e.,

$$\mathbf{x}_i{=}[x_i(1), x_i(2), \cdots, x_i(K)]; \tag{35}$$

and the K errors $e_i^{(n)}(k),\ i = 1, 2, \ldots, K$ are stacked into an error vector $\mathbf{e}_i(n)$.

$$\mathbf{e}_i(n) = [e_i^{(n)}(1), e_i^{(n)}(2), \cdots, e_i^{(n)}(K)]. \tag{36}$$

Finally, (33) is simplified as

$$w_i(n) = w_i(n-1){+}\frac{\mathbf{x}_i\mathbf{e}_{i-1}^T(n)}{\mathbf{x}_i\mathbf{x}_i^H}. \tag{37}$$

The detailed steps of the BCPW are summarized in **Algorithm 2**.

C. COMPLEXITY ANALYSIS

Here, we compare the computational complexities of the proposed algorithms and other existing ones. In Table 1, the computational complexities of these algorithms are evaluated in terms of the number of complex multiplications per symbol. As shown in the table 1, except for the SL-LMS [9], the numbers of complex multiplications of all the methods scale quadratically with M. The LMS type approaches usually enjoy low computational complexities and simplicities, meanwhile, suffer slow convergence rate and instability. The MSWF type approaches need orthogonal decomposition, which leads to extra computational burden. The dimension of each blocking matrix of the MSWF-CSSI-NI equals to M while that is decreasing stage by stage in the HMSWF-NI, so the computational complexity of the MSWF-CSSI-NI is larger than the HMSWF-NI. Our proposed algorithms utilize perpendicular partition to avoid matrix, and they don't need intermediate transformation or orthogonal/non-orthogonal decomposition or matrix inversion. All in all, the computational complexities of the CPW and the BCPW are less than that of the RLS, the MSWF-CSSI-NI [30] and the HMSWF-NI [31].

Fig. 4 depicts the computational complexity for per symbol versus the number of antennas. For comparison purposes, we set $D = 3$ for the HMSWF-NI and the CSSI-MSWF-NI.

Algorithm 2 Batch Cyclic Per-Weight (BCPW)

```
Initialization:
$\mathbf{w}(0) = [w_1(0), w_2(0), \cdots, w_M(0)]^T = \mathbf{v}_0(1) = \mathbf{0}$,
$n = 1,\ \mathbf{w}(1) = \mathbf{u}_1 = [1, 0, \ldots, 0]^T,\ \mathbf{e}_0(1) = \mathbf{d}$.
Iteration procedure:
while $||\mathbf{w}(n) - \mathbf{w}(n-1)||^2 > \varsigma$ (small real value) do
    for $i = 1, 2, \ldots, M$ do
        $w_i(n) = w_i(n-1){+}\frac{\mathbf{x}_i\mathbf{e}_{i-1}^T(n)}{\mathbf{x}_i\mathbf{x}_i^H}$,
        $\mathbf{v}_i(n) =$
        $[w_1(n), \cdots, w_i(n), w_{i+1}(n-1), \cdots, w_M(n-1)]$,
        $\mathbf{e}_i(n) = \mathbf{d} - \mathbf{v}_i^H(n)\mathbf{X}$.
    end for
    $\mathbf{w}(n){=}\mathbf{v}_M(n)$, $\mathbf{e}_0(n+1) = \mathbf{e}_M(n)$, $n = n+1$.
end while
```

TABLE 1. Computational complexity comparison.

Algorithms	Complex multiplications per symbol
CPW	$2M^2 + 4M$
BCPW	$M^2 + 3M$
HMSWF-NI	$4D(M{+}1){+}M^2{+}\frac{M^2D(1+D)}{3}{-}2D^2{+}3$
MSWF-CSSI-NI	$4D(M+1) + M^2 + \frac{M^2D(1+D)}{2} + 2$
RLS	$4M^2 + 2M + 2$
SL-LMS	$3M + 13$

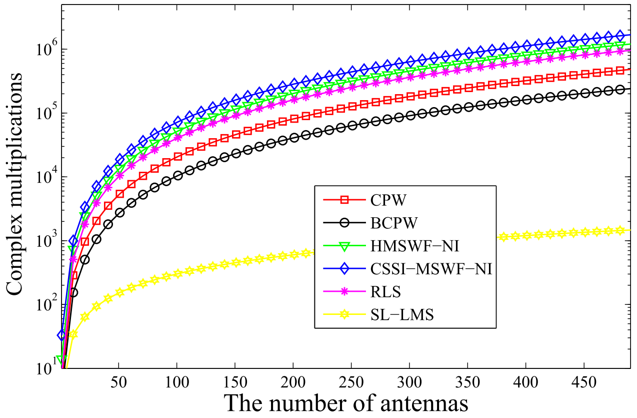

FIGURE 4. Computational complexity comparison among six algorithms: CPW, BCPW, HMSWF-NI, CSSI-MSWF-NI, RLS and SL-LMS.

As can be seen from the figure, the SL-LMS has the lowest computational quantity, followed by the BCPW and the CPW; the CSSI-MSWF-NI has the highest computational quantity. It should be pointed out that the computational complexity of the BCPW is lower than the CPW because of $K = 1$. The computation complexity of the BCPW increases with the number of training symbols; so the BCPW is more suitable for low snapshots scenarios than the CPW.

In summary, the CPW and the BCPW enjoy lower complexities than the RLS, the CSSI-MSWF-NI and the HMSWF-NI. Although the SL-LMS has lowest computational complexity, it usually suffers poor performance.

IV. CONVERGENCE ANALYSIS

The MSE between the desired signal and the output signal is denoted as ε,

$$\varepsilon(n) = \mathbb{E}[|e^{(n)}(k)|^2], \ \varepsilon_i(n) = \mathbb{E}[|e_i^{(n)}(k)|^2]. \tag{38}$$

Then, $\varepsilon_i(n)$ is calculated as

$$\begin{aligned}\varepsilon_i(n) &= \mathbb{E}[|e_i^{(n)}(k)|^2] = \mathbb{E}[|d(k) - \mathbf{v}_i^H(n)\mathbf{x}(k)|^2] \\ &= \mathbb{E}[|d(k) - \mathbf{v}_{i-1}^H(n)\mathbf{x}(k) - \Delta w_i^*(n)x_i(k)|^2] \\ &= \varepsilon_{i-1}(n) + \Delta w_i^*(n)\mathbb{E}[|x_i(k)|^2]\Delta w_i(n) \\ &\quad - 2\Delta w_i^*(n)\mathbb{E}[x_i(k)e_{i-1}^{(n)*}(k)] \\ &= \varepsilon_{i-1}(n) - \Delta w_i^*(n)\mathbb{E}[|x_i(k)|^2]\Delta w_i(n) \\ &= \varepsilon_{i-1}(n) - |\Delta w_i(n)|^2\mathbb{E}[|x_i(k)|^2], \\ &\qquad i = 1, 2, \ldots, M, \ n = 1, 2, \ldots, \end{aligned} \tag{39}$$

and from (32), we have $\mathbb{E}[x_i(k)e_{i-1}^{(n)*}(k)] = \mathbb{E}[|x_i(k)|^2]\Delta w_i(n)$.

Considering $|\Delta w_i(n)|^2\mathbb{E}[|x_i(k)|^2] \geq 0$, in (39), we get $\varepsilon_i(n) \leq \varepsilon_{i-1}(n)$. According to (20) and (38), we have

$$\varepsilon_0(n) = \varepsilon(n-1), \quad \varepsilon_M(n) = \varepsilon(n). \tag{40}$$

So we can obtain relationships are as follows

$$\begin{aligned}\varepsilon_1(n) &\leq \varepsilon_0(n) = \varepsilon(n-1), \\ \varepsilon(n) &= \varepsilon_M(n) \leq \cdots \leq \varepsilon_1(n),\end{aligned} \tag{41}$$

and

$$\begin{aligned}\varepsilon_1(n+1) &\leq \varepsilon_0(n+1) = \varepsilon(n), \\ \varepsilon(n+1) &= \varepsilon_M(n+1) \leq \cdots \leq \varepsilon_1(n+1).\end{aligned} \tag{42}$$

Following (41) and (42), we obtain

$$\begin{aligned}&\varepsilon_{i+1}(n) \leq \varepsilon_i(n) \leq \varepsilon_{i-1}(n), \\ &\varepsilon(n+1) \leq \varepsilon(n) \leq \varepsilon(n-1), \\ &\qquad i = 1, 2, \ldots, M, \ n = 1, 2, \ldots.\end{aligned} \tag{43}$$

So $\varepsilon_i(n)$ is a non-negative monotonically decreasing Cauchy sequence. Therefore the sequence $\varepsilon_i(n)$ converges, which can be expressed as

$$\lim_{n\to\infty} |\varepsilon_i(n) - \varepsilon_i(n-1)| = 0, \quad i = 1, 2, \ldots, M. \tag{44}$$

Considering $\mathbb{E}[|x_i(k)|^2] > 0$, and it is incorporated with (39) and (44) simultaneously we have

$$\lim_{n\to\infty} ||\Delta w_i(n)||^2 = 0, \quad i = 1, 2, \ldots, M. \tag{45}$$

In the end, we have

$$\lim_{n\to\infty} ||\mathbf{w}(n) - \mathbf{w}(n-1)||^2 = 0. \tag{46}$$

Hence, the convergence properties of the CPW and the BCPW are verified. It is worth noting that with the increasing of the number of iterations, the MSEs of proposed algorithms are decreasing monotonically. So the weight vectors of our approaches are steadily converge optimal solution.

V. SIMULATION RESULTS AND ANALYSES

In this section, we study and analyze the performances of the proposed schemes via computer simulations. Here, the CPW is compared with the RLS and the SL-LMS; the BCPW is compared with the HMSWF-NI and the CSSI-MSWF-NI. In these simulations, a uniform line array (ULA) is used with the inter-element spacing of half wavelength and all antennas are omnidirectional. The desired signal is presumed to arrive at $\theta_0 = 0°$, while three interferers are impinge on this array with DOAs of $\theta_1 = -50°$, $\theta_2 = 30°$ and $\theta_3 = 60°$. Supposed that they are all far-field narrowband signals and uncorrelated with each other. The input signal-to-noise-ratio (SNR) is 0dB with spatially and temporally white Gaussian noise, and the interference-to-noise-ratio (INR) of each interferer is fixed at 10dB. Without loss of generality, all numerical results are averaged over 200 independent Monte-Carlo experiments. The performances of the algorithms are assessed by the output SINRs calculated as

$$SINR = \frac{\mathbf{w}^H\mathbf{R}_s\mathbf{w}}{\mathbf{w}^H\mathbf{R}_{i+n}\mathbf{w}}, \tag{47}$$

where $\mathbf{R}_s$ and $\mathbf{R}_{i+n}$ are the covariance matrices of signal and the interference plus noise, respectively.

A. OUTPUT PERFORMANCES IN SLOW CHANGING SCENARIOS

This simulation is conducted to test the performances of the CPW in slow changing scenarios. The number of array elements and snapshots are $M = 16$ and $K = 500$ respectively. In Fig. 5, we compare the CPW with the SL-LMS [9], the RLS and the SMI. In slow changing scenarios, to achieve the best performance for the RLS, the forgetting factor is set to 0.999. The beampatterns, the output SINRs and the MSEs of four algorithms are depicted in Fig. 5. The theoretical bounds for the performances are given by the SMI. It is seen that in Fig. 5(a) the mainlobes of all beam patterns aim to the direction of the desired signal while the nulls are adjusted to the directions of the interferers. As shown in Fig. 5(b) and Fig. 5(c), the output SINR and the MSE of the CPW are similar with that of the SMI. The CPW enjoys fastest convergence and the best steady-state performance among three algorithms (SL-LMS and RLS). It is reasonable that the

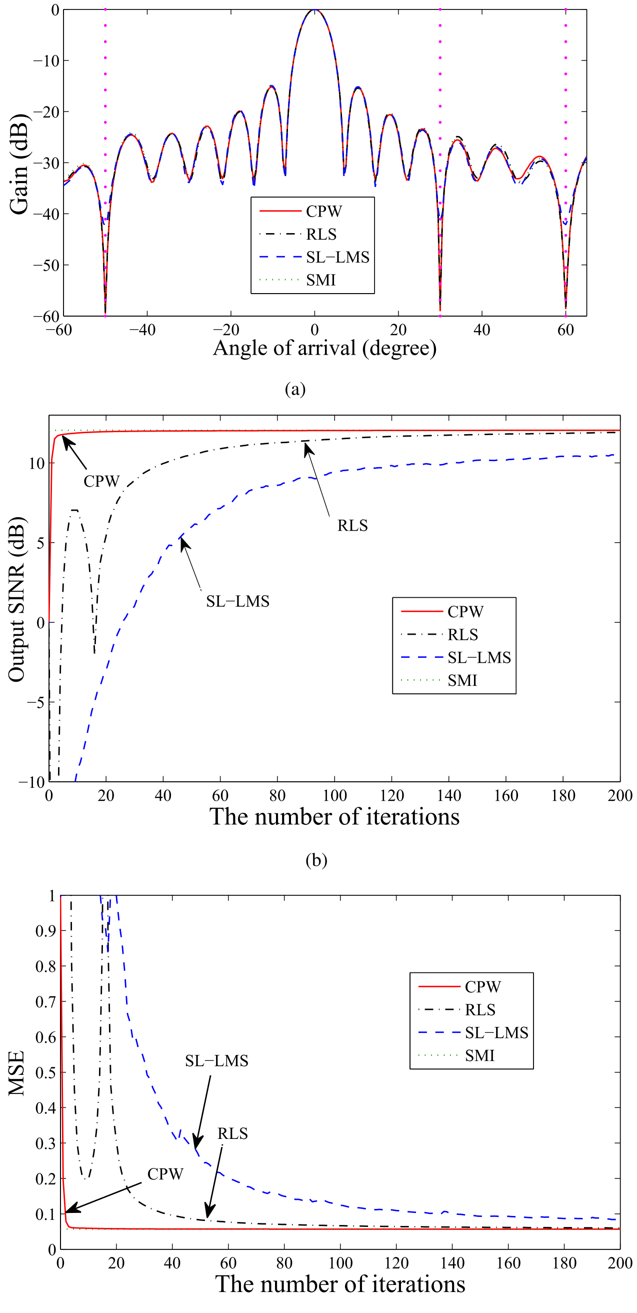

FIGURE 5. The performance comparison of four algorithms (CPW, SL-LMS, RLS and SMI). (a) Beam pattern; (b) Output SINR versus the number of iterations; (c) MSE versus the number of iterations.

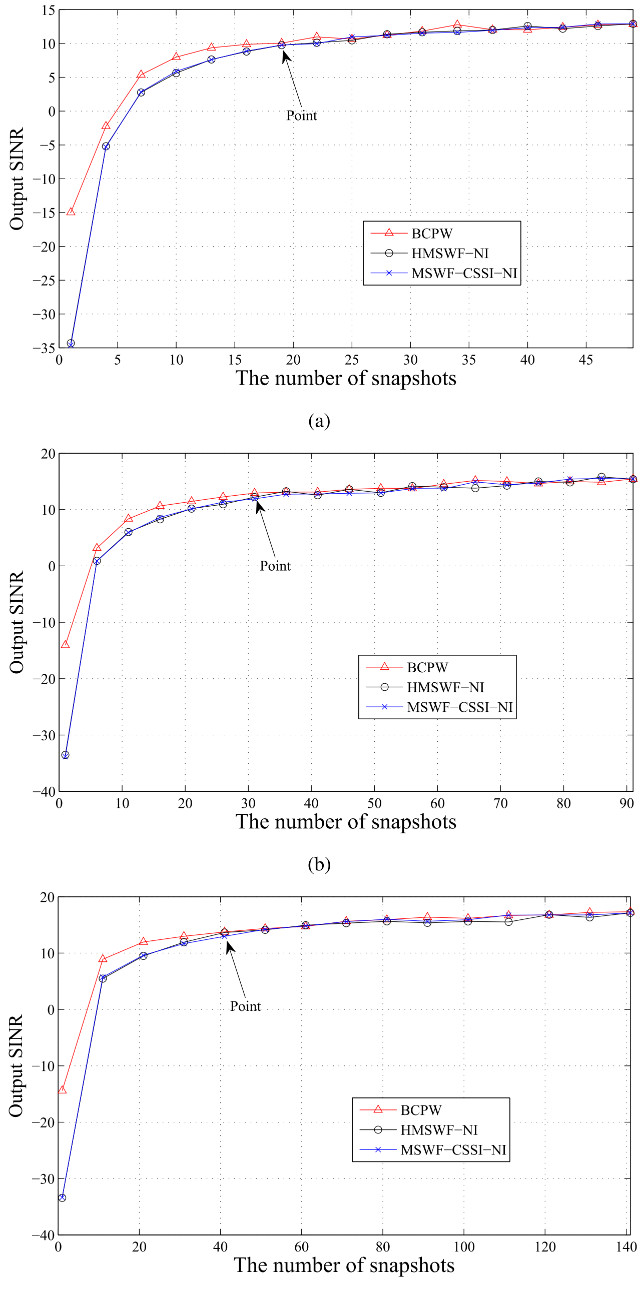

FIGURE 6. The output SINR performance comparison versus the training size of three algorithms (BCPW, HMSWF-NI and MSWF-CSSI-NI); (a) $M = 32$; (b) $M = 64$; (c) $M = 128$.

SL-LMS suffers the slowest convergence rate and the lowest output SINR because the essence of the SL-LMS is stochastic gradient descent. It should be noted that the CPW has faster convergence rate and better output performance than the RLS while the computational complexity of the CPW is actually less than the RLS.

B. OUTPUT SINR VERSUS THE NUMBER OF SNAPSHOTS

The approaches of reduced-rank or the reduced-dimension are designed to enhance the convergence rate and alleviate the computational complexity in large array or fast changing scenarios. This simulation demonstrates the performances of the BCPW in low snapshots and large array situations.

Fig. 6 shows the output SINR performance versus the number of snapshots for the BCPW and existing works including the HMSWF-NI and the MSWF-CSSI-NI [30], [31]. There are three cases of $M = 32$, $M = 64$ and $M = 128$ in Fig. 6(a), Fig. 6(b) and Fig. 6(c), respectively. From Fig. 6, we can see that with the increasing of the number of the snapshots, the output SINRs of three methods converge to the steady state. When the number of snapshots is lower than 19, 31 and 41, the BCPW has highest output SINR among the three algorithms for $M = 32$, $M = 64$ and $M = 128$ respectively. This is because that the low snapshots bring errors of block matrices in the HMSWF-NI and the MSWF-CSSI-NI, and the errors are accumulated stage by stage. Moreover, when the snapshots are larger than the points, the performance of the BCPW is similar to the HMSWF-NI and the MSWF-CSSI-NI. In summary, the BCPW has lower computational complexity and better output SINR performance than the HMSWF-NI and the MSWF-CSSI-NI in low snapshots situations.

VI. CONCLUSION

This paper has conducted the reduced-dimension Wiener filter based on perpendicular partition. Two effective algorithms have been proposed, i.e., the CPW and the BCPW. By cyclically update on the per-weight, the optimal MMSE beamformer can be achieved. Without intermediate transformation or orthogonal/non-orthogonal decomposition or matrix inversion, the implementations of the proposed algorithms are simple and desirable in practical application. The convergence and computational complexities analysis of our algorithms have been investigated. Simulation results demonstrated the advantages of proposed algorithms. In particular, the CPW has faster convergence rate than existing algorithms, and the BCPW has low computational complexity and good output SINR performance in small number of snapshots situations. This paper focused on the case in which the desired signal is assumed to be known exactly. In the following research we will study robust beamforming algorithm in the case of incomplete knowledge of desired signal.

REFERENCES

[1] L. Titarenko and A. Barkalov, *Methods of Signal Processing for Adaptive Antenna Arrays*. New York, NY, USA: Springer, 2013.

[2] H. L. VanTrees, *Optimum Array Processing*. New York, NY, USA: Wiley, 2002.

[3] X. Li, D. Feng, H. W. Liu, and D. Luo, “Dimension-reduced space-time adaptive clutter suppression algorithm based on lower-rank approximation to weight matrix in airborne radar,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 53–69, Jan. 2014.

[4] Y. Cai and R. C. de Lamare, “Low-complexity variable step-size mechanism for code-constrained constant modulus stochastic gradient algorithms applied to CDMA interference suppression,” *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 313–323, Jan. 2009.

[5] Y. Cai, R. C. de Lamare, B. Champagne, B. Qin, and M. Zhao, “Adaptive reduced-rank receive processing based on minimum symbol-error-rate criterion for large-scale multiple-antenna systems,” *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4185–4201, Nov. 2015.

[6] S. R. D. Poulo, *Adaptive Filtering: Algorithms and Practical Implementations*, 4th ed. New York, NY, USA: Springer, 2013.

[7] C. A. Balanis and P. L. Ioannides, *Introduction to Smart Antennas*. San Rafael, CA, USA: Morgan & Claypool, 2007.

[8] S. Haykin, *Adaptive Filtering Theory*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[9] Y. M. Shi, L. Huang, C. Qian, and H. C. So, “Shrinkage linear and widely linear complex-valued least mean squares algorithms for adaptive beamforming,” *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 119–131, Jan. 2015.

[10] E. Eweda, N. J. Bershad, and J. C. M. Bermudez, “Stochastic analysis of the LMS and NLMS algorithms for cyclostationary white Gaussian and non-Gaussian inputs,” *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4753–4765, Sep. 2018.

[11] B. Qin, Y. Cai, B. Champagne, R. C. de Lamare, M. Zhao, and S. Yousefi, “Low-complexity variable forgetting factor constant modulus RLS-based algorithm for blind adaptive beamforming,” *Signal Process.*, vol. 105, no. 12, pp. 277–282, Dec. 2014.

[12] Y. Cai, R. C. de Lamare, M. Zhao, and J. Zhong, “Low-complexity variable forgetting factor mechanism for blind adaptive constrained constant modulus algorithms,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3988–4002, Aug. 2012.

[13] Y. Cai, X. Wu, M. Zhao, R. C. de Lamare, and B. Champagne, “Low-complexity reduced-dimension space–time adaptive processing for navigation receivers,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 3160–3168, Dec. 2018.

[14] R. Schreiber, “Implementation of adaptive array algorithms,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 5, pp. 1038–1045, Oct. 1986.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 4th ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2013.

[16] H. Ruan and R. C. de Lamare, “Robust adaptive beamforming based on low-rank and cross-correlation techniques,” *IEEE Trans. Signal Process.*, vol. 64, no. 15, pp. 3919–3932, Aug. 2016.

[17] B. D. Van Veen, “Eigenstructure based partially adaptive array design,” *IEEE Trans. Antennas Propag.*, vol. AP-36, no. 3, pp. 357–362, Mar. 1988.

[18] A. M. Haimovich and Y. Bar-Ness, “An eigenanalysis interference canceler,” *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 76–84, Jan. 1991.

[19] J. R. Guerci, J. S. Goldstein, and I. S. Reed, “Optimal and adaptive reduced-rank STAP,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 647–663, Apr. 2000.

[20] J. S. Goldstein and I. S. Reed, “Reduced rank adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 492–496, Feb. 1997.

[21] R. C. de Lamare and R. Sampaio-Neto, “Reduced-rank space–time adaptive interference suppression with joint iterative least squares algorithms for spread-spectrum systems,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1217–1228, Mar. 2010.

[22] L. Wang, R. C. de Lamare, and M. Yukawa, “Adaptive reduced-rank constrained constant modulus algorithm based on joint iterative optimization of filters for beamforming,” *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2983–2997, Jun. 2010.

[23] R. C. de Lamare, L. Wang, and R. Fa, “Adaptive reduced-rank LCMV beamforming algorithms based on joint iterative optimization of filters: Design and analysis,” *Signal Process.*, vol. 90, no. 2, pp. 640–652, 2010.

[24] L. Wang, R. C. D. Lamare, and M. Haardt, “Direction finding algorithms based on joint iterative subspace optimization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2541–2553, Oct. 2014.

[25] Y. Cai, B. Qin, and H. Zhang, “An improved adaptive constrained constant modulus reduced-rank algorithm with sparse updates for beamforming,” *Multidimensional Syst. Signal Process.*, vol. 27, no. 2, pp. 321–340, 2016.

[26] R. Fa and R. C. De Lamare, “Reduced-rank STAP algorithms using joint iterative optimization of filters,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 1668–1684, Jul. 2011.

[27] N. Song, W. U. Alokozai, R. C. de Lamare, and M. Haardt, “Adaptive widely linear reduced-rank beamforming based on joint iterative optimization,” *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 265–269, Mar. 2014.

[28] L. Wang and R. C. de Lamare, “Adaptive constrained constant modulus algorithm based on auxiliary vector filtering for beamforming,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5408–5413, Oct. 2010.

[29] 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, Nov. 1998.

[30] M. Zhang, A. Zhang, and J. Li, “Fast and accurate rank selection methods for multistage Wiener filter,” *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 973–984, Feb. 2016.

[31] S. Werner, M. With, and V. Koivunen, “Householder multistage Wiener filter for space-time navigation receivers,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 975–988, Jul. 2007.

[32] H. Qing, Y. Liu, G. Xie, and J. Gao, “Wideband spectrum sensing for cognitive radios: A multistage Wiener filter perspective,” *IEEE Signal Process. Lett.*, vol. 22, no. 3, pp. 332–335, Mar. 2015.

[33] S. Qiu, W. Sheng, X. Ma, Y. Han, and R. Zhang, "A robust reduced-rank monopulse algorithm based on variable-loaded MWF with spatial blocking broadening and automatic rank selection," *Digital Signal Process.*, vol. 78, pp. 205–217, Jul. 2018.

[34] N. Song, R. C. D. Lamare, M. Haardt, and M. Wolf, "Adaptive widely linear reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4003–4016, Aug. 2012.

[35] C. C. Hu and J. F. Chang, "DS-UWB downlink subband adaptive chip-equalization using reduced-rank multistage Wiener filtering technique," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1883–1889, Apr. 2011.

[36] M. L. Honig and J. S. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 986–994, Jun. 2002.

[37] M. Zhang, A. Zhang, and Q. Yang, "Robust adaptive beamforming based on conjugate gradient algorithms," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 6046–6057, Nov. 2016.

[38] C. Jiang, H. Li, and M. Rangaswamy, "On the conjugate gradient matched filter," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2660–2666, May 2012.

[39] 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. Signal Process.*, vol. 56, no. 10, pp. 5013–5019, Oct. 2008.

[40] S. D. Somasundaram, "A framework for reduced dimension robust capon beamforming," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jun. 2011, pp. 425–428.

[41] F. Huang, W. Sheng, C. Lu, and X. Ma, "A fast adaptive reduced rank transformation for minimum variance beamforming," *Signal Process.*, vol. 92, no. 12, pp. 2881–2887, Dec. 2012.

[42] G. M. Zilli, C. A. Pitz, E. L. O. Batista, and R. Seara, "LCMV-based reduced-rank beamforming algorithm with enhanced tracking capability," *IEEE Wireless Commun. Lett.*, vol. 5, no. 3, pp. 328–331, Jun. 2016.

[43] S. D. Somasundaram, N. H. Parsons, P. Li, and R. C. de Lamare, "Reduced-dimension robust capon beamforming using Krylov-subspace techniques," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 270–289, Jan. 2015.

[44] X. Wu, Y. Cai, M. Zhao, and R. C. de Lamare, "Adaptive widely linear constrained constant modulus reduced-rank beamforming," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 477–492, Feb. 2017.

[45] L. Wang and R. C. de Lamare, "Low-complexity constrained adaptive reduced-rank beamforming algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2114–2128, Oct. 2013.

[46] F. Huang, W. Sheng, C. Lu, and X. Ma, "Robust STAP against unknown mutual coupling based on middle subarray clutter covariance matrix reconstruction," *IEEE Access*, vol. 7, pp. 48109–48118, Apr. 2019.

[47] M. Zhang, X. Chen, and A. Zhang, "A simple tridiagonal loading method for robust adaptive beamforming," *Signal Process.*, vol. 157, pp. 103–107, Apr. 2019.

[48] A. Elnashar, S. M. Elnoubi, and H. A. El-Mikati, "Further study on robust adaptive beamforming with optimum diagonal loading," *IEEE Trans. Antennas Propag.*, vol. 54, no. 12, pp. 3647–3658, Dec. 2006.

TUANNING LIU received the B.S. degree from the School of Information Technology, Luoyang Normal University, Luoyang, China, and the M.S. degree from the School of Computer and Information Engineering, Henan Normal University, Xinxiang, China, in 2016. She is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering, Sichuan University, Chengdu, China. Her research interests include adaptive array antenna, adaptive signal processing, massive MIMO, and wireless communications.

YUANPING ZHOU received the B.S. degree in radio engineering from Chongqing University, Chongqing, China, in 1982, the M.S. degree in electrical engineering from the University of Illinois at Chicago, USA, in 1986, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, USA, in 1999. From 2000 to 2002, he was with Motorola, Inc., as a Lead Systems Engineer. He is currently a Professor with the School of Electronics and Information Engineering, Sichuan University, Chengdu, China. His research interests include space-time signal processing, MIMO systems, adaptive antennas, interference suppression, and modulation methods.

RONGZHEN MIAO received the B.S. degree from the College of Physics and Information Engineer, Fuzhou University, Fuzhou, China. He is currently pursuing the M.S. degree with the School of Electronics and Information Engineering, Sichuan University, Chengdu, China. His research interests include adaptive antenna processing, MIMO communications, and wireless networks.