

Received December 21, 2020, accepted January 25, 2021, date of publication February 1, 2021, date of current version March 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3055947

Efficient Hardware Realization of a New Variable Regularized PAST Algorithm With Multiple Deflation

WEI ZHAO^{1,2}, SHING-CHOW CHAN^{1,3}, (Member, IEEE), AND JIAN-QIANG LIN^{1,3}

¹Institute of Electrical Engineering, Chinese Academy of Sciences, Beijing 100190, China

²Ganjiang Innovation Academy, Chinese Academy of Sciences, Ganzhou 341000, China

³Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

Corresponding author: Jian-Qiang Lin (jqlin@eee.hku.hk)

ABSTRACT This paper proposes a new variant of the projection approximation subspace tracking (PAST) algorithm with multiple deflation (MD) and its efficient hardware architecture. It extends the PAST with deflation (PAST-d) algorithm by performing multiple deflations at each step and relies on a recently introduced variable forgetting factor, and variable regularized PAST algorithm to improve the overall convergence rate, steady-state error, and numerical properties. It shares the same simple hardware structure of the PAST-d algorithm in pipeline realization but offering a more flexible tradeoff between complexity and performance. Moreover, methods for estimating the eigenvalues and the dimension of the signal subspace are proposed. Novel simplifications of the proposed variable forgetting factor (VFF) and variable regularization (VR) PAST-MD algorithm are also developed to avoid the expensive cubic root and division operations involved to facilitate its hardware implementation. Moreover, a combined data-regularization update is introduced to avoid the additional QR decomposition (QRD) update associated with the regularization, at the expense of very slight performance degradation. A novel pipelined hardware implementation of the simplified VFF-VR-PAST-MD algorithm based on the QRD and the COordinate Rotation DIgital Computer (CORDIC) is also proposed and implemented in Xilinx field programmable gate array (FPGA). Thanks to the proposed “root-and division-free” schemes, our proposed architecture can achieve around 20.2% higher working speed and save 1.9% lookup tables (LUTs), 1.8% slice register, and 22.8% digital signal processors (DSPs) over conventional implementation of the proposed architecture. Compared to the previous work, which is based on PAST-d algorithm, the proposed QRD-based algorithms offer better performance and a more flexible tradeoff between hardware resources and performance.

INDEX TERMS Forgetting factor, FPGA, hardware implementation, projection approximation, subspace tracking.

I. INTRODUCTION

Subspace estimation and tracking have important applications in array signal processing [1], [2], system identification [3], [4], speech processing [5], directions of arrival (DOA) estimation in radar, sonar, and mobile/wireless communication systems [6]–[8], etc. For instance, subspace-based methods have been proposed for high-resolution spatial domain spectral analysis in multiple signal classification (MUSIC) method [9], the minimum-norm method [10], the estimation of signal parameter via rotational invariance techniques

(ESPRIT) method [11] and the weighted subspace fitting (WSF) method [12]. These methods are also widely used for estimation and tracking of DOA in antenna arrays. Conventional methods for computing the subspace are usually based on the batch eigenvalue decomposition (ED) or singular value decomposition (SVD) of the data covariance matrix, which can be computational intensive, especially for moving sources. To reduce the arithmetic complexity, efficient subspace tracking algorithms with much lower arithmetic complexity have been proposed [1], [13]–[17]. An efficient algorithm is the projection approximation subspace tracking (PAST) [13] method, which has a low computational complexity of only $\mathcal{O}(Nr)$, where N and r denote the dimension of

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

the input data vector and the dimension of the subspace. Other effective subspace tracking algorithms include the orthonormal PAST (OPAST) algorithm [18], bi-iteration SVD algorithm [16], Bi-LS [19], QS-decomposition-based algorithms [20], fast approximated power iteration (FAPI) [14], YAST [21], etc. For more information on these and other related algorithms, please refer to [22] for more details.

Compared with other algorithms, the conventional PAST algorithm and related algorithms such as the OPAST algorithm are based on the recursive least squares (RLS) algorithm using a fixed forgetting factor (FF). Consequently, efficient hardware structure such as QR decomposition (QRD) using the COordinate Rotation DIgital Computer (CORDIC) technique can be used for its efficient realization. On the other hand, it is known that using a variable FF in the RLS algorithm can considerably improve its tracking speed in time-varying environment and steady-state error in stationary environment [23]–[26]. Another possible problem of the conventional RLS algorithm is that the data covariance matrix may become ill conditioned when the input is not persistently exciting. This is often encountered in situations with signal fading where the signal power level drops rapidly and hence, the estimation error may increase dramatically. To tackle this problem, regularization techniques are commonly employed to reduce the estimation variance [27].

Due to the wide applications of subspace techniques in real time systems, it is highly desirable to develop an efficient hardware for realizing the PAST algorithm. To our best knowledge, hardware realizations of subspace tracker are limited and pioneering works can be found in [28]–[31]. In [29], an efficient parallel implementation of the ED algorithm was proposed. Another novel hardware structure based on a special case of the PAST algorithm called the PAST with deflation (PAST-d) algorithm was proposed in [31]. The PAST-d algorithm extracts the subspace vectors one at a time using the PAST algorithm via the deflation technique. Therefore, its can be realized efficiently in a pipelined structure with a much lower hardware per stage than direct implementation using the QRD [31]. In fact, the hardware complexity of the CORDIC implementation of QRD grows with $\mathcal{O}(r^2)$, which can be substantial if r is large. Moreover, it is rather complicated to vary r in applications where r is adaptively determined. For PAST-d, one can cascade more pipelining stages to determine the appropriate dimension to be used. For QRD, increasing r may need a triangular array with much more elements working in pipeline. However, due to possible error accumulation of the deflation technique and the use of a fixed FF, the steady-state error and tracking speed of the PAST-d algorithm have to be considerably compromised.

Motivated by the needs for a subspace tracking algorithm with good tracking performance and an efficient hardware implementation, we propose in this paper a novel variable FF (VFF) and variable regularized (VR) PAST algorithm with multiple deflation (MD), and its efficient hardware architecture. We first extend the conventional PAST-d algorithm to perform multiple deflations at each step, which can be

achieved by applying successively the basic PAST algorithm with a subspace of dimension $P \geq 1$, instead of restricting it to one as in the PAST-d algorithm. This leads to a faster algorithm and improved numerical properties at the expense of slightly increased complexity at each stage. However, since P is fixed, a QRD implementation would require a fixed complexity of $\mathcal{O}(P^2)$. Moreover, multiple such QRD modules can be cascaded for pipeline implementation and can be made variable to determine the appropriate dimension of the subspace to be used. Thus, the proposed PAST-MD algorithm shares the same hardware advantage of the PAST-d algorithm in reusing the same basic hardware module for modular and pipeline realization. Compared with the basic PAST-d implementation, the error accumulation will be reduced leading to better overall accuracy. Therefore, by choosing an appropriate subspace dimension at each deflation step, P , a more flexible tradeoff between hardware complexity and accuracy over the conventional PAST and PAST-d algorithms in [13], [17], [31] can be achieved. Moreover, we show that the eigenvalues and the dimension of the signal subspace can also be estimated conveniently from the score and residual vectors after each deflation. More precisely, the eigenvalues associated with each extracted subspace can be obtained from the roots of the characteristic equation associated with the covariance matrix of the score vector. Using the ratio of the power of the residual vector to that of the input power, one can also determine the dimension of the signal subspace to retain a given fraction of input power in the signal subspace. To compensate for the degraded convergence and steady mean square error of the PAST-MD algorithm due to the use of multiple deflation, we focus on a recently proposed VFF-VR PAST algorithm [17] as the basic hardware module for improving the convergence speed, steady-state error and stability of the proposed PAST-MD algorithm. In particular, we shall make use of the VFF-VR PAST algorithm in [17] at each stage of the deflation to improve the tracking speed, steady state error and stability. The algorithm in [17] models the channel using a local polynomial and optimizes the FF as well as the regularization to minimize the asymptotic MSE.

Since the VFF scheme of the VFF-VR PAST algorithm in [17] involves cubic root and division operations, novel techniques are proposed in this work to reduce its processing delay and hardware complexity. In particular, novel “root- and division- free” discretized VFF and VR schemes are introduced. Moreover, a combined data-regularization update is introduced to avoid the additional QRD update required for incorporating the regularization, at the expense of very slight performance degradation. Furthermore, the constant coefficient multiplications involved in the proposed VFF and VR schemes are implemented using the canonical signed digits (CSD) or sum-of-power-two numbers [32] resulting in multiplier-less realization.

Finally, we propose a novel pipelined hardware implementation of our proposed VFF-VR-PAST-MD algorithm. The architecture extracts a subspace of P dimension from the input at each pipelined stage using the proposed modified

VFF-VR PAST algorithm. The QR decomposition and the COordinate Rotation DIgital Computer (CORDIC) technique are used to realize the basic PAST algorithm due to its good numerical property and attractive parallel implementation. Moreover, the three-angle complex rotation (TACR) [33] is adopted to simplify the Givens rotation for complex data. Due to the “root- and division-free” schemes, the excessive delay due to division can be avoided, resulting in a lower hardware resources and higher throughput compared to the conventional implementation. To verify the efficiency of the proposed approach, the proposed QRD-based VFF-VR-PAST-MD architecture is implemented in Xilinx Vertex 7 (XC7VX980T) field programmable gate array (FPGA). Compared to the previous work in [31], which is based on the PAST-d algorithm, our proposed QRD-based VFF-VR-PAST-MD algorithms offer better convergence and steady state error performances and a more flexible tradeoff between hardware resources and performance. For a 10-element uniform linear array (ULA), the proposed architecture can be implemented in 22-bit wordlength with a very impressive maximum operating speed of 143 MHz for different values of P at 5 MHz sampling rate. Compared with the conventional implementation of the proposed architecture using multiplier and divider, our proposed work can achieve around 20.2% higher working speed and save 1.9% LUTs, 1.8% Slice Register, and 22.8% DSPs, respectively.

The rest of this paper is organized as follows: the proposed VFF-VR-PAST-MD algorithm will be introduced in section II. The hardware-friendly QRD-based VFF-VR-PAST algorithm and its efficient architecture are described in sections III and IV respectively. Computer simulation, FPGA implementation and comparison with other conventional works will be presented in section V and conclusion is drawn in section VI.

II. THE PROPOSED VFF-VR-PAST-MD ALGORITHM

A. SUBSPACE TRACKING AND DOA TRACKING

Subspace technique is frequently used in DOA estimation. Consider for instance a uniform linear array (ULA) with L omni-directional and identical sensors impinged by K far-field narrow-band uncorrelated sources $s_k[n]$, $k = 1, \dots, K$ with DOAs $\theta_1, \dots, \theta_K$, respectively. The signal vector recorded from the L sensors at time instant n can then be described by the following signal model:

$$\mathbf{x}[n] = \mathbf{A}(\theta)\mathbf{s}[n] + \boldsymbol{\eta}[n], \quad (1)$$

where $\mathbf{x}[n] = [x_1[n], \dots, x_L[n]]^T$ and $\mathbf{s}[n] = [s_1[n], \dots, s_K[n]]^T$ denote the sensor signal and the source signal vector respectively. $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_K)]$ contains the steering vectors associated with the K DOAs, $\theta = [\theta_1, \theta_2, \dots, \theta_K]^T$, of the sources, and $\boldsymbol{\eta}[n]$ is the additive sensor noise vector which is modeled as an independent and identically distributed (IID) white Gaussian noise (AWGN) with zero mean and covariance matrix $\sigma^2\mathbf{I}$, where \mathbf{I} is the identity matrix. For ULAs,

the steering vector of a source with angle θ can be written as

$$\mathbf{a}(\theta) = [1, e^{j2\pi\lambda^{-1}d\sin(\theta)}, \dots, e^{j2\pi\lambda^{-1}(L-1)d\sin(\theta)}]^T, \quad (2)$$

where λ is the wavelength of the propagating signals and d is the inter-sensor spacing. From (1), one gets the following relationship between the signal covariance matrix \mathbf{R}_{xx} and the matrix $\mathbf{A}(\theta)$

$$\mathbf{R}_{xx} = \mathbb{E}[\mathbf{x}[n]\mathbf{x}^H[n]] = \mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}^H(\theta) + \sigma^2\mathbf{I}, \quad (3)$$

where $\mathbf{R}_S = \mathbb{E}[\mathbf{s}[n]\mathbf{s}^H[n]]$ is the source signal covariance matrix. As the K source signals are uncorrelated, the covariance matrix in (3) can be expressed in terms of the signal and noise subspaces as follows

$$\mathbf{R}_{xx} = \mathbf{U}_S\boldsymbol{\Sigma}_S\mathbf{U}_S^H + \mathbf{U}_N\boldsymbol{\Sigma}_N\mathbf{U}_N^H, \quad (4)$$

where $\mathbf{U}_S = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]$ and

$\mathbf{U}_N = [\mathbf{u}_{K+1}, \mathbf{u}_{K+2}, \dots, \mathbf{u}_L]$ are respectively the signal subspace and noise subspace, and $\boldsymbol{\Sigma}_S$ and $\boldsymbol{\Sigma}_N$ are diagonal matrices containing the eigenvalues associated with the K source signals and sensor noise, respectively. In practice, the covariance matrix is estimated from snapshots of the signal vector $\mathbf{x}[n]$ as $\hat{\mathbf{R}}_{xx} = \frac{1}{M} \sum_{n=1}^M \mathbf{x}[n]\mathbf{x}^H[n]$, where M is the number of snapshots. Since the desired steering vectors are in the signal subspace and they are orthogonal to the noise subspace, this implies that $\mathbf{a}^H(\theta_i)\mathbf{U}_N = 0$. Hence, the DOAs can be found from the local peaks of the MUSIC spectrum

$$P(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_N\mathbf{U}_N^H\mathbf{a}(\theta)} = \frac{1}{\|\mathbf{a}^H(\theta)\mathbf{U}_N\|_2^2}, \quad (5)$$

through grid search in the angles given the steering vector of an array. When the number of sources is small, it is advantageous to estimate the signal subspace \mathbf{U}_S and hence the noise subspace by $\mathbf{U}_N = \mathbf{I} - \mathbf{U}_S\mathbf{U}_S^H$. For DOA tracking, the continuous estimation of \mathbf{U}_S is computational intensive and recursive algorithm such as the PAST algorithm can significantly reduce the arithmetic complexity. In this paper, we shall focus on the applications of the proposed VFF-VR-PAST-MD algorithm and hardware architecture to the problem of DOA estimation and tracking.

B. THE PROPOSED PAST ALGORITHM WITH MULTIPLE DEFLATION (PAST-MD)

The signal subspace containing the major \tilde{P} eigenvectors of \mathbf{R}_{xx} can be recursively estimated from $\mathbf{x}[n]$ by minimizing the following least squares (LS) function

$$J(\mathbf{W}[n]) = \sum_{i=1}^n \lambda^{n-i} \|\mathbf{x}[i] - \mathbf{W}[n]\mathbf{y}[i]\|_2^2, \quad (6)$$

where $\mathbf{W}[n] \in C^{L \times \tilde{P}}$, $\hat{\mathbf{x}}[i] = \mathbf{W}[n]\mathbf{y}[i]$, $\mathbf{y}[i] = \mathbf{W}^H[n]\mathbf{x}[i]$ is the projection of $\mathbf{x}[i]$ in $\mathbf{W}[n]$, and λ is a positive forgetting factor less than 1. Hence, the energy outside the space spanned by $\mathbf{W}[n]$, $\|\mathbf{e}[i]\|_2^2 = \|\mathbf{x}[i] - \mathbf{W}[n]\mathbf{y}[i]\|_2^2$ is minimized. In PAST, the projection approximation $\tilde{\mathbf{y}}[i] \approx \mathbf{W}^H[n-1]\mathbf{x}[i]$ is used so that (6) can be simplified and solved using the recursive least squares (RLS) algorithm. Since $\|\mathbf{e}[i]\|_2^2$ is the

sum of the squared values of its components, the PAST algorithm can also be viewed as the following L LS sub-problems with the same input $\tilde{\mathbf{y}}[i]$

$$\tilde{\mathcal{J}}(\mathbf{w}_l[n]) = \sum_{l=1}^L \sum_{i=1}^n \lambda^{n-i} \|\mathbf{x}_l[i] - \mathbf{w}_l^H[n]\tilde{\mathbf{y}}[i]\|_2^2, \quad (7)$$

$l = 1, \dots, L$, where $\mathbf{w}_l^H[n]$ denotes the l -th row of $\mathbf{W}[n]$. It should be noted that these sub-problems are independent of each other and hence, they can be solved independently and the optimal solution to (7) is

$$\mathbf{w}_l[n] = \mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1} \mathbf{R}_{\tilde{\mathbf{y}}\mathbf{x}_l}[n], \quad (8)$$

where $\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}[n] = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{y}}[i]\tilde{\mathbf{y}}^H[i]$ and $\mathbf{R}_{\tilde{\mathbf{y}}\mathbf{x}_l}[n] = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{y}}[i]\mathbf{x}_l^*[i]$. Applying the RLS algorithm for solving (8), one gets the PAST algorithm in Table 1.

TABLE 1. The PAST algorithm [13].

<p>Initialization: Choose $\lambda, \mathbf{P}[0] = \delta \mathbf{I}$ and $\mathbf{W}^T[0] = [\mathbf{I}_{\tilde{P} \times \tilde{P}}, \mathbf{0}_{\tilde{P} \times (L-\tilde{P})}]$ for appropriate $\lambda \in (0, 1], \delta > 0$.</p> <p>Recursion: For $n = 1, 2, \dots, N$ do $\tilde{\mathbf{y}}[n] = \mathbf{W}^H[n-1]\mathbf{x}[n]$ $\mathbf{h}[n] = \mathbf{P}[n-1]\tilde{\mathbf{y}}[n]$ $\mathbf{g}[n] = \mathbf{h}[n]/(\lambda + \tilde{\mathbf{y}}^H[n]\mathbf{h}[n])$ $\mathbf{P}[n] = \text{Tri}\{\mathbf{P}[n-1] - \mathbf{h}[n]\mathbf{h}^H[n]\}/\lambda$ $\mathbf{e}[n] = \mathbf{x}[n] - \mathbf{W}[n-1]\tilde{\mathbf{y}}[n]$ $\mathbf{W}[n] = \mathbf{W}[n-1] + \mathbf{e}[n]\mathbf{g}^H[n]$ End Notes: The operator $\text{Tri}(\cdot)$ indicates that only the upper (or lower) triangular part of $\mathbf{P}[n]$ is calculated and its Hermitian transposed version is copied to the other lower (or upper) triangular part.</p>
--

1) PAST-D AND PAST-MD

In the PAST with deflation (PAST-d) algorithm, the subspace is extracted one by one by means of deflation. Specifically, let the optimal weight vector at the first stage of the deflation process be $\mathbf{w}^{(1)}[n]$ and $\mathbf{x}^{(0)}[n] = \mathbf{x}[n]$. Then, the residual vector after extraction, $\mathbf{x}^{(1)}[n] = \mathbf{x}^{(0)}[n] - \mathbf{w}^{(1)}[n]\tilde{\mathbf{y}}^{(0)}[n]$, will consist of components from the remaining subspace and the process can be repeated for $\mathbf{x}^{(1)}[n]$ and so on to give

$$\mathbf{x}^{(r)}[n] = \mathbf{x}^{(r-1)}[n] - \mathbf{w}^{(r)}[n]\tilde{\mathbf{y}}^{(r-1)}[n], \quad (9a)$$

$$\tilde{\mathbf{y}}^{(r-1)}[n] = (\mathbf{w}^{(r)}[n-1])^H \mathbf{x}^{(r-1)}[n], r = 1, \dots, \tilde{P}, \quad (9b)$$

and hence the eigenvectors can be successively extracted as $\mathbf{w}^{(r)}[n], r = 1, \dots, \tilde{P}$. The final subspace has a dimension of \tilde{P} . The PAST-d algorithm is summarized in Table 2.

One of the advantages of the conventional PAST-d algorithm is its simplicity, especially for hardware implementation as the basic module for extracting one eigenvector can be time multiplexed or pipelined for efficient realization. On the other hand, the deflation process in (9a) may affect the orthogonality of the vectors extracted due to error accumulation and its convergence rate will also be slower.

TABLE 2. The PAST-d algorithm [13].

<p>Initialization: Choose $\lambda, \mathbf{W}[0] = [\mathbf{w}^{(1)}[0], \mathbf{w}^{(2)}[0], \dots, \mathbf{w}^{(\tilde{P})}[0]]$, $\mathbf{W}^T[0] = [\mathbf{I}_{\tilde{P} \times \tilde{P}}, \mathbf{0}_{\tilde{P} \times (L-\tilde{P})}]$ for appropriate $\lambda \in (0, 1]$.</p> <p>Recursion: For $n = 1, 2, \dots, N$ do $\mathbf{x}^{(0)}[n] = \mathbf{x}[n]$ For $r = 1, 2, \dots, \tilde{P}$ do $\tilde{\mathbf{y}}^{(r-1)}[n] = (\mathbf{w}^{(r)}[n-1])^H \mathbf{x}^{(r-1)}[n]$ $d^{(r)}[n] = \lambda d^{(r)}[n-1] + \tilde{\mathbf{y}}^{(r-1)}[n] ^2$ $\mathbf{e}^{(r)}[n] = \mathbf{x}^{(r-1)}[n] - \mathbf{w}^{(r)}[n-1]\tilde{\mathbf{y}}[n]$ $\mathbf{g}[n] = \mathbf{h}[n]/(\lambda + \tilde{\mathbf{y}}^H[n]\mathbf{h}[n])$ End $\mathbf{P}[n] = \text{Tri}\{\mathbf{P}[n-1] - \mathbf{h}[n]\mathbf{h}^H[n]\}/\lambda$ $\mathbf{e}[n] = \mathbf{x}[n] - \mathbf{W}[n-1]\tilde{\mathbf{y}}[n]$ $\mathbf{W}[n] = \mathbf{W}[n-1] + \mathbf{e}[n]\mathbf{g}^H[n]$ End Notes: The operator $\text{Tri}(\cdot)$ indicates that only the upper (or lower) triangular part of $\mathbf{P}[n]$ is calculated and its Hermitian transposed version is copied to the other lower (or upper) triangular part.</p>
--

Here, we extend the conventional PAST-d algorithm to perform multiple deflations at each step. This leads to a faster algorithm and improved numerical properties at the expenses of slightly increased complexity. More precisely, we propose to extract a subspace of small dimension successively instead of one. This is possible by using successively the basic PAST algorithm with a subspace of dimension P , say, as follows

$$\mathbf{x}^{(r)}[n] = \mathbf{x}^{(r-1)}[n] - \mathbf{W}^{(r)}[n]\tilde{\mathbf{y}}^{(r-1)}[n], \quad (10a)$$

$$\tilde{\mathbf{y}}^{(r-1)}[n] = (\mathbf{W}^{(r)}[n-1])^H \mathbf{x}^{(r-1)}[n], r = 1, \dots, R, \quad (10b)$$

where $\mathbf{W}^{(r)}[n] \in C^{L \times P}$, R denotes the number or stage of deflation and the final subspace dimension has a size of RP . Specifically, when $P = 1$, the PAST-MD algorithm will be reduced to the PAST-d algorithm.

In general, the size of the subspace extracted at each step of deflation may be different. Of course, using the same size has the advantage of possible reuse of hardware modules and modular hardware realization in pipeline realization. Thus, the proposed algorithm offers a more flexible trade-off between arithmetic and hardware complexity over the conventional PAST and PAST-d algorithms. The proposed PAST-MD algorithm is summarized in Table 3. A comparison of arithmetic complexity of the PAST, PAST-d, and PAST-MD algorithms is given in Table 4. We now proposed novel VFF and VR schemes for improving the convergence speed, steady-state error and stability of these algorithms. The hardware architecture for their efficient realization will be discussed later in Section IV.

2) EIGENVALUES ESTIMATION AND ADAPTIVE SUBSPACE DIMENSION DETERMINATION

One of the advantage of the PAST-d algorithm is its simplicity in estimating the eigenvalues of the subspace, which is in

fact the power of the projected score $\bar{y}^{(r-1)}[n]$ in (9b). In the PAST-MD algorithm, each extract subspace has a dimension of P , and hence the projected score $\bar{y}^{(r-1)}[n]$ will be a P -th vector. Since $\mathbf{W}^{(r)}[n] \in \mathcal{C}^{L \times P}$ only span the subspace and they are not eigenvectors, one cannot estimate the corresponding eigenvalues from the power of the elements in $\bar{y}^{(r-1)}[n]$ as in PAST-d. In fact, the eigenvalues can be estimated from the eigenvalues of the covariance matrix of $\bar{y}^{(r-1)}[n]$, $\mathbf{C}_{y^{(r-1)}} = \mathbb{E}[\bar{y}^{(r-1)}[n]\bar{y}^{(r-1)}[n]^H]$. Since P is usually not a very large number, the eigenvalues of $\mathbf{C}_{y^{(r-1)}}$ can be estimated by solving the roots of the characteristic equations as follows

$$p^{(r-1)}(\lambda) = |\mathbf{C}_{y^{(r-1)}} - \lambda \mathbf{I}| = 0. \quad (11)$$

Moreover, for P less than or equal to 4, analytical formulas are available to compute all the roots of $p^{(r-1)}(\lambda)$, which allows the eigenvalues to be computed on the fly. From our simulation in the supplementary materials, the eigenvalues can be estimated quickly with high accuracy.

Another advantage of the PAST-d and PAST-MD algorithms is its convenient in determining the dimension of the subspace to be tracked. This can be done by examining the ratio of the powers of the residual vector at the r -th deflation step to that of the input:

$$E^{(r)} = \mathbb{E}[\|\mathbf{x}^{(r)}\|_2^2] / \mathbb{E}[\|\mathbf{x}\|_2^2], \quad (12)$$

$E^{(r)}$ represents the fraction of the input energy in the residual vector after the r -th stage of deflation. Therefore, if it is sufficiently small, one can terminate the deflation process and determine the dimension of the signal subspace of interest. Next, we shall focus on methods to improve the convergence speed and the steady-state error of the basic PAST algorithm.

C. THE PROPOSED VFF-VR-PAST-MD ALGORITHM

The FF of the PAST algorithm plays an important role in its convergence speed and steady-state error. In stationary environment, a large FF is desirable to achieve a low mean square error (MSE). For time-varying environment, a relatively small forgetting factor should be employed to achieve fast tracking speed. Thus, a variable FF is desirable for both stationary and dynamic environment. Due to the close relationship between the RLS algorithm and the PAST algorithm, the VFF-VR scheme for real-valued RLS algorithm [34] recently proposed by one of the authors can be extended to the complex case [17]. In particular, the steady-state MSD of $w_l[n]$ in (7) under the local polynomial time-varying model of the RLS is given by [34],

$$J_{MSD}(w_l[n]) \approx \frac{1 - \lambda[n]}{1 + \lambda[n]} \sigma_{\Sigma_l}^2 \text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1}) + \frac{\lambda^2[n]}{(1 - \lambda[n])^2} \sigma_{w_l}^2, \quad (13)$$

where $\sigma_{\Sigma_l}^2$ and $\sigma_{w_l}^2$ represent the total error variance $\mathbb{E}[\|\mathbf{e}[n]\|_2^2]$ and the variance of the true parameter $w_l[n]$, respectively. For notational convenience, we shall drop the subscript (r) in describing the operations at the r -th deflation

step in the subsequent discussion. Thus, the VFF and VR procedures described will be applied to the PAST algorithm at each stage of deflation. The expression is the same as the real-valued case but the variance $\sigma_{\Sigma_l}^2$ and $\sigma_{w_l}^2$ are for the complex noise and weight vector. The estimation of these quantities will be discussed later in this section. Since the mean squares deviation (MSD) of the PAST estimator is equal to the sum of all its component, one can obtain from [34] the MSD of $\mathbf{W}[n]$ by summing (13) for all $l = 1, \dots, L$, which yields

$$J_{MSD}(n) \approx \frac{1 - \lambda[n]}{1 + \lambda[n]} \sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1}) + \frac{\lambda^2[n]}{(1 - \lambda[n])^2} \sigma_W^2, \quad (14)$$

where $\sigma_{\Sigma}^2 = \sum_{l=1}^L \sigma_{\Sigma_l}^2$ and $\sigma_W^2 = \sum_{l=1}^L \sigma_{w_l}^2$ denote the total noise variance and variance of the true parameter $\mathbf{W}[n]$ (or system variance), respectively. Similarly to [34], the locally optimal $\lambda[n]$ at time n , which minimizes $J_{MSD}[n]$, can be obtained by partial differentiating (14) with respect to $\lambda[n]$ and setting the resultant derivative to zero, which yields

$$\frac{-2}{(1 + \lambda[n])^2} \sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1}) + \frac{2\lambda[n]}{(1 - \lambda[n])^3} \sigma_W^2 = 0. \quad (15)$$

Using the change of variable $\mu = (1 + \lambda[n]) / (1 - \lambda[n])$ as in [34], (15) can be reduced to $\mu^2(\mu - 1) = 2\sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1}) / \sigma_W^2$. Assuming that the value of μ is much larger than one when $\lambda[n]$ is near to one, we have $\mu - 1 \approx \mu$ and the desired FF can be simplified to

$$\lambda_{opt} = (\mu - 1) / (\mu + 1), \quad (16)$$

where $\mu = (2\sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1}) / \sigma_W^2)^{1/3}$. It should be noticed that the term $\text{Tr}(\mathbf{R}_{\bar{y}\bar{y}}^{-1})$ can be estimated by using the value of $\mathbf{P}[n]$ in the PAST algorithm as $(\sum_{i=1}^n \lambda[i]) \text{Tr}(\mathbf{P}[n])$. On the other hand, the noise variance and system variance can be estimated from the error vector $\mathbf{e}[n] = \mathbf{x}[n] - \mathbf{W}[n]\bar{y}[n]$ and difference of weight vector $\Delta \mathbf{W}[n] = \mathbf{W}[n] - \mathbf{W}[n-1]$ as follows:

$$\hat{\sigma}_{\Sigma}^2[n] = \lambda_L \hat{\sigma}_{\Sigma}^2[n-1] + (1 - \lambda_L) \|\mathbf{e}[n]\|_2^2, \quad (17)$$

$$\hat{\sigma}_W^2[n] = \lambda[n] \hat{\sigma}_W^2[n-1] + (1 - \lambda[n]) \|\Delta \mathbf{W}[n]\|_2^2, \quad (18)$$

where λ_L and $\lambda[n]$ denote respectively a large fixed FF and the FF obtained at time n . Due to the use of a large FF to estimate the steady-state noise, it may experience a lag when sudden system change occurs. To address this issue, a system change detection scheme is employed based on the long-term and short-term estimates of noise variance, in which (17) can be regarded as a long-term estimate $\hat{\sigma}_{\Sigma-L}^2$ and the short-term estimate is given by

$$\hat{\sigma}_{\Sigma-S}^2[n] = \lambda_S \hat{\sigma}_{\Sigma-S}^2[n-1] + (1 - \lambda_S) \|\mathbf{e}[n]\|_2^2, \quad (20)$$

where λ_S is a relatively small FF compared to λ_L .

Suppose that $\hat{\sigma}_{\Sigma-S}^2 \geq \chi \hat{\sigma}_{\Sigma-L}^2[n]$ for some sufficiently large constant χ , it suggests that there is a sudden system change. Under this situation, the system variance is roughly equal to the total error variance and a smaller FF should be used for the latter update. Hence, we suggest to estimate

TABLE 3. Summary of efficient QRD-based VFF-VR-PAST-MD Algorithm.

<p>Initialization: Choose $\lambda[0]$, $\chi = 4$, $\mathbf{R}[0] = \delta \mathbf{I}$, $\mathbf{U}[0] = \mathbf{0}$ and $\mathbf{W}[0] = \mathbf{I}$ for appropriate $\lambda[0] \in (0, 1]$, $\delta > 0$.</p> <p>Recursion for time: Given $\mathbf{R}[n-1]$, $\mathbf{U}[n-1]$, $\mathbf{W}[n-1]$ and $\mathbf{x}[n]$, compute at time n</p> <p>Recursion for deflation: Given $\mathbf{x}^{(r)}[n]$ and $\mathbf{W}^{(r)}[n-1]$, compute at deflation r</p> <ol style="list-style-type: none"> (1) Compute $\bar{\mathbf{y}}^{(r)}[n]$ using (10b), (2) Select $\lambda^{(r)}[n]$ from (13), (3) Detect system change in (19), (4) Compute $\kappa^{(r)}[n]$ using (20), (5) Update for $\mathbf{R}^{(r)}[n]$ and $\mathbf{U}^{(r)}[n]$, $\begin{bmatrix} \mathbf{R}^{(r)}[n] & \mathbf{U}^{(r)}[n] \\ \mathbf{0}^H & c[n] \end{bmatrix} = \mathbf{Q}[n].$ $\begin{bmatrix} \sqrt{\lambda[n]} \mathbf{R}^{(r)}[n-1] & \sqrt{\lambda[n]} \mathbf{U}^{(r)}[n-1] \\ (\bar{\mathbf{y}}^{(r)}[n])^H + \sqrt{\kappa^{(r)}[n]} P \mathbf{d}_i^H[n] & (\mathbf{x}^{(r)}[n])^H \end{bmatrix}, (*)$ <ol style="list-style-type: none"> (6) Compute $\mathbf{W}^{(r)}[n]$ from $(\mathbf{W}^{(r)}[n])^H = (\mathbf{R}^{(r)}[n])^{-1} \mathbf{U}^{(r)}[n]$ <p>End recursion for deflation Update $\mathbf{x}^{(r+1)}[n]$ in (10a).</p> <p>End recursion for time</p>

$\hat{\sigma}_W^2[n]$ by $\hat{\sigma}_{\Sigma_S}^2[n]$. When $\hat{\sigma}_{\Sigma_S}^2[n] < \chi \hat{\sigma}_{\Sigma_L}^2[n]$, the algorithm is converging and hence the system variance should be updated by (18). This suggests the following equation for updating $\hat{\sigma}_W^2[n]$, shown at the bottom of this page, and χ is an algorithmic parameter which can be chosen as 4, which corresponds to 99.99% confidence that system changes have been detected. From simulation results to be presented in Section V, it is found that the VFF scheme can improve considerably the convergence speed and steady-state error over the conventional PAST algorithm.

In practical applications, the input power may be time-varying and may not be persistently exciting especially at low signal level, the covariance matrix may be in poor condition or even singular, which may affect numerical stability of the PAST algorithm as it is based on the RLS algorithm. To tackle this problem, the regularized RLS algorithm can be used where a regularization term $\kappa[n]\mathbf{D}$ is added to the covariance matrix $\mathbf{R}_{\bar{\mathbf{y}}\bar{\mathbf{y}}}[n]$ in (8) to improve its condition number and hence the variance of the estimator in exchange for a certain bias. The resulting solution is given by

$$(\mathbf{R}_{\bar{\mathbf{y}}\bar{\mathbf{y}}} + \kappa[n]\mathbf{D})\mathbf{w}[n] = \mathbf{R}_{\bar{\mathbf{y}}\mathbf{x}}[n], \quad (21)$$

where \mathbf{D} is a positive definite symmetric matrix and $\kappa[n] \geq 0$ is a regularization parameter. Here, we adopt the variable regularization parameter proposed in [27] with $\mathbf{D} = \mathbf{I}$ and

$$\kappa[n] = \sqrt{\gamma[n] \sigma_{\Sigma}^2 \sigma_{\bar{\mathbf{y}}}^2 \|\mathbf{W}_0\|_2^{-2}}, \quad (22)$$

TABLE 4. Complexity comparison of PAST, PAST-d and VFF-VR-PAST-MD algorithms.

Algorithm	Complexity
PAST [13]	$3L\tilde{P} + \mathcal{O}(\tilde{P}^2)$
PAST-d [13]	$4L\tilde{P} + \mathcal{O}(\tilde{P})$
VFF-VR-PAST-MD	$7L\tilde{P} + 3L + 9\tilde{P} + \mathcal{O}(\tilde{P})$

where σ_{Σ}^2 and $\sigma_{\bar{\mathbf{y}}}^2$ represent the total noise power and signal power, respectively. \mathbf{W}_0 denotes the theoretical weight vector and $\gamma[n] = (1 - \lambda[n]) / (\frac{\chi}{2}\lambda[n] + (1 - \lambda[n])(1 + L))$. Since the subspace vectors should be orthogonal to each other and has unit norm, hence $\|\mathbf{W}_0\|_2^2$ is equal to P .

The arithmetic complexity comparison of various algorithms is shown in Table 4. The arithmetic complexities of the PAST and PAST-MD algorithms are $3L\tilde{P} + \mathcal{O}(\tilde{P}^2)$ and $4L\tilde{P} + \mathcal{O}(\tilde{P})$ per update, respectively, where \tilde{P} is the dimension of the tracked subspace and L is the dimension of the input data vector. The proposed VFF and VR schemes require additional $3L\tilde{P} + 3L + 7\tilde{P}$ and $2\tilde{P}$ operations, respectively, per update. Hence, the total arithmetic complexity of the proposed VFF-VR-PAST-MD is $7L\tilde{P} + 3L + 9\tilde{P} + \mathcal{O}(\tilde{P})$. It should be noted that $\tilde{P} = RP$ and when P is equal to one, the PAST-MD approach will reduce to the PAST-d algorithm. It can be noticed that the proposed approach has comparable and acceptable arithmetic complexity compared to other algorithms.

III. HARDWARE-FRIENDLY SIMPLIFICATIONS OF THE PROPOSED VFF-VR-PAST-MD ALGORITHM

In Section II, we have proposed an efficient class of VFF-VR-PAST-MD algorithms. The proposed VFF and VR schemes lead to better convergence speed and steady-state error. Using deflation with more than one vector at a time, a more flexible tradeoff between arithmetic complexity, convergence speed and numerical accuracy can be achieved. Moreover, it is amenable to hardware implementation in a pipelined manner through iterative reuse of the basic RLS hardware module.

The basic VFF-VR-PAST-MD algorithm however still present certain challenges to efficient hardware implementation: 1) Firstly, from the expression of FF in (16), it can be seen that it requires the cubic root operation, which is rather complicated to evaluate and implement in hardware. Secondly, it also involves a division, which may introduce much latency in pipeline implementation and hence lead to a lower operating frequency. 2) The basic RLS is rather sensitive to numerical error and hence QR decomposition (QRD)-based algorithm using CORDIC is usually preferred. In this case, the term $\text{Tr}(\mathbf{R}_{\bar{\mathbf{y}}\bar{\mathbf{y}}}^{-1})$ is expensive to compute since QRD updates only the Cholesky factor of $\mathbf{R}_{\bar{\mathbf{y}}\bar{\mathbf{y}}}$ and computing

$$\hat{\sigma}_W^2[n] = \begin{cases} \hat{\sigma}_{\Sigma_S}^2[n], & \hat{\sigma}_{\Sigma_S}^2[n] \geq \chi \hat{\sigma}_{\Sigma_L}^2[n] \\ \lambda[n] \hat{\sigma}_W^2[n-1] + (1 - \lambda[n]) \|\Delta \mathbf{W}[n]\|_2^2, & \hat{\sigma}_{\Sigma_S}^2[n] < \chi \hat{\sigma}_{\Sigma_L}^2[n] \end{cases} \quad (19)$$

$\mathbf{R}_{\hat{y}\hat{y}}^{-1}$ is expensive. In [34], it is proposed to approximate $\mathbf{R}_{\hat{y}\hat{y}}$ as a diagonal matrix so that $\mathbf{R}_{\hat{y}\hat{y}}^{-1}$ can be more conveniently computed from the reciprocal of the diagonal elements of $\mathbf{R}_{\hat{y}\hat{y}}$. Though this approach considerably simplifies the updating of the VFF and yields good performance, the division operations may limit the maximum operating frequency in hardware implementation. Moreover, the incorporation of the regularization term in the QRD usually requires an additional QRD update, which will also slow down the throughput of the QRD RLS algorithm.

We now propose several novel techniques to facilitate the hardware implementation of the proposed VFF-VR-PAST-MD algorithm and the overall architecture will be presented later in Section IV. In particular, a novel ‘‘cubic root- and division- free’’ discretized VFF and VR schemes will be introduced for addressing the first challenge mentioned above. A novel ‘‘division-free’’ method is also proposed for computing $\text{Tr}(\mathbf{R}_{\hat{y}\hat{y}}^{-1})$. Furthermore, a combined data-regularization update is introduced to avoid the additional QRD update, at the expense of very slight performance degradation.

A. THE DISCRETIZED VFF AND VR SCHEMES

From the expression of FF in (16), it can be seen that it requires the cubic root operation, which is rather complicated to evaluate and implement in hardware. Moreover, it also involves a division, which may introduce much latency in pipeline implementation and hence lead to a lower operating frequency. To tackle this problem, we propose a novel discretized FF scheme to avoid these expansive operations. More precisely, we quantize the FF inside the given range, say [0.9, 1], into a set of representative FF values and select the best one with the lowest MSD as given by (13).

$$J[\hat{\lambda}_j] \approx \frac{1 - \hat{\lambda}_j}{1 + \hat{\lambda}_j} \sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\hat{y}\hat{y}}^{-1}) + \frac{\hat{\lambda}_j}{(1 - \hat{\lambda}_j)^2} \sigma_W^2, \quad (23)$$

where $\hat{\lambda}_j, j = 1, \dots, F$ represent the set of F discretized values of FF. Moreover, these representative values are chosen as canonical signed digits (CSD) or sum-of-powers-of-two (SOPOT) coefficients so that the terms $(1 - \hat{\lambda}_j)/(1 + \hat{\lambda}_j)$ and $\hat{\lambda}_j(1 - \hat{\lambda}_j)^{-2}$ involving $\hat{\lambda}_j$ above can be pre-computed and represented as CSD. Since multiplication with CSD can be implemented as additions and hardware shifts, these multiplications can be realized in additions only. Moreover, the simultaneous multiplication of $\sigma_{\Sigma}^2 \text{Tr}(\mathbf{R}_{\hat{y}\hat{y}}^{-1})$ and σ_W^2 with different values of $(1 - \hat{\lambda}_j)/(1 + \hat{\lambda}_j)$ and $\hat{\lambda}_j(1 - \hat{\lambda}_j)^{-2}$ for $j = 1, \dots, F$ can be implemented as a multiplier block [35]. From the simulation to be presented in Section V, it is found that $F = 4$ is sufficient to give a performance close to that of (16)). Therefore, the proposed discretized FF significantly reduces the arithmetic and hardware complexity in realizing the VFF-PAST algorithm.

Meanwhile, since $\sqrt{\gamma/P}$ in updating the VF in (8) is also a function of these discretized FFs, they can again be pre-computed and represented as CSD for multiplier-less

realization. Using the selected FF, say $\hat{\lambda}_j$, the corresponding value of $\sqrt{\hat{\gamma}_j/P}$ can then be multiplied to $\sigma_{\eta}^2 \sigma_y^2$. Consequently, the arithmetic and hardware complexity of computing the variable regularization in (22) can also be reduced significantly. In summary, the discretized VFF-VR version further simplifies the computation of the VFF and VR schemes, which avoids the expensive cubic root and division operations and simplifies the evaluation of regularization parameter.

B. QRD-BASED VFF-VR-PAST ALGORITHM

In the QR-RLS algorithm, the rank-1 update of $\mathbf{R}_{\hat{y}\hat{y}} = \lambda[n]\mathbf{R}_{\hat{y}\hat{y}}[n-1] + \hat{y}[n]\hat{y}^H[n]$ can be efficiently implemented by updating the Cholesky factor $\mathbf{R}[n]$ of $\mathbf{R}_{\hat{y}\hat{y}}[n] = \mathbf{R}^H[n]\mathbf{R}[n]$ recursively from the data using Givens rotation or Householder reflection. To avoid the complicated update of the full rank term $\kappa[n]\mathbf{D}$, it can be factorized as $\kappa[n]\mathbf{D} = \kappa[n] \sum_{i=1}^P \mathbf{d}_i \mathbf{d}_i^T$, where \mathbf{d}_i is a P -length vector. Hence, $\mathbf{R}_{\hat{y}\hat{y}}[n]$ can be updated once for the data vector $[\hat{y}^T[n], \mathbf{x}^T[n]]$ followed by the regularization vectors $\mathbf{d}_i, i = 1, \dots, P$. To avoid updating the regularization term at once, one can update one \mathbf{d}_i per time so that the regularization term can be gradually imposed over time. For $\mathbf{D} = \mathbf{I}$, updating \mathbf{d}_i among to using a data vector $[\sqrt{\kappa[n]P}\mathbf{d}_i, \mathbf{0}]$ in the QRD update, where \mathbf{d}_i has its i -th element equal to one and zero otherwise. The factor \sqrt{P} is to account for the reduced updating of the each regularization vector. At each iteration, a regularization vector can be randomly picked from $\mathbf{d}_i, i = 1, \dots, P$, or it may be presented in a sequential order.

To further reduce the number of QRD updates, we notice that the data and regularization updates can be viewed as two equations to be satisfied as follows: $\mathbf{W}[n]\hat{y}[n] = \mathbf{x}[n]$ and $\mathbf{W}[n]\sqrt{\kappa[n]P}\mathbf{d}_i[n] = \mathbf{0}$. Adding the equations gives $\mathbf{W}[n](\hat{y}[n] + \sqrt{\kappa[n]P}\mathbf{d}_i[n]) = \mathbf{x}[n]$, which serves as an approximation to the original two measurement equations. Using this approximate data input $[\hat{y}^H[n] + \sqrt{\kappa[n]P}\mathbf{d}_i^H[n], \mathbf{x}^H[n]]$ allows us to reduce the QRD update from two to one. Simulation results show that satisfactory performance can be achieved with a slight performance degradation. The reason is that normally $\kappa[n]$ is close to zero when the $\mathbf{R}_{\hat{y}\hat{y}}[n]$ is well-conditioned. During input signal fading, the presence of a regularization term greatly improves the condition number and hence reduces the variances of the estimator. On the other hand, the exact value of the regularization term is less critical. This variant of the proposed algorithm is referred to as combined VR (CVR) scheme and the resultant algorithm is called VFF-CVR-PAST.

C. DIAGONAL AND DIVISION-FREE APPROXIMATION OF $\text{Tr}(\mathbf{R}_{\hat{y}\hat{y}}^{-1}[n])$

As mentioned earlier, when computing the term $\text{Tr}(\mathbf{R}_{\hat{y}\hat{y}}^{-1}[n])$ for updating the VFF, $\mathbf{R}_{\hat{y}\hat{y}}[n]$ is treated as a diagonal matrix and its p -th diagonal value is recursively estimated as follows

$$\sigma_{\hat{y}_p}^2[n] = \lambda_p \sigma_{\hat{y}_p}^2[n-1] + (1 - \lambda_p) |\hat{y}_p[n]|^2, \quad (24)$$

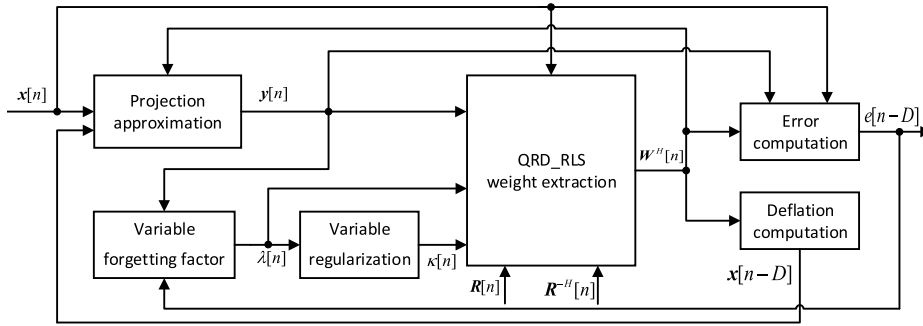


FIGURE 1. Data flow diagram of the proposed QRD-based VFF-VR-PAST-MD architecture.

where $\bar{y}_p[n]$ is the p -th element of $\bar{\mathbf{y}}[n]$ and $\lambda_p \in (0, 1)$ is a forgetting factor. Computing $\text{Tr}(\mathbf{R}_{\bar{\mathbf{y}\bar{\mathbf{y}}}}^{-1}[n])$ amounts to taking the reciprocal of $\sigma_{\bar{y}_p}^2[n]$, $p = 1, \dots, P$, and adding them together. This requires divisions which may limit the maximum operating speed in hardware implementation. Here, we propose to quantize $\sigma_{\bar{y}_p}^2[n]$ to discrete levels so that the reciprocals of these values can be precomputed and implemented as multiplications. Moreover, since these are constant coefficient multiplications, they can be implemented in CSD using limited additions and shifts. More precisely, suppose that the sensor signals are in normalized fixed-point format with a magnitude less than one. Then $|\bar{y}_p|^2 = \|\mathbf{w}_p^H[n-1]\mathbf{x}[n]\|_2^2 \leq \|\mathbf{w}_p[n-1]\|_2^2 \|\mathbf{x}[n]\|_2^2 = L$ since \mathbf{w}_p , the p -th column of \mathbf{W} , has unit norm. As its power is limited to L , it makes sense to quantize its value to a set of reconstruction levels $\hat{\sigma}_j$, $j = 1, \dots, q$, where q is the total number of discrete levels in approximating $\sigma_{\bar{y}_p}^2[n]$. The quantization process can be rewritten as

$$\hat{\sigma}_j = Q[\sigma_{\bar{y}_p}^2[n]], \quad j \in \{1, \dots, q\}. \quad (25)$$

From simulations to be presented in Section V, this approximation gives satisfactory performance when $\sigma_{\bar{y}_p}^2[n]$ is uniformly quantized to 16 levels for a $L = 10$ sensor array. For an L -length sensor antenna, simulated experiments show that discrete levels is $2^{\lceil \log_2 2L \rceil}$ or $2^{\lceil \log_2 4L \rceil}$ where $\lceil a \rceil$ denotes the nearest integer smaller than a . Since $\hat{\sigma}_j, j \in \{1, \dots, q\}$, is a set of constants, their reciprocals can be computed offline as $\hat{\sigma}_j^{-1}$, $j \in \{1, \dots, q\}$. Consequently, the division is now replaced by a constant multiplication from one of these values. Moreover, since they are constant coefficients multiplications, they can be represented as CSD for multiplier-less realization. Our derived efficient QRD-based VFF-VR-PAST-MD algorithm is summarized in Table 3.

IV. HARDWARE ARCHITECTURE OF THE PROPOSED QRD-BASED VFF-VR-PAST-MD ALGORITHM

According to the functionality of each operation, the proposed architecture for QRD-based VFF-VR-PAST is divided into six processing units, including projection approximation unit, variable forgetting factor unit, variable regularization unit, QRD-RLS weight extraction unit, and error computation unit. The processing units work in a pipelined manner

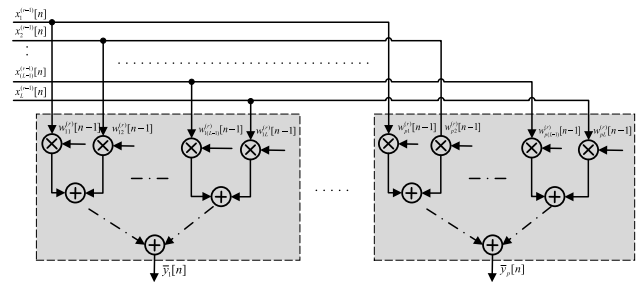


FIGURE 2. Basic structure of projection approximation.

and the data flow diagram of the proposed architecture is shown in Fig. 1. This unit-based design can simplify the modification of the hardware architecture for other alternative subspace tracking algorithms. The structures and functionalities of these processing units at the r -th iteration are briefly summarized as follows.

A. PROJECTION APPROXIMATION UNIT

The unit computes the projection approximation $\bar{\mathbf{y}}[n] = \mathbf{W}^H[n-1]\mathbf{x}[n]$ as matrix-vector multiplication and its structure is illustrated in Fig 2.

B. VARIABLE FORGETTING FACTOR UNIT

As we described in Section II-B, the computation of VFF can be simplified by using the discretized FF scheme. A set of VFF candidates is evaluated with (23) and the one with the minimum value will be chosen as the desired FF. The block diagram of this unit is illustrated in Fig. 3 where the grey area denotes the evaluation of (23) for a particular discretized FF. According to (23), the computation is further divided into three subunits for σ_{Σ}^2 , $\text{Tr}(\mathbf{R}_{\bar{\mathbf{y}\bar{\mathbf{y}}}}^{-1})$ and $\sigma_{\mathbf{W}}^2$, which will be described below.

1) σ_{Σ}^2 COMPUTATION SUBUNIT

From (17), σ_{Σ}^2 is recursively updated using $\|e[n]\|_2^2 = \sum_{i=1}^L e_i[n]e_i^*[n]$ which requires $2L$ multipliers and $2L - 1$ additions. Meanwhile, the constant multiplication with λ_L in $\lambda_L \hat{\sigma}_{\Sigma}^2[n-1]$ and $(1 - \lambda_L)$ in $(1 - \lambda_L)\|e[n]\|_2^2$ can be realized using the SOPOT coefficients shown in Table 6 as a series of additions and shifts. The block diagram of this subunit is illustrated in Fig. 4.

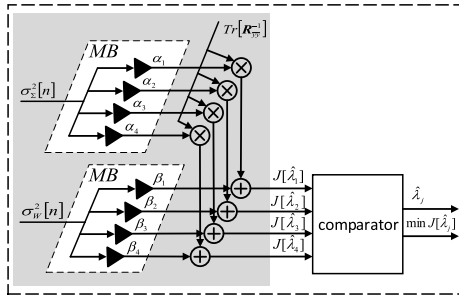


FIGURE 3. Structure of variable forgetting factor unit, where $\alpha_j = (1 - \hat{\lambda}_j)/(1 + \hat{\lambda}_j)$, $\beta_j = \hat{\lambda}_j(1 - \hat{\lambda}_j)^{-2}$ and $\hat{\lambda}_j$, for $j = 1, \dots, 4$. MB: multiplier block.

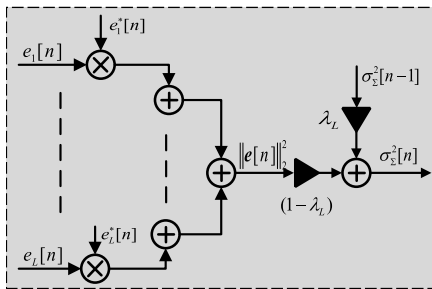


FIGURE 4. Structure of σ_w^2 computation subunit.

2) $\text{Tr}(\mathbf{R}_{yy}^{-1})$ COMPUTATION SUBUNIT

The subunit is based on the proposed division-free approach where the quantization operation in (25) is performed by a quantizer, where the input is successively compared with a set of thresholds arranged in a tree-like structure to determine which region it belongs to. Once the desired region or interval is determined, the pre-stored reciprocal \hat{s}_j^{-1} can be forwarded to a multiplier for approximating the division. In contrast to a division, it only requires $\lceil \log_2 q \rceil$ comparisons, where q is the total number of discrete levels. As DSP blocks in FPGA device is very valuable, our division-free method using the look-up-table to replace the division will result in considerable saving of DSPs. For uniform quantization, the quantizer can be further simplified to the structure in Fig. 5. The diagonal values of the approximated \mathbf{R}_{yy}^{-1} are sent to an adder tree for accumulation.

3) σ_w^2 COMPUTATION SUBUNIT

From (17)-(20), we can see that the computation of σ_w^2 and $\hat{\sigma}_{\Sigma_S}^2$ has the same computation operations with $\hat{\sigma}_{\Sigma_{L}}^2$. Consequently, the structure for (IV-B1) can also be used for computing σ_w^2 and $\hat{\sigma}_{\Sigma_S}^2$ in (IV-B3). Due to the page limitation, the block diagram of this subunit is omitted.

C. VARIABLE REGULARIZATION UNIT

From (22), we can see that the value of $\kappa[n]$ is directly dependent on $\lambda[n]$. Forming the term $\sqrt{\sigma_{\Sigma}^2 \sigma_y^2}$ requires a multiplier for computing the product of σ_{Σ}^2 and σ_y^2 and a square root operation. The term $\sqrt{\hat{y}_j^H P}$ can be precomputed and represented as SOPOT coefficient, which is presented

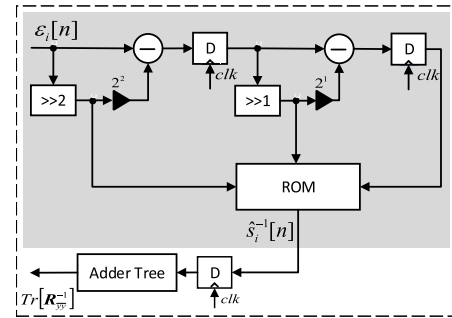


FIGURE 5. Structure of computation of $\text{Tr}(\mathbf{R}_{yy}^{-1})$ with $q = 8$. D: register; \gg right shift; \ll left shift.

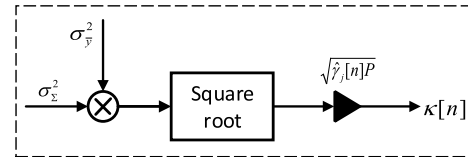


FIGURE 6. Structure of computation of $\kappa[n]$.

in Table 6, to simply its multiplication with $\sqrt{\sigma_{\Sigma}^2 \sigma_y^2}$. The block diagram of this unit is illustrated in Fig. 6.

D. QRD-RLS WEIGHT EXTRACTION UNIT

The Givens rotation implementation of QRD is attractive for its efficient parallel hardware implementation [36]–[38] using say the CORDIC algorithm. For the conventional implementation of complex-valued QRD, the complex Givens rotation, $Q[n]$, is employed to zero out the lower left row of the matrix on the right hand side of (*) in Table 3 so as to restore the upper triangular matrix, $\mathbf{R}[n]$, on the left hand side. In particular, the (1, 1) element of $\sqrt{\lambda}[n]\mathbf{R}[n-1]$ is used to zero out the first element of $\hat{\mathbf{y}}^H[n] + \sqrt{\kappa}[n]P\hat{\mathbf{d}}_i^H[n]$. Similarly, the (2, 2) element of $\sqrt{\lambda}[n]\mathbf{R}[n-1]$ is used to zero out the 2nd element of the rotated row of $\hat{\mathbf{y}}^H[n] + \sqrt{\kappa}[n]P\hat{\mathbf{d}}_i^H[n]$. The process is repeated until the entire row is zeroed out. In summary, the QRD starts with the 1st row, and use its first element with the newly appended row to compute the corresponding rotation and apply it to the remaining elements of the two rows. This is then repeated and for the j -th step, it takes in the j -th row and the last row and uses its leading nonzero coefficients to determine the rotation and apply it to the remaining elements.

The weight vector can be solved from $\mathbf{W}^H[n] = \mathbf{R}^{-1}[n]\mathbf{U}[n]$ by back-substitution using a divider, which results in slow working speed and high hardware consumption. An alternative method based on the extended QRD-RLS algorithm [13] [39] can provide a fully concurrent computation for the weight extraction as (26), shown at the bottom of the next page, and

$$\mathbf{w}[n] = \mathbf{w}[n-1] - \mathbf{g}^H[n]\mathbf{c}[n]. \quad (27)$$

This algorithm can be easily implemented with a double triangular systolic array [13] [38], which is illustrated in Fig. 7.

In our proposed structure, an alternative complex Givens rotation matrix [33], namely three angle complex rota-

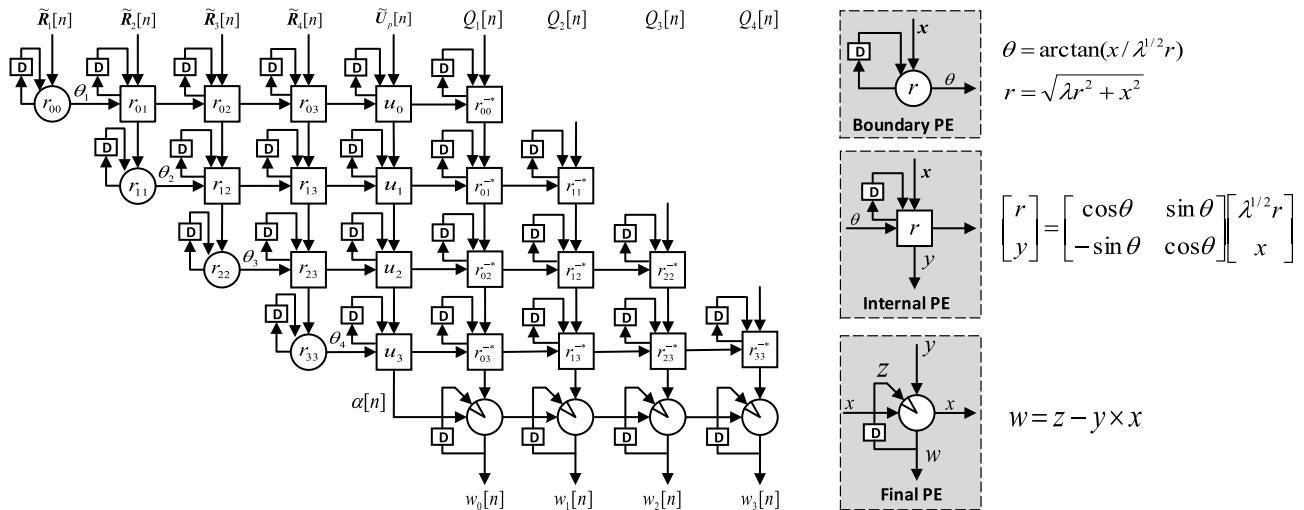


FIGURE 7. The structure of a double triangular systolic array for QRD-RLS weight extraction, where $\tilde{R} = [\tilde{R}_1, \tilde{R}_2, \dots, \tilde{R}_L]$, $L = 4$; $\tilde{U} = [\tilde{U}_1, \tilde{U}_2, \dots, \tilde{U}_P]$, $P = 1, \dots, P$.

tion (TACR), is used, which provides significant reduction of latency. In the TACR method, two Givens rotations are operated sequentially in two stages, and each stage has a similar hardware implementation. The basic idea of TACR-based QRD can be illustrated by an example of 2×2 complex-valued matrix A where we wish to zero out the lower-left element of the matrix by Givens rotation:

$$A = \begin{bmatrix} x_1 e^{j\theta_{x_1}} & x_2 e^{j\theta_{x_2}} \\ y_1 e^{j\theta_{y_1}} & y_2 e^{j\theta_{y_2}} \end{bmatrix}, \quad (28)$$

where $j = \sqrt{-1}$, x_1, x_2, y_1, y_2 are the magnitudes and $\theta_{x_1}, \theta_{x_2}, \theta_{y_1}, \theta_{y_2}$ are the angles of the complex entries in A . In particular, the complex Givens rotation matrix can be expressed as

$$G = \begin{bmatrix} \cos(\theta_1) e^{-j\theta_{x_1}} & -\sin(\theta_1) e^{-j\theta_{y_1}} \\ \sin(\theta_1) e^{-j\theta_{x_1}} & \cos(\theta_1) e^{-j\theta_{y_1}} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \cdot \begin{bmatrix} e^{-j\theta_{x_1}} & 0 \\ 0 & e^{-j\theta_{y_1}} \end{bmatrix}. \quad (29)$$

The computation process is performed in two stages. In the first stage, it converts the leading-column entries to real-valued elements as

$$\begin{bmatrix} e^{-j\theta_{x_1}} & 0 \\ 0 & e^{-j\theta_{y_1}} \end{bmatrix} \cdot \begin{bmatrix} x_1 e^{j\theta_{x_1}} & x_2 e^{j\theta_{x_2}} \\ y_1 e^{j\theta_{y_1}} & y_2 e^{j\theta_{y_2}} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 e^{j(\theta_{x_2} - \theta_{x_1})} \\ y_1 & y_2 e^{j(\theta_{y_2} - \theta_{y_1})} \end{bmatrix}. \quad (30)$$

Then, the real-valued Givens rotation matrix is applied in the second stage to introduce a zero at the desired position

$$\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 e^{j(\theta_{y_1} - \theta_{x_1})} \\ y_1 & y_2 e^{j(\theta_{y_2} - \theta_{x_2})} \end{bmatrix} = \begin{bmatrix} \sqrt{x_1^2 + y_1^2} & x_2' e^{j\theta_{x_2}'} \\ 0 & y_2' e^{j\theta_{y_2}'} \end{bmatrix}. \quad (31)$$

From (29), we can see that the TACR method converts the diagonal elements in real numbers except the one at the lowest position, which implies that it requires one additional rotation. This operation is illustrated in Fig. 8.

The required Givens rotation in the TACR can be efficiently implemented based on the CORDIC algorithm mentioned earlier. The CORDIC-based QRD hardware structure for processing each pair of input rows has its leading element called vector normalization mode (NM), which determines the rotation angle θ . Then, this angle will be applied to other CORDIC elements operating in the vector rotation mode (RM) in parallel for rotating the remaining elements in the two rows. This operation can be pipelined and it suggests an implementation in form of double triangular systolic array as shown in Fig. 7. The fully pipelined triangular CORDIC-based QRD will require $(P+1)P/2$ elements which can be area and resource intensive for a large P . To reduce the hardware resources, one can employ a linear pipelined CORDIC structure for processing two rows at a time and use it to process the N pairs of rows consecutively. This provides

$$\begin{bmatrix} \mathbf{R}[n] & \mathbf{U}[n] & \mathbf{R}^{-H}[n] \\ \mathbf{0}^H & c[n] & \mathbf{g}[n] \end{bmatrix} = \mathbf{Q}[n] \cdot \begin{bmatrix} \sqrt{\lambda[n]} \mathbf{R}[n-1] & \sqrt{\lambda[n]} \mathbf{U}[n-1] & (\sqrt{\lambda[n]})^{-1} \mathbf{R}^{-H}[n-1] \\ \tilde{\mathbf{y}}^H[n] + \sqrt{\kappa[n]} \mathbf{P} a_i^H[n] & \mathbf{x}^H[n] & c_1 \end{bmatrix} = \mathbf{Q}[n] \cdot \begin{bmatrix} \tilde{\mathbf{R}}_{(L+1) \times L}[n-1] & \tilde{\mathbf{U}}_{(L+1) \times P}[n-1] & (\sqrt{\lambda[n]})^{-1} \mathbf{R}^{-H}[n-1] \\ & & c_1 \end{bmatrix} \quad (26)$$

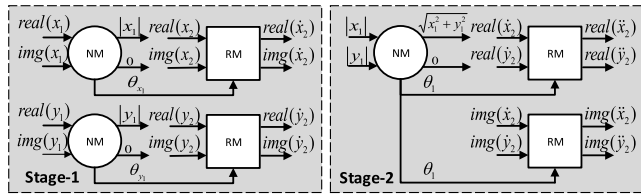


FIGURE 8. QRD structure based on two-stage TACR method for a 2×2 matrix.

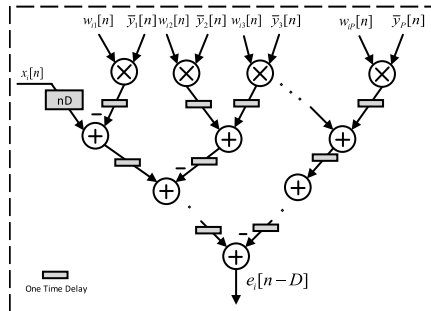


FIGURE 9. The block diagram for error computation, where nD : n delays.

another tradeoff between hardware resources and processing speed and is attractive for lower speed applications.

Yet another method to reduce the arithmetic/hardware complexity is to use a small value of P and then use multiple stages of deflation, say R , to compute a subspace of size RP . In this case, the above order- P CORDIC-QRD PAST can be cascaded to yield very high throughput with modest complexity at the expenses of slightly larger error and slower convergence speed due to error accumulation during the deflation process. The resulting hardware and arithmetic complexity is of the order $R(P + 1)P/2$, rather than $(RP + 1)RP/2$ if the conventional PAST is used.

E. ERROR VECTOR COMPUTATION UNIT

This unit computes the error vector $e[n] = x[n] - W[n]\bar{y}[n]$, where each inner product can be realized using L complex multipliers and an adder tree is used to accumulate the final result as shown in Fig. 9. The small grey rectangle stands for one time delay.

F. DEFLATION COMPUTATION UNIT

The unit computes the deflation operation $x^{(r)}[n] = x^{(r-1)}[n] - w^{(r)}[n]\bar{y}^{(r-1)}[n]$ and its structure of this unit for $x_i^{(r)}[n]$ is illustrated in Fig. 10.

V. SIMULATION AND IMPLEMENTATION RESULTS

Simulation results are now presented to evaluate the performance of the proposed QRD-based VFF-VR-PAST-MD, VFF-PAST-MD, and other conventional work [31] in DOA estimation and tracking. Simulations are performed on a ULA with 10 sensors separated by half wavelength. Both stationary and dynamic environments are investigated. Unless specified otherwise, all results are averaged over 100 Monte-Carlo simulations. A Verilog described fixed-point VFF-VR-PAST-MD is also designed based on

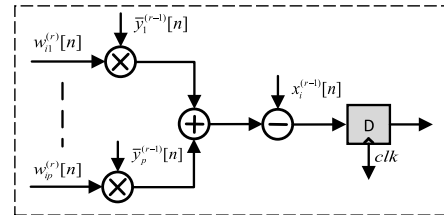


FIGURE 10. The deflation computation unit for $x_i^{(r)}[n]$.

our proposed architecture to evaluate the performance of the hardware implementation. The SOPOT coefficients of the parameters used in our simulation as presented in Table 6. This architecture has been simulated and synthesized using Xilinx ISE 14.7 and successfully implemented on Xilinx Vertex 7 (XC7VX980T) FPGA, and the implementation results are shown in Table 5. Considering the tradeoff between hardware resource and system performance from our simulation result to be presented later, the wordlength for the QRD-based VFF-VR-PAST-MD algorithm is chosen as 22 bits. The resultant pipelined implementation achieves an impressive maximum working speed of 143 MHz at 5 MHz sampling rate for different values of P . If the proposed VFF-PAST-MD approach is implemented using multipliers and dividers (denoted by VFF-CVR-PAST-MD where C is referred to as combined update) with the same P value, our proposed hardware architecture can achieve around 20.2% higher working speed and save 1.9% LUTs, 1.8% Slice Register, and 22.8% DSPs, respectively. On the other hand, the full implementation of the proposed VFF-VR-PAST-MD architecture can achieve the same working speed with our proposed one, it will double the latency at the QRD update, which leads to decreased system throughput. From the implementation results, we can also see the VR scheme only requires a slight increase in hardware, but it can improve considerably the system robustness. In summary, the proposed architecture offers high throughput rate and different tradeoffs between hardware resources and performances, which will be further elaborated in the following section.

A. DOA ESTIMATION IN STATIONARY ENVIRONMENTS

In this simulation study, four uncorrelated narrow band signals located at directions $\theta_1 = 8^\circ$, $\theta_2 = 20^\circ$, $\theta_3 = 45^\circ$ and $\theta_4 = 60^\circ$ with equal power are considered. The short-term and long-term FFs are chosen as 0.9 and 0.99, respectively and the upper bound for the regularization parameter is set at 0.2. The confidence factor χ is chosen as 4, which corresponds to 99.99% confidence that a system change has been detected. The number FF candidates is $F = 4$.

The FF of the PAST-MD is set to 0.98. The average DOA deviation and the root mean squared error (RMSE) are used to evaluate the performance of different algorithms

$$RMSE = \sqrt{\sum_{i=1}^{K_M} \sum_{n=1}^K (\theta_n - \hat{\theta}_{i,n})^2 / (K_M K)}, \quad (32)$$

TABLE 5. Proposed QRD-based VFF-PAST-MD architectures implementation results, $R=4$.

Architecture	Proposed VFF-CVR-PAST-MD ($P=4$)	Proposed VFF-CVR-PAST-MD ($P=2$)	Conventional VFF-CVR-PAST-MD ($P=2$)	Proposed VFF-VR-PAST-MD ($P=2$)	Proposed VFF-PAST-MD ($P=2$)
LUTs	88884	68224	69542	69710	69430
Slice Register	137522	110848	112896	113044	112798
DSPs	1873	1308	1694	1308	1290
Working Frequency with Sampling Rate of 5 MHz	143 MHz	143 MHz	119 MHz	143 MHz	143 MHz

TABLE 6. SOPOT coefficients of the parameters in QRD-based VFF-VR-PAST algorithm.

Parameter	Value	SOPOT Representation	
λ_L	0.99	$2^0 - 2^{-7} - 2^{-9} - 2^{-12}$	
$\lambda_S = \lambda_p$	0.9	$2^0 - 2^{-4} - 2^{-5} - 2^{-8} - 2^{-9} - 2^{-11}$	
λ_1	0.99	$2^0 - 2^{-7} - 2^{-9} - 2^{-12}$	
λ_2	0.97	$2^0 - 2^{-5} - 2^{-10} - 2^{-12}$	
λ_3	0.95	$2^0 - 2^{-5} - 2^{-6} - 2^{-9} - 2^{-10} - 2^{-12}$	
λ_4	0.93	$2^0 - 2^{-4} - 2^{-7} - 2^{-12}$	
$P = 2$	$\sqrt{\gamma_1/P}$	0.0162	$2^{-6} + 2^{-11}$
	$\sqrt{\gamma_2/P}$	0.0286	$2^{-5} - 2^{-9} - 2^{-11} - 2^{-12}$
	$\sqrt{\gamma_3/P}$	0.0338	$2^{-5} + 2^{-9} + 2^{-11}$
	$\sqrt{\gamma_4/P}$	0.0366	$2^{-5} - 2^{-8} - 2^{-10} - 2^{-11}$
$P = 4$	$\sqrt{\gamma_1/P}$	0.0081	$2^{-7} + 2^{-12}$
	$\sqrt{\gamma_2/P}$	0.0143	$2^{-6} - 2^{-10} - 2^{-12}$
	$\sqrt{\gamma_3/P}$	0.0169	$2^{-6} + 2^{-10} - 2^{-12}$
	$\sqrt{\gamma_4/P}$	0.0183	$2^{-6} + 2^{-9} + 2^{-11} + 2^{-12}$

TABLE 7. Average deviations of different number of discrete levels.

Number of Levels	4	8	16	32	64
Average Deviation ($\times 10^{-4}$)	4.39	0.74	0.62	0.61	0.58

where K_M and K denote respectively the number of Monte-Carlo runs and the number of signals, θ_n represents the n -th DOA and $\hat{\theta}_{i,n}$ is the n -th estimated DOA in the i -th Monte-Carlo simulation.

1) EXPERIMENT 1

This experiment is carried out to evaluate the performance of the VFF-VR-PAST-MD algorithm with different number of discrete levels q for $\sigma_{y_p}^2[n]$ for $P = 4$ and $L = 10$ in a stationary environment. The deviation between the floating-point division and division-free approaches for $q = 4, 8, 16, 32$ and 64 are shown in Fig. 11. It can be seen that when q is larger than 16, the results are generally satisfactory. The average deviations over the stimulated horizon as shown in Table 7 also reveals a similar observation. Considering the hardware resource required and operation speed, $q = 16$ is adopted in the subsequent experiments. For an L -length sensor antenna, simulated experiments show that $2^{\lceil \log_2 2L \rceil}$ or $2^{\lceil \log_2 4L \rceil}$ yields reasonable approximation where $\lceil a \rceil$ denotes the nearest integer smaller than a .

2) EXPERIMENT 2

This experiment is conducted to determine the appropriate wordlength, including the integer bit and fractional bit,

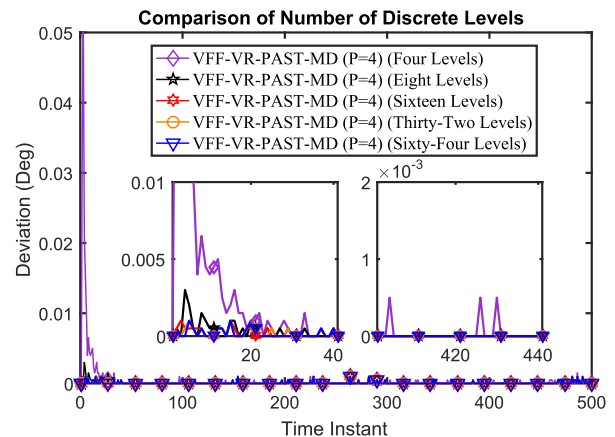


FIGURE 11. Comparison of number of discrete levels.

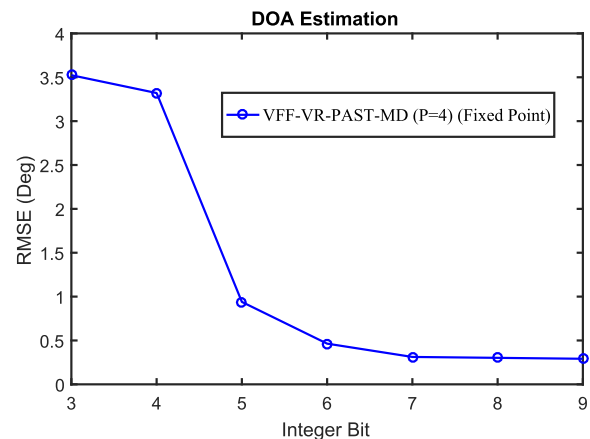


FIGURE 12. Comparison of RMSEs under different integer bits.

for hardware implementation. The VFF-VR-PAST-MD algorithm with $P = 4$ and $L = 10$ is utilized to compare the performance. Firstly, we shall determine the integer bit. Given sufficient fractional bit, say, 20 bits, the RMSEs under different integer bits are shown in Fig. 12. It can be seen that the RMSE decreases with increasing integer bit. The gap between adjacent integer bits used is also decreasing and the differences after 7 bits are much smaller than before. Therefore, the integer bit is chosen to be 7 bits. Fig. 13 shows the RMSEs under different fractional bits used. As expected, the RMSEs decreases significantly initially when the fractional bits used increases from a relatively small value and then gradually level off after 14 bits. Hence, a fractional part of 14 bits is adopted to guarantee a sufficiently low RMSE. Thus, including the additional sign bit, a total

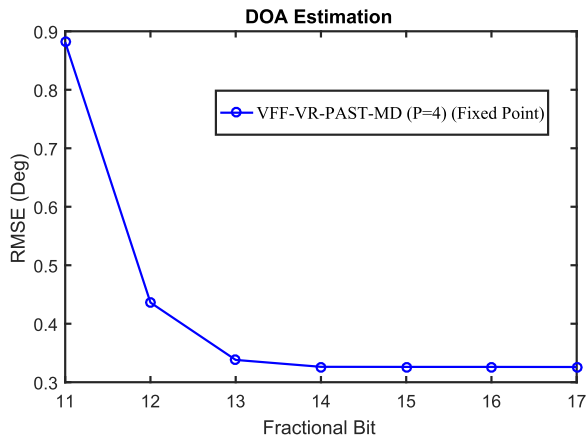


FIGURE 13. Comparison of RMSEs under different fractional bits.

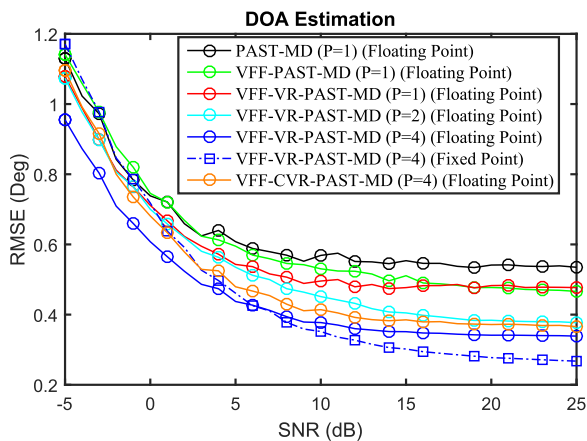


FIGURE 14. Comparison of DOA estimation accuracy in terms of RMSEs for different methods under different SNRs in stationary environment. The fixed-point implementation uses a 22-bit wordlength and $q = 16$ levels.

wordlength of 22 bits is used in our simulation and hardware implementation.

3) EXPERIMENT 3

This experiment examines the DOA estimation accuracy of the proposed algorithm under different single-to-noise ratios (SNRs). A set of SNR levels ranging from -5 dB to 25 dB are recorded at the 800-th snapshot, where the algorithms have been converged. Fig. 14 illustrates the DOA estimation results of different methods under different SNRs in stationary environment. It is seen that the VFF-PAST-MD algorithm performs better than the PAST-MD at high SNR levels, while they are comparable at low SNR levels. The proposed VR scheme further improves the performance at low SNRs. As for VFF-VR-PAST-MD, larger P results in better performance both in high and low SNRs. With the increase of SNR, the superiority gradually increases and the RMSE decreases. The performance degradation due to the CVR QRD is acceptable with only half the arithmetic complexity. The performance of the 22-bit fixed-point implementation with $q = 16$ has a slightly inferior initial convergence speed as its floating point counterpart. This is due

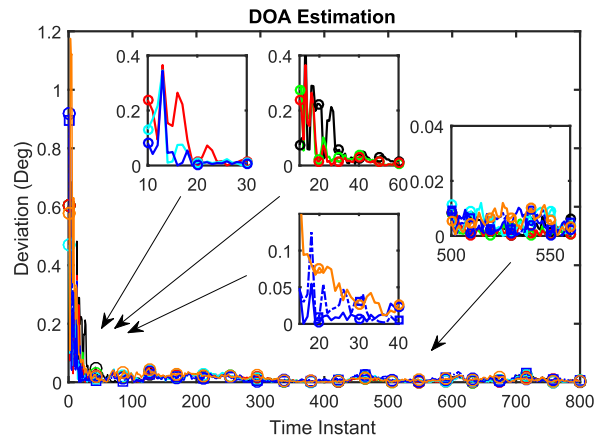


FIGURE 15. Comparison of estimated DOA deviation from the ground-truth using different methods under 5 dB SNR in stationary environment. The algorithms are labeled using the same color scheme as in Fig. 14.

TABLE 8. Convergence performance of different methods.

PAST-MD ($P = 1$)	PAST	VFF-PAST	VFF-VR-PAST
Convergence Time (number of iterations)	27	23	22
VFF-VR-PAST-MD	$P = 1$	$P = 2$	$P = 4$
Convergence Error	0.0077	0.0074	0.0041

to the quantization of $\text{Tr}(\mathbf{R}_{yy}^{-1})$ which slow down the update of the FF. On the other hand, the variation of the FF at the steady-state is reduced and hence its steady-state performance is better than its floating point counterpart.

4) EXPERIMENT 4

This experiment inspects the convergence performance of different methods at 5dB SNR. Fig. 15 shows the average deviation between DOAs of the four designated signals and the estimated looking directions. To better illustrate the convergence speed, Table 8 displays the convergence time, which is denoted by the point after which the deviation is no more than 10% of the average maximum deviation value of the given method. It can be seen that the VFF-based algorithms converge faster than the corresponding PAST-MD algorithm with a constant FF. The VFF-VR-PAST-MD algorithm converges slightly better than the VFF-PAST-MD algorithm. The averaged error after convergence which is referred to as the “convergence error” gradually decreases with increasing values of P . The performance of CVR update is still comparable with the full update with only slight degradation. The performances in floating-point arithmetic and fixed point arithmetic for VFF-VR-PAST-MD algorithm with $P = 4$ are similar.

B. DOA TRACKING IN DYNAMIC ENVIRONMENTS

The case of four uncorrelated narrow band signals with equal power is again considered. However, two sources are assumed to be invariant and are located at directions $\theta_1 = 8^\circ$ and

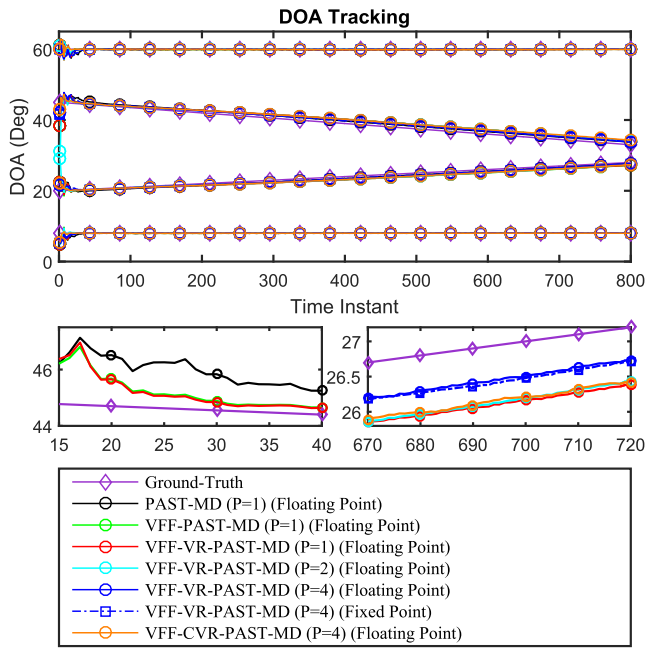


FIGURE 16. Comparison of DOA tracking using different methods under 5 dB SNR for time-varying DOAs.

$\theta_4 = 60^\circ$, while the other two signals are time varying. All the settings for various methods are identical to the stationary experiment.

1) EXPERIMENT 1

In this experiment, the DOA tracking performance of the proposed method is evaluated in nonstationary cases under SNR with 5 dB. θ_2 and θ_3 are assumed to be linearly and slowly varying as follows

$$\theta_2 = 20 - 1 \times 10^{-2}t, 0 \leq t \leq 800,$$

$$\theta_3 = 45 - 1.5 \times 10^{-2}t, 0 \leq t \leq 800. \quad (33)$$

Fig. 16 depicts the tracking results of different methods for both invariant and time-varying DOAs. It is seen that for $P = 1$, both VFF-PAST-MD and VFF-VR-PAST-MD algorithms outperform PAST-MD method with smaller tracking errors. Moreover, VR-based algorithms can achieve a slightly better performance. Larger value of P yields a result closer to the ground truth. Although the tracking performance of VFF-CVR-PAST-MD with $P = 4$ suffers from slight degradation, it is still comparable to the VFF-VR-PAST-MD algorithms with $P = 1$ or $P = 2$. The fixed-point implementation of the proposed approach is reasonably close to its floating-point counterpart.

2) EXPERIMENT 2

To further verify the performance of the proposed VFF-VR-PAST-MD algorithm, we carry out an experiment with signal fading in which a short period with low signal power will be

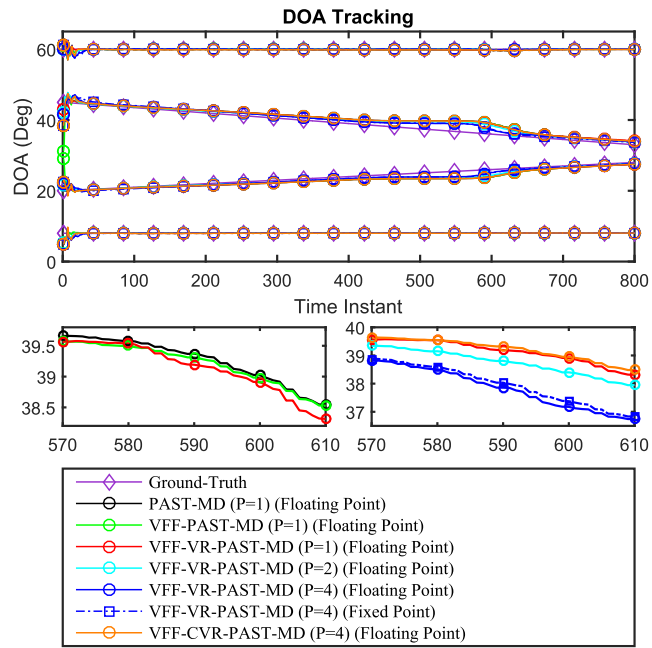


FIGURE 17. Comparison of DOA tracking using different methods under 5 dB SNR for time-varying DOAs with signal fading.

experienced. The settings of the four sources are the same as the last experiment. For illustration, the amplitudes of the signals are given by

$$A(t) = \begin{cases} A_0, & 0 \leq t < 400 \\ [1 - 0.9(t - 400)/80]A_0, & 400 \leq t < 480 \\ 0.1A_0, & 480 \leq t < 560, \\ [0.1 + 0.9(t - 560)/80]A_0, & 560 \leq t < 640 \\ A_0, & 640 \leq t \leq 800 \end{cases} \quad (34)$$

where A_0 is the original signal amplitude.

Fig. 17 shows the resultant performance of the various algorithms. VR-based algorithms generally offer improved performances over the VFF and fixed FF algorithms due to reduced estimation error variance. VFF-VR-PAST-MD with larger P results in smaller tracking deviation from the ground truth. The CVR-QRD also experienced acceptable performance degradation. The performances between floating-point arithmetic and fixed-point arithmetic are similar for VFF-VR-PAST-MD algorithm with $P = 4$. In summary, the advantages of this MD algorithm is that 1) it can be used to adaptively estimate the total subspace dimension by monitoring the residuals at each stage. 2) it can be implemented in a pipelined manner which is very efficient and scalable as it only requires the cascade of a core VFF-VR-PAST module of a given size P , which can be easily optimized rather than the original PAST or VFF-VR-PAST algorithms which involves a complexity which grows with $\mathcal{O}((RP)^2)$, where R is the number of deflation stages which may be

unknown in practice. This makes the hardware optimization rather difficult. 3) it has a better performance than the PAST-d algorithm with slight increase in hardware complexity, thus offer a more flexible tradeoff between hardware resources and performance.

VI. CONCLUSION

A new VFF-VR-PAST-MD algorithm for subspace tracking and its related efficient hardware architecture have been presented. The VFF and VR schemes improve the convergence speed, steady-state error and stability of the conventional PAST algorithm. Novel simplifications of the VFF-VR-PAST algorithm are also proposed to avoid the cubic root and division operations involved to facilitate its hardware implementation. A novel pipelined hardware implementation of the simplified VFF-VR-PAST-MD algorithm employing the TACR-based QRD is developed. The proposed QRD-based VFF-VR-PAST-MD architecture is successfully synthesized and implemented in FPGA with a reduced hardware resources and higher operating speed than conventional approaches. The proposed algorithms offer better performance and a more flexible tradeoff between hardware resources and performance. The efficient architecture can also be applied to real-time scenarios including adaptive subspace identification, digital communication, and etc.

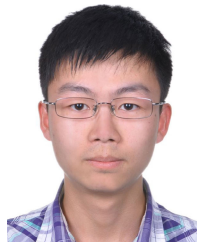
REFERENCES

- [1] S. Affes and Y. Grenier, "A signal subspace tracking algorithm for microphone array processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 5, pp. 425–437, Sep. 1997.
- [2] G. Wang, J. Xin, J. Wang, N. Zheng, and A. Sano, "Subspace-based two-dimensional direction estimation and tracking of multiple targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1386–1402, Apr. 2015.
- [3] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. Signal Process.*, vol. 43, no. 2, pp. 516–525, Feb. 1995.
- [4] Z.-M. Liu, Z.-T. Huang, Y.-Y. Zhou, and J. Liu, "Direction-of-arrival estimation of noncircular signals via sparse representation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2690–2698, Jul. 2012.
- [5] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, Jul. 1995.
- [6] B. Liao, Z.-G. Zhang, and S.-C. Chan, "DOA estimation and tracking of ULAs with mutual coupling," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 891–905, Jan. 2012.
- [7] S. Min, D. Seo, K. B. Lee, H. M. Kwon, and Y.-H. Lee, "Direction-of-arrival tracking scheme for DS/CDMA systems: Direction lock loop," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 191–202, Jan. 2004.
- [8] Y. Shi, L. Huang, C. Qian, and H. C. So, "Direction-of-arrival estimation for noncircular sources via structured least squares-based esprit using three-axis crossed array," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1267–1278, Apr. 2015.
- [9] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.
- [10] R. Kumaresan and D. W. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 1, pp. 134–139, Jan. 1983.
- [11] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.
- [12] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1110–1121, May 1991.
- [13] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.
- [14] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2931–2941, Aug. 2005.
- [15] X. G. Doukopoulos and G. V. Moustakides, "Fast and stable subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1452–1465, Apr. 2008.
- [16] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. Signal Process.*, vol. 45, no. 5, pp. 1222–1240, May 1997.
- [17] S.-C. Chan, H.-J. Tan, J.-Q. Lin, and B. Liao, "A new local polynomial modeling based variable forgetting factor and variable regularized PAST algorithm for subspace tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1530–1544, Jun. 2018.
- [18] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Process. Lett.*, vol. 7, no. 3, pp. 60–62, Mar. 2000.
- [19] S. Ouyang and Y. Hua, "Bi-iterative least-square method for subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2984–2996, Aug. 2005.
- [20] P. Strobach, "The fast householder bi-SVD subspace tracking algorithm," *Signal Process.*, vol. 88, no. 11, pp. 2651–2661, Nov. 2008.
- [21] R. Badeau, G. Richard, and B. David, "Fast and stable YAST algorithm for principal and minor subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3437–3446, Aug. 2008.
- [22] T. Adali and S. Haykin, *Adaptive Signal Processing: Next Generation Solutions*, vol. 55. Hoboken, NJ, USA: Wiley, 2010.
- [23] T. R. Fortescue, L. S. Kershenbaum, and B. E. Ydstie, "Implementation of self-tuning regulators with variable forgetting factors," *Automatica*, vol. 17, no. 6, pp. 831–835, Nov. 1981.
- [24] S.-H. Leung and C. F. So, "Gradient-based variable forgetting factor RLS algorithm in time-varying environments," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3141–3150, Aug. 2005.
- [25] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Process. Lett.*, vol. 15, pp. 597–600, 2008.
- [26] A. K. Kohli and A. Rai, "Numeric variable forgetting factor RLS algorithm for second-order volterra filtering," *Circuits, Syst., Signal Process.*, vol. 32, no. 1, pp. 223–232, Feb. 2013.
- [27] S. C. Chan, Y. J. Chu, Z. G. Zhang, and K. M. Tsui, "A new variable regularized QR decomposition-based recursive least M-estimate algorithm—Performance analysis and acoustic applications," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 5, pp. 907–922, May 2013.
- [28] S. Erlich, "Systolic arrays for adaptive eigenstructure decomposition of correlation matrices," Ph.D. dissertation, UCLA, 1990.
- [29] J. Gotze, M. Ali, and J. A. Nossek, "A bit level algorithm for subspace tracking," in *Proc. IEEE Workshop VLSI Signal Process.*, Oct. 1993, pp. 352–360.
- [30] D. J. Evans and K. Margaritis, "Systolic designs for eigenvalue-eigenvector computations using matrix powers," *Parallel Comput.*, vol. 14, no. 1, pp. 77–87, May 1990.
- [31] F. Xu and A. N. Willson, "Efficient hardware architectures for eigenvector and signal subspace estimation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 3, pp. 517–525, Mar. 2004.
- [32] Y. Lim and S. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 3, pp. 583–591, Jun. 1983.
- [33] A. Maltsev, V. Pestretsov, R. Maslennikov, and A. Khoryaev, "Triangular systolic array with reduced latency for QR-decomposition of complex matrices," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 4–7.
- [34] Y. J. Chu and S. C. Chan, "A new local polynomial modeling-based variable forgetting factor RLS algorithm and its acoustic applications," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 11, pp. 2059–2069, Nov. 2015.
- [35] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 9, pp. 569–577, Sep. 1995.
- [36] Z.-Y. Huang and P.-Y. Tsai, "Efficient implementation of QR decomposition for gigabit MIMO-OFDM systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 10, pp. 2531–2542, Oct. 2011.
- [37] B. Yang and J. F. Bohme, "Rotation-based RLS algorithms: Unified derivations, numerical properties, and parallel implementations," *IEEE Trans. Signal Process.*, vol. 40, no. 5, pp. 1151–1167, May 1992.
- [38] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays," in *Proc. 4th Real-Time Signal Process.*, Jul. 1982, pp. 19–26.
- [39] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.



WEI ZHAO received the B.Sc. degree in measurement and instrument engineering from the Harbin University of Science and Technology, China, in 2010, the M.Sc. degree in telecommunication engineering from City University, London, U.K., in 2011, and the Ph.D. degree in digital signal processing from The University of Hong Kong, Hong Kong, in 2017. He is currently an Assistant Professor with the Institute of Electrical Engineering, Chinese Academy of Sciences, China, and the

Ganjiang Innovation Academy, Chinese Academy of Sciences. His research interests include array signal processing, adaptive filtering, digital hardware design, and embedded systems.



JIAN-QIANG LIN received the bachelor's degree in information engineering from Beihang University, Beijing, China, in 2015, and the Ph.D. degree in digital signal processing from the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, in 2020. His current research interests include array signal processing, subspace identification, adaptive filtering, and mathematics.

...



SHING-CHOW CHAN (Member, IEEE) received the B.Sc. degree in engineering and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 1986 and 1992, respectively. Since 1994, he has been with the Department of Electrical and Electronic Engineering, The University of Hong Kong, where he is currently a Professor. His research interests include fast transform algorithms, filter design and realization, multi-

rate and biomedical signal processing, communications and array signal processing, high-speed A/D converter architecture, bioinformatics, smart grid, and image-based rendering. He is also a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was the Chair for the IEEE Hong Kong Chapter of Signal Processing, during 2000–2002, and organizing committee members of 2003 *IEEE ICASSP* and 2010 *IEEE ICIP*. He is an Associate Editor of the *Journal of Signal Processing Systems*. He was an Associate Editor of *Digital Signal Processing*, the *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II*, from 2012 to 2016, and the *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I*, from 2008 to 2009.