IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 18, 2018, accepted September 27, 2018, date of publication October 11, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875409

A Division-Free and Variable-Regularized LMS-Based Generalized Sidelobe Canceller for Adaptive Beamforming and Its Efficient Hardware Realization

W. ZHAO [1], J. Q. LIN [2], S. C. CHAN [2], AND H. K.-H. SO [2]
[1]Institute of Electrical Engineering, Chinese Academy of Sciences, Beijing 100190, China
[2]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

Corresponding author: W. Zhao (wzhao@mail.iee.ac.cn)

This work was supported by the Natural Science Foundation of China under Grant 51761135014.

ABSTRACT This paper proposes a new division-free generalized sidelobe canceller-based adaptive beamformer and its efficient hardware realization. A discrete cosine transform-based blocking matrix is proposed for uniform linear array to decorrelate the input so as to achieve a faster convergence speed. A new variable step-size least mean squares algorithm for complex input is proposed to further improve the convergence and the steady-state performance of the adaptive beamformer. Moreover, a variable regularization scheme is incorporated to mitigate possible signal cancellation due to possible mismatches in steering vector. Furthermore, a statistical analysis on the mean and mean squares convergence of the algorithm is performed and validated using Monte Carlo simulations. An efficient architecture of the proposed adaptive beamformer is also proposed for its real-time implementation. It employs a novel division-free approach by quantizing the normalization factor into a limited number of levels so that the division can be implemented using canonical signed digits, resulting in multiplier-less realization. The performance of the resultant division-free implementation can achieve similar convergence and steady-state performance as a conventional divider approach while achieving at least 21% less hardware resources and 26.85% higher operating speed in Xilinx Virtex7 (XC7VX330T) field programming gate array for an eight-sensor uniform linear array. Finally, the beam can be stabilized remarkably in only 1 μs at a system clock frequency of 124 MHz.

INDEX TERMS Array processing, adaptive beamformer, variable step-size, variable regularization, approximation, division-free, FPGA.

I. INTRODUCTION

Adaptive beamformers (ABFs) using sensor arrays are frequently employed in a wide range of applications including speech and acoustic processing, radio astronomy, wireless communications [1]–[4], etc. Many adaptive beamformers have been proposed for enhancing signals at specified direction while attenuating interference from other directions. A commonly used adaptive beamformer is the minimum variance distortionless response (MVDR) beamformer [5], which minimizes the output power subject to a unity gain constraint in the looking or desired direction. More linear constraints can be incorporated through the linearly constrained minimum variance (LCMV) beamformer [6]. An equivalent implementation, where the linear constraints can be structurally imposed, can be realized using the generalized sidelobe canceller (GSC) structure [7], [8]. It transforms the minimization problem into an unconstrained one by decomposing the beamforming weights into a fixed nonadaptive and an adaptive part where conventional adaptive filtering algorithms such as the variants of least mean square (LMS) and the recursive least squares (RLS) algorithms are applicable. The RLS-based algorithms are well known for their fast convergence speed. However, they have a higher arithmetic complexity of $\mathcal{O}(N^2)$ (where N is the number of taps in the filter), as compared with $\mathcal{O}(N)$ for the conventional LMS-based algorithms. Moreover, special care has to be taken to avoid ill-conditioning of covariance matrix in the implementation of the RLS algorithms. On the other hand, LMS [9] and

normalized-LMS (NLMS) [10], [11] algorithms are widely used due to their simplicity in software and hardware implementation, despite their lower convergence speed compared with the RLS algorithm. More efficient variants and hardware structures of the LMS-like algorithms are thus continual areas of active research. For instance, by utilizing variable step-size (VSS) [11]–[15], both convergence speed and steady-state misadjustment of the LMS-like algorithms can be improved. Fractional evolutionary strategies [16]–[18], as well as, fractional adaptive signal processing algorithms based on Fractional LMS and NLMS [18]–[21] can also be applied in terms of accuracy and convergence. In [22], a variable regularized and switch mode noise constrained transformation domain normalized LMS (VR-SNC-TDNLMS) algorithm with a fast convergence speed and reduced variance in signal fading situations was proposed. In the context of ABFs, the regularization can also improve the robustness of the ABFs with mismatches in direction of arrival (DOA) [23]. This motivated us to develop an efficient hardware structure for narrowband ABFs based on the concept of the VR-SNC-TDNLMS algorithm.

One of the key obstacles in implementing such algorithm is the need for a fast divider to perform the normalization required in the NLMS or TDNLMS algorithm, which may otherwise reduce significantly the maximum throughput of the ABF. Another issue is the implementation of the blocking matrix (BM) in the adaptive part of the ABF. The Griffiths-Jim (GJ) BM is widely used as it involves only additions. However, its performance can be significantly limited due to the resulting large eigenvalue spread of the covariance matrix associated with the adaptive part of the beamformers.

In summary, there is a need to develop a high performance LMS-based algorithm and efficient hardware for ABFs for various real-time applications. To address these issues, we first extend the VR-SNC-TDNLMS to complex value under the GSC framework. We then propose to implement the BM using the multiplier-block (MB) technique [24], [25], which can be implemented with minimum number of additions rather than multipliers. Moreover, for uniform linear array (ULA), we found that a BM based on the discrete cosine transform (DCT) leads to a structure very similar to the VR-SNC-TDNLMS algorithm with considerably better performance than the GJ BM. As a uniform circular array (UCA) can also be transformed to a ULA through the phase mode [26], the proposed approach is also applicable to UCA and frequency invariant (FI) UCAs and UCCAs [27] for broadband applications. Another contribution of this paper is a division-free implementation of the VR-SNC-TDNLMS algorithm, which is also applicable to other LMS-like algorithms involving normalization. The basic idea is to divide or quantize the normalization factor, which is usually the power of the input or related quantities, to a limited number of levels so that the division of these fixed numbers can be implemented as constant multiplications. Moreover, these constant multiplications can be implemented using canonical signed digits (CSD) or sum-of-power-of-two (SOPOT) numbers [28], [29] resulting in multiplier-less realization. We also found that by properly selecting these levels, the performance of the resultant division-free VR-SNC-TDNLMS can achieve similar convergence speed with smaller weight variations, due to reduction of step-size adaptation noise resulting from continuous step-size adaptation. To verify the efficiency of the proposed approach, a pipelined architecture of the proposed ABF is proposed and implemented in field programming gate array (FPGA). As the delay time of division operation is usually much longer than that for multiplication and addition [30], while consuming more area and power, the proposed architecture leads to much higher operating frequency as well as lower hardware resources. Compared to previous works in [31] and [32], which are based on LMS algorithm, our algorithm leads to much faster convergence speed with comparable complexity. Our architecture also outperforms another hardware beamformer based on NC-LMS algorithm [33] in convergence speed and resilience to DOA mismatches.

In summary, the resultant VR-SNC GSC-based ABF possesses the following novelties and features:

- the VR SNC NLMS algorithm is extended for complex inputs so that it can be employed in array signal processing applications,
- a DCT-based blocking matrix for uniform linear array is proposed with faster convergence speed than the conventional GJ blocking matrix,
- the use of VR scheme for mitigating possible signal cancellation due to possible mismatches in steering vector,
- a statistical analysis on the mean and mean square convergence performance is presented and validated using Monte-Carlo simulations,
- an efficient architecture of the proposed ABF using a novel division-free approach and CSD is proposed, and
- the proposed pipelined architecture achieves similar convergence and steady-state performance as conventional divider approach while achieving at least 21% less hardware resources and 26.85% higher operating speed in Xilinx Virtex7 (XC7VX330T) FPGA for an 8-sensor ULA. Finally, the beam can be stabilized remarkably in only one microsecond at a system clock frequency of 124 MHz.

The rest of this paper is organized as follows. The proposed VR-SNC GSC-based ABF will be introduced and analyzed in Section II. Section III is devoted to the hardware architecture of the proposed ''division-free'' VR-SNC-TDNLMS GSC. Computer simulation and comparison with other conventional techniques will be presented in Section IV. The hardware realization of the proposed ''division-free'' VR-SNC-TDNLMS GSC in FPGA and a brief comparison with conventional approaches will also be presented. Finally, conclusions are drawn in section V.

II. ADAPTIVE GSC BEAMFORMER

A. DCT-BASED GSC BEAMFORMER WITH ARRAY DATA MODEL

Consider an antenna array with L sensors impinged by $K+1$ narrow-band uncorrelated signals including one signal-of-interest (SOI) or source signal and K interferences. The narrow-band signal received by the antenna array at the n-th snapshot can be written as

$$\boldsymbol{x}(n) = \boldsymbol{s}(n) + \boldsymbol{i}(n) + \boldsymbol{\zeta}(n), \tag{1}$$

where $\boldsymbol{x}(n) = [x_1(n), x_2(n), ..., x_L(n)]^{\mathsf{T}}$, $\boldsymbol{s}(n) = [s_1(n), s_2(n), ..., s_L(n)]^{\mathsf{T}}$, $\boldsymbol{i}(n) = [i_1(n), i_2(n), ..., i_L(n)]^{\mathsf{T}}$ and $\boldsymbol{\zeta}(n) = [\zeta_1(n), \zeta_2(n), ..., \zeta_L(n)]^{\mathsf{T}}$ are respectively the SOI, interference and sensor noise vectors, $(\cdot)^{\mathsf{T}}$ is the matrix transposition and the subscript "i" denotes the component at the i-th sensor. The sensor noise is assumed to be zero mean white circular symmetric complex Gaussian noise with covariance $\mathbb{E}[\boldsymbol{\zeta}(n)\boldsymbol{\zeta}^{\mathsf{H}}(n)] = \sigma_\zeta^2\boldsymbol{I}$, where $(\cdot)^{\mathsf{H}}$ is the Hermitian transpose. Let θ_0 and θ_k , $k = 1, ..., K$, be respectively the direction-of-arrivals (DOAs) of the source and interference, and their corresponding steering vectors be presented by $\boldsymbol{a}(\theta_0)$ and $\{\boldsymbol{a}(\theta_k)\}_{k=1}^K$. For a uniform linear array (ULA), the steering vector can be represented as

$$\boldsymbol{a}(\theta) = [1, e^{j2\pi fc^{-1}d\sin(\theta)}, ..., e^{j2\pi fc^{-1}(L-1)d\sin(\theta)}]^{\mathsf{T}}, \tag{2}$$

where f, c, d and θ are the carrier frequency, propagation velocity, inter-sensor spacing and DOA, respectively. In MVDR or Capon beamformer, the ABF $\boldsymbol{w}(n)$ is chosen to minimize the power of the output

$$y(n) = \boldsymbol{w}^{\mathsf{H}}(n)\boldsymbol{x}(n), \tag{3}$$

subject to a unity gain constraint on the desired steering vector $\boldsymbol{a}(\theta_0)$, which can be formulated as the following optimization problem

$$\min_{\boldsymbol{w}} \mathbb{E}[|\boldsymbol{w}^{\mathsf{H}}\boldsymbol{x}(n)|^2] \quad \text{s.t.} \quad \boldsymbol{w}^{\mathsf{H}}\boldsymbol{a}(\theta_0) = 1, \tag{4}$$

where $\mathbb{E}[\cdot]$ denotes mathematical expectation and the last constraint ensures that the gain in the looking direction θ_0 is equal to one. The optimal solution to (4) is given by

$$\boldsymbol{w}_{opt} = (\boldsymbol{a}^{\mathsf{H}}(\theta_0)\boldsymbol{R}_{xx}^{-1}\boldsymbol{a}(\theta_0))^{-1}\boldsymbol{R}_{xx}^{-1}\boldsymbol{a}(\theta_0), \tag{5}$$

where $\boldsymbol{R}_{xx} = \mathbb{E}[\boldsymbol{x}(n)\boldsymbol{x}^{\mathsf{H}}(n)]$ is the covariance matrix of the sensor input vector, which can be estimated by its ensemble average $\hat{\boldsymbol{R}}_{xx} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}(n)\boldsymbol{x}^{\mathsf{H}}(n)$ using say N snapshots.

In the GSC, the optimal solution is decomposed into two components, one in the subspace of the constraint and the other orthogonal to it, which yields

$$\boldsymbol{w}_{opt} = \boldsymbol{w}_q - \boldsymbol{B}^{\mathsf{H}}\boldsymbol{w}_a, \tag{6}$$

where $\boldsymbol{w}_q = \boldsymbol{a}(\theta_0)(\boldsymbol{a}^{\mathsf{H}}(\theta_0)\boldsymbol{a}(\theta_0))^{-1}$ is the fixed beamformer to satisfy the constraint while $\boldsymbol{w}_a$ is the adaptive weight in a subspace spanned by the $(L-1)\times L$ BM $\boldsymbol{B}$, which is orthogonal to the constraint. Consequently, $\boldsymbol{B}$ has to satisfy $\boldsymbol{B}\boldsymbol{a}(\theta_0) = \boldsymbol{0}$. The basic block diagram of the GSC is illustrated in Fig. 1 and

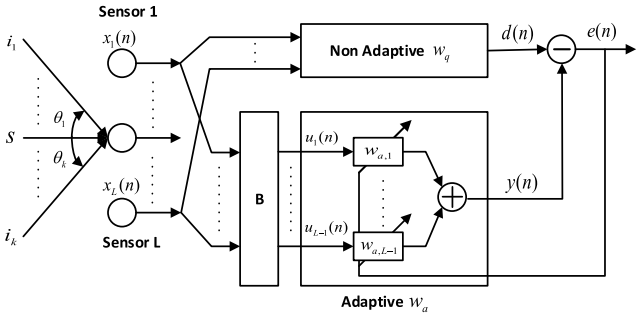


FIGURE 1. Basic block diagram of adaptive GSC beamformer.

one can see that the signal in the looking direction is eliminated by the BM $\boldsymbol{B}$ due to the orthogonality constraints so that only the interference-plus-noise will appear at the output of the BM. The adaptive weight $\boldsymbol{w}_a$ is then used to cancel out the signal-plus-interference component at the output of the fixed beamformer by minimizing the output power of ABF. Usually, the DOA of the source signal, θ_0, is estimated and is assumed to be known. To simplify the design of the BM, which in principle depends on the looking direction θ_0, appropriate phase shift is introduced at each sensor input so that the looking direction from the ABF point of view always lie at zero degree. Consequently, the steering vector of the looking direction is given by $\boldsymbol{a}(\theta_0) = [1, 1, ..., 1]^{\mathsf{T}}$ and the fixed BF amounts to average the outputs of all sensors though conventional beampatterns with lower sidelobes can also be used. The orthogonality constraints then translate into the condition that the coefficients of each row of the BM should sum up to zero and the rows should span a subspace of dimension of $L-1$. As a result, the input signal to the adaptive filter $\boldsymbol{w}_a$, $\boldsymbol{u}(n) = \boldsymbol{B}\boldsymbol{x}(n)$, is an $L-1$-th vector. The choice of the blocking matrix in the GSC is not unique and the following GJ BM,

$$\boldsymbol{B} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -1 & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \tag{7}$$

is commonly used as it requires only simple subtractions of consecutive sensor inputs. More sophisticated BM can be derived from the eigendecomposition (EVD) of $\mathbb{E}[\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)]$, which completely whitens $\boldsymbol{u}(n)$ so that the LMS algorithm can be significantly accelerated. An alternative approach is to employ the RLS algorithm with higher complexity. Here, we propose to implement the BM using the MB, where its coefficients are approximated by SOPOT coefficients. Moreover, for the commonly encountered case of ULA, it is advantageous to employ a frequency decomposition-like BFs for the rows of $\boldsymbol{B}$. This is because the corresponding steering vector in (2) represents a complex sinusoid for a given DOA θ. Thus, using such a BM is equivalent to focusing at different angular space using its

$L-1$ rows. Accordingly, interference from each region will mainly appear at the corresponding component of $\boldsymbol{u}(n)$. Such decoupling greatly simplifies the selection of the appropriate weight to annihilate the corresponding interference at the fixed BF output, which leads to significant speed up of the LMS algorithm through individual power normalization of the components in $\boldsymbol{u}(n)$. An attractive choice of $\boldsymbol{B}$ is the basic function of the DCT, except the first component which corresponds to $\frac{1}{L}[1, 1, ..., 1]^{\mathsf{T}} = \frac{1}{L}\boldsymbol{a}(\theta_0) = \boldsymbol{w}_q$, the region around the looking direction. Another interpretation of this choice of $\boldsymbol{B}$ and the subsequent component power normalization is to view it as a decorrelation transformation which approximately pre-whitens the input $\boldsymbol{u}(n)$ to reduce the eigenvalue spread of its covariance matrix. Consequently, the convergence rate can be improved significantly as in the TDNLMS algorithm [10]. More precisely, the (p, q) element of the DCT-based BM reads

$$\boldsymbol{B}_{p,q} = \sqrt{2/L}\cos[(2q-1)p\pi/(2L)], \tag{8}$$

$p = 1, 2, ..., L-1, q = 1, 2, ..., L$. The problem for finding $\boldsymbol{w}_a$ can therefore be reformulated as the following unconstrained problem

$$\begin{aligned}\min_{\boldsymbol{w}_a} &\mathbb{E}[|(\boldsymbol{w}_q - \boldsymbol{B}^{\mathsf{H}}\boldsymbol{w}_a)^{\mathsf{H}}\boldsymbol{x}(n)|^2] \\ &= \min_{\boldsymbol{w}_a}\mathbb{E}[|d(n) - \boldsymbol{w}_a^{\mathsf{H}}\boldsymbol{u}(n)|^2] = \min_{\boldsymbol{w}_a}\mathbb{E}[|e(n)|^2],\end{aligned} \tag{9}$$

where $d(n) = \boldsymbol{w}_q^{\mathsf{H}}\boldsymbol{x}(n)$ and $\boldsymbol{u}(n) = \boldsymbol{B}\boldsymbol{x}(n)$. This least square problem in $\boldsymbol{w}_a$ can be solved recursively using an adaptive filtering with desired signal $d(n) = \boldsymbol{w}_q^{\mathsf{H}}\boldsymbol{x}(n)$ and input $\boldsymbol{u}(n) = \boldsymbol{B}\boldsymbol{x}(n)$ where the sensor input $\boldsymbol{x}(n)$ has been transformed by the BM $\boldsymbol{B}$. Moreover, to improve the robustness to uncertainties, (9) can be modified by adding a regularization term as follows

$$\min_{\boldsymbol{w}_a}\mathbb{E}[|e(n)|^2] + \xi\|\boldsymbol{D}\boldsymbol{w}_a(n)\|_2^2, \tag{10}$$

where the second term represents a weighted $\ell 2$ regularization term on $\boldsymbol{w}_a(n)$, ξ is the regularization parameter and $\boldsymbol{D} > \boldsymbol{0}$ is a positive definite weight matrix. In general, it can be made adaptive so as to approximate different regularization methods as in the conventional iterative reweighted least squares method [34] Specifically, for $\ell 2$ regularization, $\boldsymbol{D}$ is chosen as the identity matrix. For $\ell 1$ regularization, $\boldsymbol{D}$ is chosen as diag$[1/\sqrt{|w_{a,1}(n)| + \epsilon}, 1/\sqrt{|w_{a,2}(n)| + \epsilon}, ..., 1/\sqrt{|w_{a,L-1}(n)| + \epsilon}]$, where ϵ is a small positive number to avoid division by zero. For the smoothly clipped absolute deviation (SCAD) penalty [35], $\boldsymbol{D}$ can be chosen as diag$[\sqrt{p_\epsilon(w_{a,1}(n))}/(w_{a,1}(n)+\epsilon), \sqrt{p_\epsilon(w_{a,2}(n))}/(w_{a,2}(n)+\epsilon), ..., \sqrt{p_\epsilon(w_{a,L-1}(n))}/(w_{a,L-1}(n)+\epsilon)]$, where

$$p_\epsilon(c) = \begin{cases} |c|, & \text{for} \quad |c| \le \xi, \\ -\frac{(|c|-\tilde{a}\xi)^2}{2(\tilde{a}-1)\xi} + \frac{(\tilde{a}+1)\xi}{2}, & \text{for} \quad \mu < |c| \le \tilde{a}\xi, \\ (\tilde{a}+1)\xi^2/2, & \text{for} \quad |c| > \tilde{a}\xi, \end{cases} \tag{11}$$

and the coefficient $\tilde{a} > 2$ is chosen as 3.7 [35]. For this paper, we focus on the $\ell 2$ regularization so as to derive an analytical formula for the regularization parameter and hence $\boldsymbol{D} = \boldsymbol{I}$. The motivation of the regularization term is to avoid significant signal cancellation when the steering vector of the SOI (and hence the blocking matrix $\boldsymbol{B}$) is subject to uncertainties due to imperfections of the antenna array, etc. More precisely, this DOA mismatch can lead to signal leakage to the adaptive part of the ABF, which will be treated as interference leading to signal cancellation. As this leaked source signal is of much smaller magnitude than the fixed BF output, the resultant adaptive weight will assume large magnitude. Hence, by penalizing the $\ell 2$ norm of $\boldsymbol{w}_a(n)$, undesirable signal cancellation can be mitigated. As the VR-SNC-TDNLMS algorithm in [22] offers a low complexity and efficient solution to (10) for real-valued input, we now propose a new ABF based on its complex generalization and present its performance analysis. Its efficient hardware implementation is described in Section III. The blocking matrix and multiplication can be efficiently implemented by SOPOT and MB technique to be presented in Section III-B.

B. COMPLEX VR-SNC-TDNLMS ALGORITHM

The update equations for the complex VR-SNC-TDNLMS algorithm for our ABF problem are given by

$$e(n) = d(n) - \boldsymbol{w}_a^{\mathsf{H}}(n)\boldsymbol{u}(n), \tag{12}$$

$$\boldsymbol{w}_a(n+1) = \boldsymbol{w}_a(n) + \mu(n)\boldsymbol{D}_u(\boldsymbol{u}(n)e^*(n) - \xi\boldsymbol{R}_w\boldsymbol{w}_a(n)), \tag{13}$$

where $\boldsymbol{u}(n)$ and $d(n)$ are respectively the transformed input and desired response of the adaptive filter $\boldsymbol{w}_a(n)$, $\mu(n) > 0$ is a variable step-size to be elaborated further below, $\xi \ge 0$ is the regularization parameter, $\boldsymbol{R}_w = \boldsymbol{D}^{\mathsf{H}}\boldsymbol{D}$, $\boldsymbol{D}_u = \text{diag}[\varepsilon_1^{-1}(n), \varepsilon_2^{-1}(n), ..., \varepsilon_{L-1}^{-1}(n)]$ is an element-wise normalization matrix where $\varepsilon_i(n)$ is the estimated power of the i-th element of $\boldsymbol{u}(n)$, $u_i(n)$, and $*$ denotes complex conjugation. Note that in the context of ABF, the blocking matrix serves the purpose of the transformation. If $\boldsymbol{R}_w$ and $\boldsymbol{D}_u$ are equal to the identity matrix with a constant step-size, the weight update reduces to the conventional leaky LMS algorithm. The normalization step can be efficiently implemented with the division-free approach to be introduced in Section III-A.

The main differences between the complex and real-valued updates are that i) the Hermitian transposition of $\boldsymbol{w}_a(n)$ is used in (12), instead of the matrix transposition and ii) $e^*(n)$ is used in (13), instead of $e(n)$. By using the output of the BF $\boldsymbol{w}_q$, i.e. $\boldsymbol{w}_q^{\mathsf{H}}\boldsymbol{x}(n)$, as the desired signal $d(n)$, the adaptive filter can be used to solve iteratively the least square problem in (9) and the steady state (SS) misadjustment of the weight vector $\boldsymbol{w}_a(n)$ as compared to the optimal least square solution decreases with the step-size $\mu(n)$. On the other hand, the convergence speed increases with $\mu(n)$ while $\mu(n)$ has to be smaller than a maximum value for ensuring stability.

The SNC scheme aims to improve the convergence rate and SS error by using a maximum step-size mode (MSM) during initial convergence and a noise constrained mode (NCM)

when the algorithm is near convergence. The MSM facilitates fast initial convergence with a designated maximum step-size (MSS) where the step-size is gradually reduced in the NCM mode to reduce the excess mean square error (EMSE) and hence misadjustment by exploring prior knowledge of the minimum MSE. As suggested in [22], the step-size of the SNC-NLMS algorithm is updated as follows

$$\mu(n) = \begin{cases} \mu_{max} & \bar{\lambda} \geq T \quad (MSM) \\ \alpha(1+\gamma\lambda(n)) & \bar{\lambda} < T \quad (NCM) \end{cases}, \tag{14a}$$

$$\bar{\lambda}(n+1) = \lfloor (1-\bar{\beta})\bar{\lambda}(n) + \bar{\beta}\hat{J}(n)/2 \rfloor_{\delta/\gamma}, \tag{14b}$$

$$\lambda(n+1) = \lfloor (1-\beta)\lambda(n) + \beta\hat{J}(n)/2 \rfloor_{\delta/\gamma}, \tag{14c}$$

where μ_{max} is the MSS used, α, γ and T are algorithmic parameters, $\hat{J}(n) = |e(n)|^2 - \hat{\sigma}_\eta^2$ is the instantaneous EMSE, $\lfloor x \rfloor_b$ lower bounds the argument x by b, and $\hat{\sigma}_\eta^2$ is a prior estimate of the additive noise power and can be chosen as zero if it is unavailable. $\bar{\lambda}(n)$ and $\lambda(n)$ are respectively short-term and long term estimates of the EMSE with the constant $\bar{\beta} \in (0,1)$ considerably larger than $\beta \in (0,1)$. When $\bar{\lambda}(n)$ is larger than a certain threshold T (its selection will be discussed later in section II-D), the MSM is invoked. On the other hand, when $\bar{\lambda}(n)$ is smaller than T, the adaptive filter is nearly converged and the NCM is invoked. The value of $\lambda(n)$ immediately after mode switching is obtained from $\bar{\lambda}(n)$. To avoid problem when $\hat{J}(n)$ is negative, both $\bar{\lambda}(n)$ and $\lambda(n)$ are lower bounded by δ/γ [22]. In case of noise variance mismatch, i.e. the prior noise variance is different from the true one, the noise variance estimates in (14b) and (14c) can be written as $\hat{\sigma}_\eta^2 = a\sigma_\eta^2$ where a is the mismatch factor. If prior information on σ_η^2 is unavailable, one can set $\hat{\sigma}_\eta^2 = 0$, which gives rise to the zero-noise-constrained (ZNC) case. To facilitate the selection of the algorithmic parameters, we briefly summarize the performance of the complex VR-SNC-TDNLMS algorithm.The constant can be efficiently realized by SOPOT to be introduced in Section III-B. The resultant algorithm is summarized in Table 1.

We now present a statistical analysis on the performance of the proposed ABF. We first show that under reasonable assumptions and Gaussian inputs, the mean weight vector of the proposed algorithm is convergent with a certain step-size bound. Hence, the mean weight vector will converge to the Wiener solution (or its regularized version if regularization is imposed). After that the weight vector will vary around its true value due to the effect of noise. The mean square deviation of the weight vector from the desired solution is then evaluated analytically in the mean square convergence analysis and the stability condition for convergence will be determined. The steady state MSE is then determined so as to determine the algorithmic parameters such as the regularization parameters, switching threshold etc, for operation. The theoretical results will be validated via Monte-Carlo simulation in Section IV to demonstrate the accuracy of the statistical analysis.

TABLE 1. The DCT-based adaptive beamformer algorithm.

Initialization:
Choose $\alpha = 0.0004$, $\beta = 0.01$, $\bar{\beta} = 0.05$ and $\gamma = 20$,
$\mu_{max} = 1/(L-1)$, $\lambda(0) = \bar{\lambda}(0) = 0$ and $\boldsymbol{w}_a(0) = \mathbf{0}_{(L-1)\times 1}$.
Compute the fixed beamformer $\boldsymbol{w}_q = \boldsymbol{a}(\theta_0)(\boldsymbol{a}^{\mathsf{H}}(\theta_0)\boldsymbol{a}(\theta_0))^{-1}$,
the DCT-based blocking matrix $\boldsymbol{B}$ in (8) and the switching
threshold T in (41).
Recursion:
Given $\boldsymbol{x}(n)$, $\boldsymbol{w}_a(n-1)$, $\lambda(n-1)$ and $\bar{\lambda}(n-1)$ compute
at time instant n
(i). Compute the input $\boldsymbol{u}(n)$ and output $d(n)$,
$\boldsymbol{u}(n) = \boldsymbol{B}\boldsymbol{x}(n)$,
$d(n) = \boldsymbol{w}_q^{\mathsf{H}}\boldsymbol{x}(n)$,
(ii). Update $\lambda(n)$ and $\bar{\lambda}(n)$ in (14c) and (14b),
(iii). Compute the variable step size $\mu(n)$ in (14a),
(iv) Compute the element-wise normalization matrix $\boldsymbol{D}_u$ in (13),
(v) Compute the regularization parameter $\xi(n)$ in (40),
(vi) Update the adaptive weight $\boldsymbol{w}_a(n)$,
$e(n) = d(n) - \boldsymbol{w}_a^{\mathsf{H}}(n-1)\boldsymbol{u}(n)$,
$\boldsymbol{w}_a(n) = \boldsymbol{w}_a(n-1) + \mu(n)\boldsymbol{D}_u(\boldsymbol{u}(n)e^*(n) - \xi(n)\boldsymbol{R}_w\boldsymbol{w}_a(n))$,
(vii) Compute the adaptive beamfomer $\boldsymbol{w}(n)$,
$\boldsymbol{w}(n) = \boldsymbol{w}_q(n) - \boldsymbol{B}\boldsymbol{w}_a(n)$.

C. STATISTICAL ANALYSIS OF PERFORMANCE

We first examine the performance of the optimal ABF in (5) without regularization. The minimum output power of the ABF is given by $\sigma_\eta^2 = \mathbb{E}[|\boldsymbol{w}_{opt}^{\mathsf{H}}\boldsymbol{x}(n)|^2] = \boldsymbol{w}_{opt}^{\mathsf{H}}\boldsymbol{R}_{xx}\boldsymbol{w}_{opt}$, where $\boldsymbol{R}_{xx} = \mathbb{E}[\boldsymbol{x}(n)\boldsymbol{x}^{\mathsf{H}}(n)]$ is the covariance matrix of the sensor signal vector, which is assumed to be independent and identical distributed over time. Using the expression of $\boldsymbol{w}_{opt}$ in (5), one gets [36]

$$\sigma_\eta^2 = (\boldsymbol{a}^{\mathsf{H}}(\theta_0)\boldsymbol{R}_{xx}\boldsymbol{a}(\theta_0))^{-1}. \tag{15}$$

Since the GSC is an equivalent implementation of the MVDR beamformer, the minimum output power is also equal to σ_η^2, which is achieved when $\boldsymbol{w}_a$ is equal to the Wiener solution $\boldsymbol{w}_0 = \boldsymbol{R}_{uu}^{-1}\boldsymbol{P}_{ud}$, where $\boldsymbol{R}_{uu} = \mathbb{E}[\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)]$ is the input covariance matrix and $\boldsymbol{P}_{ud} = \mathbb{E}[\boldsymbol{u}(n)\boldsymbol{d}^*(n)]$ is the cross-correlation vector between $\boldsymbol{u}(n)$ and $d(n)$. The output error at time n is thus given by

$$e(n) = [\boldsymbol{w}_0 - \boldsymbol{w}_a(n)]^{\mathsf{H}}\boldsymbol{u}(n) + \eta(n), \tag{16}$$

where $\eta(n)$ contains the target signal corrupted by the residual interference plus noise after adaptive beamforming. If the target signal, interference and sensor noise are independent and identically distributed circular symmetric complex Gaussian signals with zero mean, then $\eta(n)$ will be a white circular symmetric complex Gaussian process with zero mean and its variance will be equal to σ_η^2. Before proceeding to the analysis of the proposed complex VR-SNC-TDNLMS, the assumptions used to facilitate the analysis are summarized below:

(**A1**) the step-size $\mu(n)$ is independent of the input and error sequence;

(**A2**) $\boldsymbol{u}(n)$ are a sequence of independent identically distributed circular symmetric complex Gaussian random vectors with zero-mean and covariance matrix $\boldsymbol{R}_{uu}$;

$\eta(n)$ is a circular symmetric complex Gaussian random variable with zero-mean and variance σ_η^2;

(**A**3) $\boldsymbol{w}(n)$, $\boldsymbol{u}(n)$ and $\eta(n)$ are statistically independent;

(**A**4) the elements in $\boldsymbol{D}_u$, $\varepsilon_i(n)$, are uncorrelated with $\boldsymbol{w}(n)$ and $\boldsymbol{u}(n)$ due to the recursive averaging effect of $\sigma_{u\_i}^2$.

(**A**1) is commonly used in most analysis of VSS LMS algorithms for mathematically tractability, which is a good approximation if the step-size is obtained from sufficiently smoothed measurements of the convergence status such as instantaneous EMSE [12], [37]. (**A**2) is a commonly used assumption for simplifying the evaluation of the expectations arising from analysis [38]. (**A**3) is the independence assumption, which is quite accurate for large filter length and small to medium step-sizes [38] and (**A**4) is generally valid after the individual power estimates of $u_i(n)$ has been stabilized, which can be achieved with reasonable number of samples, say 20 [22], [39].

Next, we shall first study the mean convergence of the algorithm in subsection II-C.1 below where we shall show that the algorithm is convergent if the step size is bound by (21). In subsection II-C.2, the mean square convergence behavior of the proposed algorithm is studied where a conservative upper bound for the step-size to ensure convergence is given. The excess mean square error (EMSE) at convergence is also derived. The regularization parameter for minimizing the overall MSE for white input is then derived. The switching threshold is determined in subsection II-D below. The analytical statistical analysis will be further validated by Monte-Carlo Simulation in Section IV.

1) MEAN CONVERGENCE

We first assume that the algorithm is convergent and show that the mean weight vector is equal to the regularized solution of (10) at the steady-state. Then, we establish condition for mean convergence. Taking expectation on both sides of (13), one gets

$$\mathbb{E}[\boldsymbol{w}_a(n+1)]=\mathbb{E}[\mu(n)]\boldsymbol{D}_u\{\boldsymbol{P}_{ud}-(\boldsymbol{R}_{uu}+\xi\boldsymbol{R}_w)\mathbb{E}[\boldsymbol{w}_a(n)]\}, \tag{17}$$

which gives at the steady state

$$\boldsymbol{w}_R=(\boldsymbol{R}_{uu}+\xi\boldsymbol{R}_w)^{-1}\boldsymbol{P}_{ud}, \tag{18}$$

where $\boldsymbol{w}_R = \mathbb{E}[\boldsymbol{w}(\infty)]$ is the desired regularized Wiener solution of (10). To examine the convergence rate, let's consider the weight error vector $\boldsymbol{v}(n) = \boldsymbol{w}_a(n) - \boldsymbol{w}_R$ in (17) at time n. We only focus on the NC adaptation mode, as it reduces to the MSM when a fixed maximum designated step-size is used. Using (13), (17) and the assumptions above, the following difference equation in the mean weight error vector is obtained

$$\mathbb{E}[\boldsymbol{v}(n+1)]=[\boldsymbol{I}-\mathbb{E}[\mu(n)]\boldsymbol{D}_u(\boldsymbol{R}_{uu}+\xi\boldsymbol{R}_w)]\mathbb{E}[\boldsymbol{v}(n)]. \tag{19}$$

Let $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{U}}^{\mathsf{H}}$ be the eigendecomposition of $\bar{\boldsymbol{R}}_{uu} = \boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{R}}_{uu}\boldsymbol{D}_u^{1/2}$ with $\tilde{\boldsymbol{R}}_{uu} = \boldsymbol{R}_{uu} + \xi\boldsymbol{R}_w$. Using the change of variable $\boldsymbol{V}(n) = \tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{-1/2}\boldsymbol{v}(n)$, (19) can be decoupled to obtain the following equation in the i-th element of $\mathbb{E}[\boldsymbol{V}(n)]$:

$$\mathbb{E}[\boldsymbol{V}(n+1)]_i=(1-\mathbb{E}[\mu(n)]\tilde{\lambda_i})\mathbb{E}[\boldsymbol{V}(n)]_i, \tag{20}$$

where $\tilde{\lambda_i}$ is the i-th eigenvalue of $\bar{\boldsymbol{R}}_{uu}$. Thus, the mean weight vector of the adaptive filter converges if

$$0<\mathbb{E}[\mu(n)]<2/\tilde{\lambda_i},\ i=1,...,L, \tag{21}$$

and the maximum possible step-size is $\mu_{max} = 2/\tilde{\lambda}_{max}$, where $\tilde{\lambda}_{max}$ is the maximum eigenvalue of $\bar{\boldsymbol{R}}_{uu}$. In the MSM, μ is chosen as a value which is close to but smaller than this maximum value. Since the convergence rate increases with the step-size, the MSM is always faster than the NC-mode initially. However, at the steady state, the gradient noise dominates and the step-size has to be reduced gradually. When the mean weight vector converges, the accuracy of the estimation is determined by the variation of the weight vector around its true value, which we shall study below.

2) MEAN SQUARE CONVERGENCE ANALYSIS

a: DIFFERENCE EQUATION

We first multiply $\boldsymbol{v}(n)$ by its Hermitian transpose and then evaluate the expectation of both sides. This yields the following difference equation in the weight error covariance matrix $\boldsymbol{\Xi}_{vv}(n) = \mathbb{E}[\boldsymbol{v}(n)\boldsymbol{v}^{\mathsf{H}}(n)]$,

$$\begin{aligned}\boldsymbol{\Xi}_{vv}(n+1) &= \boldsymbol{\Xi}_{vv}(n)-\mathbb{E}[\mu(n)](\boldsymbol{D}_u\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(n))\\ &+\boldsymbol{\Xi}_{vv}(n)\tilde{\boldsymbol{R}}_{uu}\boldsymbol{D}_u^{\mathsf{T}})+\mathbb{E}[\mu^2(n)]\boldsymbol{D}_u\mathbb{E}[\boldsymbol{u}(n)e^*(n)\\ &-\xi\boldsymbol{R}_w\boldsymbol{w}_a(n))(\boldsymbol{u}^{\mathsf{H}}(n)e(n)-\xi\boldsymbol{w}_a^{\mathsf{H}}(n)\boldsymbol{R}_w^{\mathsf{H}})]\boldsymbol{D}_u^{\mathsf{T}}.\end{aligned} \tag{22}$$

By noting that $e(n) = [\boldsymbol{w}_0 - \boldsymbol{w}_a(n)]^{\mathsf{H}}\boldsymbol{u}(n) + \eta(n) = \hat{\boldsymbol{v}}^{\mathsf{H}}\boldsymbol{u}(n) + \eta(n)$, where $\hat{\boldsymbol{v}} = \boldsymbol{w}_0 - \boldsymbol{w}_R - \boldsymbol{v}(n) = \Delta\boldsymbol{w} - \boldsymbol{v}(n)$, the last term can be rewritten as $\boldsymbol{S}(n) = \mathsf{E}[\mu^2(n)]\boldsymbol{D}_u[\boldsymbol{A}(n) + \boldsymbol{R}_{uu}\sigma_\eta^2]\boldsymbol{D}_u^{\mathsf{T}}$ where $\boldsymbol{A}(n) = \boldsymbol{A}_0(n) - \boldsymbol{A}_1(n) - \boldsymbol{A}_2(n) + \boldsymbol{A}_3(n)$, $\boldsymbol{A}_0(n) = \mathbb{E}[\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)\boldsymbol{\Xi}_{\tilde{v}\tilde{v}}\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)]$, $\boldsymbol{A}_1(n) = \boldsymbol{A}_2^{\mathsf{H}}(n) = \xi\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{\tilde{v}w}\boldsymbol{R}_w^{\mathsf{H}}$, $\boldsymbol{A}_3(n) = \xi^2\boldsymbol{R}_w\boldsymbol{\Xi}_{ww}(n)\boldsymbol{R}_w^{\mathsf{H}}$ with $\boldsymbol{\Xi}_{\tilde{v}\tilde{v}}(n) = \Delta\boldsymbol{w}\Delta\boldsymbol{w}^{\mathsf{H}} + \boldsymbol{\Xi}_{vv}(n) - \bar{\boldsymbol{v}}(n)\Delta\boldsymbol{w}^{\mathsf{H}} - \Delta\boldsymbol{w}\bar{\boldsymbol{v}}^{\mathsf{H}}(n)$, $\boldsymbol{\Xi}_{ww}(n) = \boldsymbol{w}_R\boldsymbol{w}_R^{\mathsf{H}} + \boldsymbol{\Xi}_{vv}(n) + \bar{\boldsymbol{v}}(n)\boldsymbol{w}_R^{\mathsf{H}} + \boldsymbol{w}_R^{\mathsf{H}}\bar{\boldsymbol{v}}^{\mathsf{H}}(n)$, $\boldsymbol{\Xi}_{\tilde{v}w}(n) = \Delta\boldsymbol{w}\boldsymbol{w}_R^{\mathsf{H}} - \boldsymbol{\Xi}_{vv}(n) - \bar{\boldsymbol{v}}(n)\boldsymbol{w}_R^{\mathsf{H}} + \Delta\boldsymbol{w}\bar{\boldsymbol{v}}^{\mathsf{H}}(n)$ and $\bar{\boldsymbol{v}}(n) = \mathbb{E}[\boldsymbol{v}(n)]$.

b: STABILITY CONDITION

Similar to [22], we multiply both sides of (22) by $\boldsymbol{D}_u^{-1}$ and take their trace to obtain the following difference equation in the measure $\varphi(n) = \mathrm{Tr}(\boldsymbol{D}_u^{-1}\boldsymbol{\Xi}_{vv}(n))$:

$$\begin{aligned}\varphi(n+1) = \varphi(n) &- \{2\mathsf{E}[\mu(n)]\mathrm{Tr}(\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(n))\\ &-\mathrm{Tr}(\boldsymbol{D}_u^{-1}\boldsymbol{S}(n))\},\end{aligned} \tag{23}$$

where $\varphi(n)$ serves the role of the Lyapunov function, and is always positive. We only need to consider those terms in $\boldsymbol{\Xi}_{vv}(n)$ as the driving terms in (23) with respect to σ_η^2, $\boldsymbol{w}_R$ and $\Delta\boldsymbol{w}$ are finite, and $\mathbb{E}[\boldsymbol{v}(n)]$ will converge to zero if (21) is satisfied. Furthermore, as $\varphi(n)$ is positive, if the term inside the brackets in (23) is positive, the system is guaranteed to be stable. Inserting $\boldsymbol{A}_0(n)$, $\boldsymbol{A}_1(n)$, $\boldsymbol{A}_2(n)$ and $\boldsymbol{A}_3(n)$ into (22),

one gets after some simplification the following sufficient condition for stability

$$\mu_{R-TD}(n) < 2\mathrm{Tr}(\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(n))/\tau(n), \tag{24}$$

where $\mu_{R-TD}(n) = \mathbb{E}[\mu^2(n)]/\mathbb{E}[\mu(n)]$ and $\tau(n) = \mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}\boldsymbol{D}_u) + \mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{vv}(n))\mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{D}_u) + \xi\mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_w\boldsymbol{D}_u) + \xi\mathrm{Tr}(\boldsymbol{R}_w\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}\boldsymbol{D}_u) + \xi^2\mathrm{Tr}(\boldsymbol{R}_w\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_w\boldsymbol{D}_u)$. Since $\tau(n)$ depends on $\boldsymbol{\Xi}_{vv}(n)$, (24) cannot be used directly to determine a bound on $\mu(n)$ for practical applications. Using the facts that $\mathrm{Tr}(\boldsymbol{AB}) \le \mathrm{Tr}(\boldsymbol{A})\mathrm{Tr}(\boldsymbol{B})$ for $\boldsymbol{A}, \boldsymbol{B} > 0$ and $\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA})$, the following upper bound for $\tau(n)$ can be obtained

$$\tau(n) \le \mathrm{Tr}(\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(n))\tilde{\tau}, \tag{25}$$

with $\tilde{\tau} = \mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{D}_u\boldsymbol{\Gamma}_u)+\mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{D}_u)\mathrm{Tr}(\boldsymbol{\Gamma}_u)+2\xi\mathrm{Tr}(\boldsymbol{R}_w\boldsymbol{D}_u\boldsymbol{\Gamma}_u)+\xi^2\mathrm{Tr}(\boldsymbol{R}_w\boldsymbol{D}_u\boldsymbol{R}_w\tilde{\boldsymbol{R}}_{uu}^{-1})$ and $\boldsymbol{\Gamma}_u = \boldsymbol{R}_{uu}\tilde{\boldsymbol{R}}_{uu}^{-1}$. Consequently, a conservative upper bound for mean square convergence is

$$\mu_{R-TD}(n) < 2/\tilde{\tau}(n). \tag{26}$$

Compared to the real-valued case, it is found that $\mathbb{E}[\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)\boldsymbol{\Xi}_{vv}(n)\boldsymbol{u}(n)\boldsymbol{u}^{\mathsf{H}}(n)]$ is equal to $\boldsymbol{R}_{uu}(n)\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}(n) + \mathrm{Tr}(\boldsymbol{R}_{uu}(n)\boldsymbol{\Xi}_{vv}(n))\boldsymbol{R}_{uu}(n)$ instead of $2\boldsymbol{R}_{uu}(n)\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}(n) + \mathrm{Tr}(\boldsymbol{R}_{uu}(n)\boldsymbol{\Xi}_{vv}(n))\boldsymbol{R}_{uu}(n)$ in the real-valued case and hence the first term of $\tau(n)$ is $\mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}\boldsymbol{D}_u)$, rather than $2\mathrm{Tr}(\boldsymbol{R}_{uu}\boldsymbol{\Xi}_{vv}(n)\boldsymbol{R}_{uu}\boldsymbol{D}_u)$ in the real-valued case.

c: STEADY-STATE EMSE

If the algorithm converges, $\mathbb{E}[\boldsymbol{v}(\infty)] = \boldsymbol{0}$, and the last term in (22) reduces to $\boldsymbol{S}(\infty) = \mathbb{E}[\mu^2(\infty)]\boldsymbol{D}_u[\boldsymbol{A}(\infty)+\boldsymbol{R}_{uu}\sigma_\eta^2]\boldsymbol{D}_u^{\mathsf{T}}$, which is further evaluated to be[1]

$$\begin{aligned}\boldsymbol{S}(\infty) \approx\ & \mathbb{E}[\mu^2(\infty)]\boldsymbol{D}_u[\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(\infty)\tilde{\boldsymbol{R}}_{uu} \\ & +(J_* + \sigma_{min}^2)\boldsymbol{R}_{uu}]\boldsymbol{D}_u^{\mathsf{T}},\end{aligned} \tag{27}$$

where $\sigma_{min}^2 = \mathrm{Tr}(\Delta\boldsymbol{w}\Delta\boldsymbol{w}^{\mathsf{H}}\boldsymbol{R}_{uu}) + \sigma_\eta^2$ is the minimum MSE. Substituting (27) into (22) gives the following equation for weight error covariance at steady state

$$\begin{aligned}\boldsymbol{\Xi}_{vv}(\infty) \approx\ & \boldsymbol{\Xi}_{vv}(\infty) - \mathbb{E}[\mu(\infty)](\boldsymbol{D}_u[\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(\infty) \\ & +\boldsymbol{\Xi}_{vv}(\infty)\tilde{\boldsymbol{R}}_{uu}\boldsymbol{D}_u^{\mathsf{T}} + \mathbb{E}[\mu^2(\infty)]\boldsymbol{D}_u\{\tilde{\boldsymbol{R}}_{uu}\boldsymbol{\Xi}_{vv}(\infty)\tilde{\boldsymbol{R}}_{uu} \\ & +(J_* + \sigma_{min}^2)\boldsymbol{R}_{uu}\}\boldsymbol{D}_u^{\mathsf{T}},\end{aligned} \tag{28}$$

where we have dropped the negative term to obtain a simplified upper bound of the covariance. By expressing $\boldsymbol{v}(n)$ in the transformed coordinate: $\boldsymbol{V}(n) = \tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{-1/2}\boldsymbol{v}(n)$ and using the fact that $\tilde{\boldsymbol{\Lambda}} = \tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{R}}_{uu}\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{U}}$, the diagonal values of $\boldsymbol{\Xi}(\infty)$ are found to be

$$\begin{aligned}[\boldsymbol{\Xi}_{VV}(\infty)]_{i,i} &= \frac{\mathbb{E}[\mu^2(\infty)](J_* + \sigma_{min}^2)\boldsymbol{\Gamma}_{1,i}}{2\mathbb{E}[\mu(\infty)]\tilde{\lambda}_i - \mathbb{E}[\mu^2(\infty)]} \\ &\approx \frac{\mathbb{E}[\mu^2(\infty)]}{2\mathbb{E}[\mu(\infty)]\tilde{\lambda}_i}(J_* + \sigma_{min}^2)\boldsymbol{\Gamma}_{1,i},\end{aligned} \tag{29}$$

[1]Due to the difference in the four order expectation of the real- and complex valued Gaussian random vectors, the expression for the complex case is somewhat different from their real counterparts.

where $\boldsymbol{\Gamma}_{1,i} = [\boldsymbol{\Gamma}_1]_{i,i}$ is the i-th diagonal value of $\boldsymbol{\Gamma}_1 = \tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{1/2}\boldsymbol{R}_{uu}\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{U}}$ and the last equation is obtained as $\mathbb{E}[\mu^2(\infty)]$ is usually small as compared to $\mathbb{E}[\mu(\infty)]$ at the steady state. Next, we note that the SS-EMSE around the converged solution is

$$J_* = \mathrm{Tr}(\boldsymbol{\Xi}_{VV}(\infty)\tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{1/2}\boldsymbol{R}_{uu}\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{U}}). \tag{30}$$

It is the excess noise power at the ABF output due to the adaptation process. Its value is related to the degradation due to the adaptive filter over ideal Wiener filter. If the transform/blocking matrix can approximately diagonalize $\boldsymbol{R}_{uu}$, then $\tilde{\boldsymbol{R}}_{uu} = \boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{R}}_{uu}\boldsymbol{D}_u^{1/2} \approx \boldsymbol{I}$, and hence $\tilde{\boldsymbol{U}}$ is approximately equal to the identity matrix. Thus, $\tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{1/2}\boldsymbol{R}_{uu}\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{U}}$ in (30) is approximately equal to $\tilde{\boldsymbol{\Lambda}} - \xi\tilde{\boldsymbol{U}}^{\mathsf{H}}\boldsymbol{D}_u^{1/2}\boldsymbol{R}_w\boldsymbol{D}_u^{1/2}\tilde{\boldsymbol{U}} \approx \tilde{\boldsymbol{\Lambda}} - \xi\boldsymbol{D}_u^{1/2}\boldsymbol{R}_w\boldsymbol{D}_u^{1/2}$. Moreover, for diagonal dominance $\boldsymbol{R}_w$, (30) can be simplified to

$$\begin{aligned}J_* &\approx \mathrm{Tr}(\boldsymbol{\Xi}_{VV}(\infty)(\tilde{\boldsymbol{\Lambda}} - \xi\boldsymbol{D}_u^{1/2}\boldsymbol{R}_w\boldsymbol{D}_u^{1/2})) \\ &\approx \sum_{i=1}^{L}[\boldsymbol{\Xi}_{VV}(\infty)]_{i,i}(\tilde{\lambda}_i - \xi\varepsilon_i^{-1}R_{w\_i,i}),\end{aligned} \tag{31}$$

where $R_{w\_i,i}$ is the i-th diagonal element of $\boldsymbol{R}_w$. Combining (29) and (31), the steady state EMSE J_* is found to be

$$J_* \approx (\frac{1}{2}\mathbb{E}[\mu^2(\infty)]/\mathbb{E}[\mu(\infty)])(J_* + \sigma_{min}^2)\phi_{TD}, \tag{32}$$

where $\phi_{TD} = \sum_{i=1}^{L}\boldsymbol{\Gamma}_{1,i}(\tilde{\lambda}_i - \xi\varepsilon_i^{-1}R_{w\_i,i})\tilde{\lambda}_i^{-1}$. For constant step-size, the term $\mathbb{E}[\mu^2(\infty)]/\mathbb{E}[\mu(\infty)]$ is simplified equal to μ, and the SS-EMSE of the corresponding fixed step-size NLMS algorithm is then given by $J_* \approx \frac{1}{2}\mu\sigma_{min}^2\phi_{TD}(1 - \frac{1}{2}\mu\phi_{TD})^{-1}$. For the SNC algorithm, one needs to evaluate the term $\mathbb{E}[\mu^2(\infty)]/\mathbb{E}[\mu(\infty)]$ from the steady-state expected values of $\mu(n)$, $\mu^2(n)$, $\lambda(n)$ and $\lambda^2(n)$. It is shown in the Appendix that

$$\mathbb{E}[\lambda(\infty)] = J_{*b}/2, \tag{33}$$

$$\begin{aligned}\mathbb{E}[\lambda^2(\infty)] \approx\ & \frac{(1-\beta)}{2(2-\beta)}(J_* - b\sigma_\eta^2)^2 \\ & +\frac{\beta}{4(2-\beta)}(b_1\sigma_\eta^2 J_* + b_2\sigma_\eta^4),\end{aligned} \tag{34}$$

$$\mathbb{E}[\mu(\infty)] = \alpha(1 + \gamma J_{*b}/2), \tag{35}$$

$$\mathbb{E}[\mu^2(\infty)] = \alpha^2(1 + 2\gamma\mathbb{E}[\lambda(\infty)] + \gamma^2\mathbb{E}[\lambda^2(\infty)]), \tag{36}$$

where $b_1 = 2(1-b)$, $b_2 = 2+(1+b)^2$, J_{*b} is the steady-state value of $J_b(n) = \mathbb{E}[|e(n)|^2] - a\sigma_\eta^2 = J(n) - b\sigma_\eta^2$, and $J(n) = \mathbb{E}[|e(n)|^2] - \sigma_\eta^2$ is the EMSE. Here, $b = a - 1$ is the excess noise mismatch factor and when $b = 0$, there is no noise mismatch and $J_b(n) = J(n) = \mathbb{E}[\hat{J}(n)]$.

Again due to the difference in the four-order expectation of the real- and complex-valued Gaussian random variables, the values of b_1 and b_2 are now $b_1 = 2(1-b)$ and $b_2 = 2+(1+b)^2$, instead of $b_1 = 2(2-b)$ and $b_2 = 3+(1+b)^2$ in the real-valued case. Consequently, the SS-EMSE can be

approximated as

$$J_* \approx \frac{\frac{1}{2}\alpha A_0 \sigma_{min}^2 \phi_{TD}}{1 - \frac{\gamma b}{2}\sigma_\eta^2 - \frac{1}{2}\alpha[A_0\phi_{TD} + A_1\sigma_{min}^2\phi_{TD}]} \approx \frac{\alpha A_0 \sigma_{min}^2 \phi_{TD}}{2(1 - \frac{\gamma b}{2}\sigma_e^2 ta)} = \frac{1}{2}\alpha(1+\delta)\sigma_{min}^2\phi_{TD}, \quad (37)$$

where $\delta = (\frac{\beta b_2 + 2(1-\beta)b^2}{4(2-\beta)}\gamma^2\sigma_\eta^4 - \frac{1}{2}\gamma b\sigma_\eta^2)$ and $A_0 = 1 - \gamma b\sigma_\eta^2 + \frac{(\beta b_2 + 2(1-\beta)b^2)\gamma^2}{4(2-\beta)}\sigma_\eta^4$. It can be seen that the nominal step-size α plays a similar role as the fixed step-size LMS-based algorithm in controlling the SS error. For stationary scenarios, the smaller the step-size, the smaller the SS-MSE will be. However, for tracking applications, the lag error will increase and an optimal SS can be derived according to the classical analysis in [38]. However, it is somewhat difficult to apply as it requires knowledge about the weight variations. Due to the page limitation, we would not consider the tracking analysis here.

Since the MSE is given by $J_R = \mathbb{E}[|\Delta \boldsymbol{w}^{\mathsf{H}}\boldsymbol{u}(n)|^2] + J_*$ and the bias $\Delta\boldsymbol{w}$ is approximately given by $\xi \boldsymbol{R}_{uu}^{-1}\boldsymbol{R}_w\boldsymbol{w}_0$ [39], we have the desired SS EMSE from (37) the desired SS EMSE as

$$J_* \cong \xi^2 \boldsymbol{w}_0^{\mathsf{H}}\boldsymbol{R}_w^{\mathsf{H}}\boldsymbol{R}_{uu}^{-1}\boldsymbol{R}_w\boldsymbol{w}_0 + \frac{1}{2}\alpha(1+\delta)\sigma_{min}^2\phi_{TD}. \quad (38)$$

d: SELECTION OF REGULARIZATION PARAMETER

Using the statistical analysis above, we now derive an expression for the regularization parameter ξ to balance the bias and variance components in the MSE deviation from the Wiener solution. For mathematical tractability, we assume that the input is white with a variance σ_u^2 and $\boldsymbol{R}_w = \boldsymbol{I}$. Hence $\boldsymbol{D}_u = \varepsilon^{-1}\boldsymbol{I}$ is a diagonal matrix, where ε is the estimate of input signal power σ_u^2. Thus, ϕ_{TD} in (32) can be simplified to $\phi_{TD} = (\sum_{i=1}^{L}\boldsymbol{\Gamma}_{1,i})/\tilde{\lambda}$, where $\tilde{\lambda} = \tilde{\lambda}_i = 1 + \xi\sigma_u^{-2}R_{w\_i,i} \approx 1 + \xi\varepsilon^{-1}R_{w\_i,i}$ and the second identity follows from the fact that the input power is assumed to be well estimated, i.e. $\sigma_u^2 \approx \varepsilon$. Combining these results, and using $\tilde{\boldsymbol{U}} \approx \boldsymbol{I}$, (38) becomes $J_R \approx \xi^2\sigma_u^{-2}\|\boldsymbol{w}_0\|_2^2 + \frac{1}{2}\alpha(1+\delta)\sigma_{min}^2(L/\tilde{\lambda})$ where $\tilde{\lambda} = 1 + \xi\varepsilon^{-1}$. To balance the two components, we require them to be equal, which yields

$$\xi_{opt}(\infty) \approx \sqrt{\frac{1}{2}\alpha(1+\delta)L\sigma_{min}^2\sigma_u^2\|\boldsymbol{w}_0\|_2^{-2}}. \quad (39)$$

By noting that $\alpha(1+\delta)$ is the equivalent step-size at the SS, we can use the following formula as the regularization parameter at time n to give the same SS error

$$\xi(n) \approx \sqrt{\frac{1}{2}\mu(n)L\sigma_\eta^2\sigma_u^2\|\boldsymbol{w}_0\|_2^{-2}}, \quad (40)$$

where for simplicity we have assumed that the step-size and regularization parameter are small so that bias components in σ_{min}^2 are small compared to the additive noise σ_η^2. For other values of n, $\xi(n)$ can be viewed as a quasi-stationary approximation of the optimal regularization, if the step-size changes slowly. This can better reflect the effect of the variable step-sizes as compared to the one using the SS value $\alpha(1+\delta)$. $\|\boldsymbol{w}_0\|_2^2$ can be estimated from prior knowledge such as simulations without steer vector mismatch. σ_u^2 denotes the input power which can be estimated as $\sigma_u^2(n) = \lambda_u\sigma_u^2(n-1) + (1-\lambda_u)\boldsymbol{u}^{\mathsf{H}}(n)\boldsymbol{u}(n)$ where λ_u is a positive forgetting factor close to but smaller than one.

D. SWITCHING THRESHOLD AND VSS PARAMETER SELECTION

We now summarize the selection of the switching threshold T and other VSS parameters such as $\beta, \bar{\beta}, \alpha, \delta$ and γ. Using the approach in [39], the threshold T can be chosen to ensure that the MSM has converged for mode switching with a high probability:

$$T = \frac{1}{4}\mu_{max}c_f + \kappa\Delta, \quad (41)$$

where $\Delta = \sqrt{\frac{\beta}{2(2-\beta)}((\sigma_{\eta\_max}^2 - \frac{1}{8}\mu_{max}c_f) + \sigma_{\eta\_max}^4)}$, $c_f = \sigma_{min}^2\phi_{TD}$ and κ is a constant and appropriate values are around 4 to 5, which correspond to around 99.99% and 99.9999% confidence that the MSM has converged. If the input statistic is unknown, the upper bound of c_f with σ_η^2 replaced by its maximum value $\sigma_{\eta\_max}^2$ can be used. Simulation results to be presented in Section IV show that it works well in practice. Typical values for selecting the other algorithmic parameters are summarized in Table 2, which will be presented in Section III-B.

TABLE 2. SOPOT coefficients of the parameters in D-VR-SNC-NLMS GSC adaptive beamformer.

Parameter	Value	SOPOT Representation
β	0.01	$2^{-7} + 2^{-9} + 2^{-10}$
$\bar{\beta}$	0.05	$2^{-5} + 2^{-6} + 2^{-9} + 2^{-10} + 2^{-13} + 2^{-14}$
α	0.0001	$2^{-14} + 2^{-15} + 2^{-17}$
λ_μ	0.01	$2^{-7} + 2^{-9} + 2^{-10}$
$\gamma\alpha$	0.03	$2^{-5} + 2^{-10} + 2^{-12} + 2^{-15}$

III. DIVISION-FREE VR-SNC-TDNLMS ABF AND ITS HARDWARE REALIZATION

A. DIVISION-FREE VR-SNC-TDNLMS ABF

Due to the high-speed requirement of ABF, efficient hardware realization is highly desirable. In particular, pipelined architecture is very attractive due to its high data throughput. To overcome the long delay time in computing the error term $e(n) = d(n) - \boldsymbol{w}_a^{\mathsf{H}}(n)\boldsymbol{u}(n)$ in updating $\boldsymbol{w}_a(n+1)$ of the conventional LMS algorithm, the delayed updating technique [40] is frequently used where the delayed input $\boldsymbol{u}(n-D)$ and error $e(n-D)$, with an appropriate positive integer D, are used instead of their values at the $(n+1)$-th time instant. This technique is also adopted in our pipelined implementation, which leads to the following delayed VR-SNC-TDNLMS

(D-VR-SNR-TDNLMS) update

$$\boldsymbol{w}_a(n+1) = \boldsymbol{w}_a(n) + \mu(n)\boldsymbol{D}_u(\boldsymbol{u}(n-D)e^*(n-D) - \xi \boldsymbol{R}_w \boldsymbol{w}_a(n)). \quad (42)$$

One of the key obstacles in implementing this and other algorithms involving normalization is the need for a fast divider to perform the normalization required in the NLMS or TDNLMS algorithm. The reason is that the throughput of a pipeline is limited by the component with the largest delay. Consequently, most previous works usually focused on the basic LMS algorithm and its simple variant [41]–[43] for high-speed operation.

To address the normalization problem in NLMS-based and related algorithms, we propose a new division-free approach where the normalization quantity is quantized to a limited number of levels so that the division of these fixed set of numbers can be implemented as constant multiplications. Moreover, these constant multiplications can be implemented using canonical signed digits (CSD) or sum-of-power-of-two numbers [28] resulting in multiplier-less realization. We found that by properly selecting these levels, the performance of the resultant division-free VR-SNC-TDNLMS can achieve similar convergence speed as the original algorithm. Moreover, a smaller weight variation is observed as the step-size remains constant around each level, which helps to reduce adaptation noise arising from continuous step-size adaptation.

More precisely, suppose that the sensor signals are in normalized fixed-point format with a magnitude less than one. Then

$$|u_i(n)|^2 = \|\boldsymbol{b}_i^{\mathsf{T}}\boldsymbol{x}(n)\|_2^2 \le \|\boldsymbol{b}_i\|_2^2 \|\boldsymbol{x}(n)\|_2^2 = L, \quad (43)$$

since $\boldsymbol{b}_i$, the i-th row of the DCT matrix, has unit norm. As its power is limited to L, it makes sense to quantize its value to a set of reconstruction levels $\hat{s}_j$, $j \in \{1, ..., q\}$, where q is the total number of discrete levels in approximating the normalization constant. The quantization process can be written as

$$\hat{s}_j = Q[\varepsilon_i], j \in \{1, ..., q\}, \quad (44)$$

where ε_i is the estimated power of the i-th element for performing the normalization. From simulation, we found that $4L$ levels give a good tradeoff between hardware resource required and performance in terms of average deviation of the discretized value to its true value. Since $\hat{s}_j, j \in \{1, ..., q\}$, is a set of constants, their reciprocals can be computed offline as fixed point number $\hat{s}_j^{-1}$, $j \in \{1, ..., q\}$. Consequently, the division is now replaced by a constant multiplication from one of these values. The quantization operation in (44) can be conveniently 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 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 and one multiplication operation. With increasing number of discrete levels, more comparators and hardware are used while the average deviation decreases accordingly. From our experimental results, a uniform quantizer with q chosen as $4L$ is a good tradeoff as there is no significant improvement beyond this value in reducing the approximation error by further increasing the number of discrete levels.

Next, we examine the updates for the step-size and regularization parameter to facilitate efficient hardware implementation. From (14)-(16), one can see that the step-size update requires multiplications with constants such as μ_{max}, α, γ, $\bar{\beta}/2$, $(1-\bar{\beta})$, $\beta/2$ and $(1-\beta)$. Since they are constant multiplications, they can be efficiently implemented using hardware shifts and additions by representing the constants as SOPOT or CSD to be detailed below. On the other hand, we can see from (40) that updating $\xi(n)$ requires the division by $\|\boldsymbol{w}_0\|_2^2$ which is assumed to be a constant prior value and a square root operation. Since $\|\boldsymbol{w}_0\|_2^2$ is a constant, its reciprocal can be precomputed and implemented as a multiplication. Moreover, it can be lumped with other constants in the expression and implemented as a SOPOT or CSD for multiplier-less realization. The square root operation on the other hand can be realized efficiently using the coordinate rotation digital computer (CORDIC) method [44].

B. PIPELINED HARDWARE IMPLEMENTATION

We now describe the pipelined implementation of the proposed D-VR-SNC-TDNLMS ABF. It is divided into six processing units and the data flow diagram is illustrated in Fig. 2. Since the normalized step size update and the regularization coefficients computation units work simultaneously, the beamformer system works in a five-stage pipelined manner. The hardware structures and functionalities of these processing units will be briefly summarized below.

1) BLOCKING MATRIX COMPUTATION UNIT

For efficient hardware implementation of a general BM, we explore the fact that all the constant multiplications can be expressed as canonical signed digit (CSD) or sum-of-power-of-two (SOPOT). More precisely, the constant coefficient are approximated by the following CSD representation

$$\lambda = \sum_{k=0}^{M-1} a_k \cdot 2^{b_k}, \quad (45)$$

where M is the number of terms used in the coefficient approximation, $a_k \in \{-1, 0, 1\}, b_k \in \{-l_k, ..., -1, 0, 1, ..., u_k\}$, and l_k and u_k are positive integers and their values determine the dynamic range of each coefficient. The larger the numbers M, l_k, and u_k, the more accurate the SOPOT approximation will be. In practice, the number of nonzero terms is usually kept to a small number to satisfy the given specification [28] so that each multiplication can be implemented by limited shifts and additions (or subtractions). Therefore, all the constant multipliers in D-VR-SNC-TDNLMS ABF can be replaced with limited shifters and adders only.

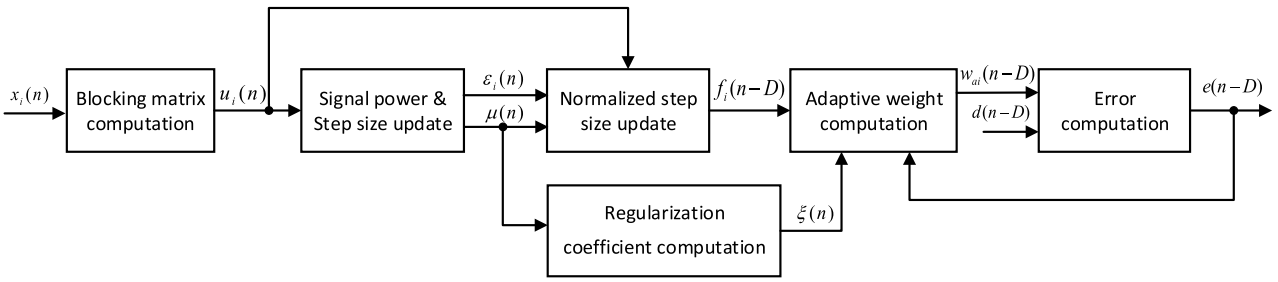


FIGURE 2. Data flow diagram of the proposed beamformer architecture.

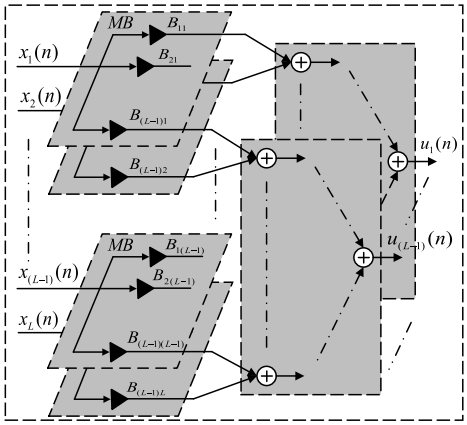


FIGURE 3. Basic structure of blocking matrix (BM) computation unit.

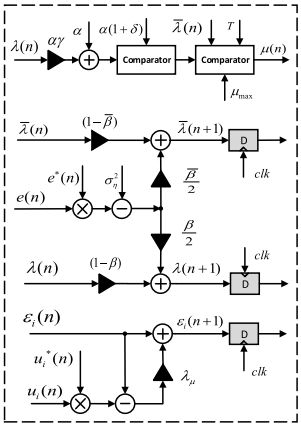


FIGURE 4. Structure of signal power and step size update unit.

To further reduce the adders required for SOPOT in the BM, the multiplier-block (MB) technique proposed in [24] is employed. MB technique is an efficient technique for reducing the number of additions in multiplying a variable input with a set of fixed coefficients in CSD or SOPOT representations. The basic principle of the MB is to reuse the intermediate results generated in realizing a group of the coefficients in order to reduce the adders for the other coefficients. Fig. 3 illustrates the realization of this unit. The triangle represents the constant multiplier. In each sub unit for computing $u_i(n)$, it costs a few adders, shifters and an adder tree. The number of adders and shifters in MB directly depend on the accuracy of SOPOT approximation. For instance, if the wordlength of $\boldsymbol{B}$ is 14-bit, the realization of this unit using MB can save 29.5% of adders. Meanwhile, the number of stages in one adder tree is directly dependent on the sensor number, $\lfloor \log_2 L \rfloor$.

2) SIGNAL POWER AND STEP SIZE UPDATE UNIT

This unit computes recursively the power of $u_i(n)$ as

$$\begin{aligned}\varepsilon_i(n+1) &= (1-\lambda_\mu)\varepsilon_i(n) + \lambda_\mu |u_i(n)|^2 \\ &= \varepsilon_i(n) + \lambda_\mu(|u_i(n)|^2 - \varepsilon_i(n)). \end{aligned} \tag{46}$$

Here, $0 < 1-\lambda_\mu < 1$ is a forgetting factor. Fig. 4 shows the power updating unit which requires $L-1$ constant multipliers with coefficient λ_μ, $2(L-1)$ adders and $(L-1)$ complex multipliers for forming $|u_i(n)|^2$, which can be implemented as $2(L-1)$ real multiplications and $(L-1)$ additions. For hardware simplicity, λ_μ is represented as SOPOT coefficients so that the constant multiplication can be implemented by shifts and additions. The output is stored in a register to facilitate the next update. The step-size is also updated in this unit according to (14a)-(14c). In the NCM mode, one comparator is required. In addition, since the VSS scheme is decided by comparing the short-term EMSE $\bar{\lambda}(n)$ against the threshold T, another comparator is also required. One complex multiplication is required to form $|e(n)|^2$, which requires two real multiplications and one addition. Five adders are also required to update the step-size. The coefficients for constant multipliers implemented are summarized in Table 2.

3) NORMALIZED STEP SIZE UPDATE UNIT

This unit computes the normalized step-size for the i-th element of the weight update equation in (42)

$$f_i(n) = \mu(n)/\varepsilon_i(n). \tag{47}$$

As shown in Fig. 5, a uniform quantizer is used to quantize $\varepsilon_i(n)$ to $\log_2(q)$ bits so as to select the appropriate value of $\hat{s}_j^{-1}$, $j \in \{1, ..., q\}$ to be multiplied to $\mu(n)$. A small look-up table is used to store $\hat{s}_j^{-1}$. The input power $\varepsilon_i(n)$ is sequentially sent to the reciprocal approximation unit and rounded result serves as the address to the ROM for retrieving $\hat{s}_j^{-1}$. This value will

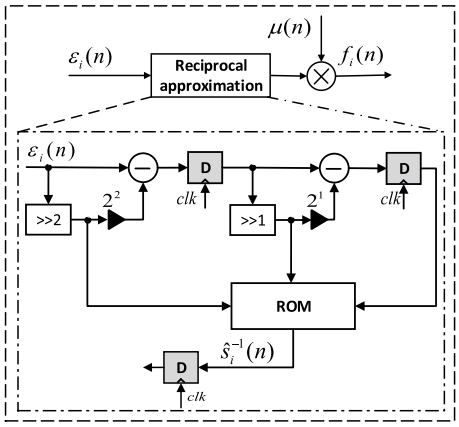


FIGURE 5. Structure of normalized step size update unit, and q = 8. D: register; >>: right shift; <<: left shift.

be multiplied to $\mu(n)$ to form the normalized step-size for the i-th component. The size of ROM directly depends on the wordlength and the number of the discretized levels, which will be introduced in Section IV-C.

4) REGULARIZATION COEFFICIENT COMPUTATION UNIT

This unit computes the regularization parameter according to (40) and its structure is shown in Fig. 6. L complex-valued multipliers, an adder tree, a real-valued constant multiplier and a square root operation are required to compute $\xi(n)$. $3L$ complex-valued multipliers, $2L$ adders, and L registers are used in this unit.

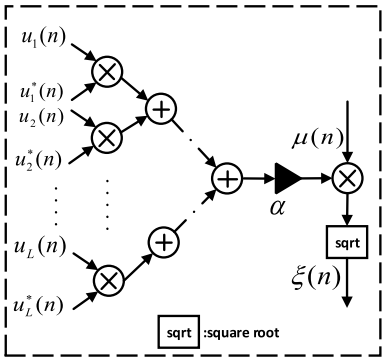


FIGURE 6. Basic structure of blocking matrix (BM) computation unit.

5) ADAPTIVE WEIGHT COMPUTATION UNIT

This unit updates the adaptive weight vector according to (42) in which $f_i(n-D)e^*(n-D)$ and hence $\boldsymbol{w}_{ai}(n+1)$ are formed and its structure for the i-th weight vector is shown in Fig. 7.

6) PREDICTED ERROR COMPUTATION UNIT

This unit computes the predicted error $e(n) = d(n) - \boldsymbol{w}_a^{\mathrm{H}}(n)\boldsymbol{u}(n)$ using L complex multipliers and an adder tree is used to accumulate the final result as shown in Fig. 8. The small grey rectangle stands for one time delay.

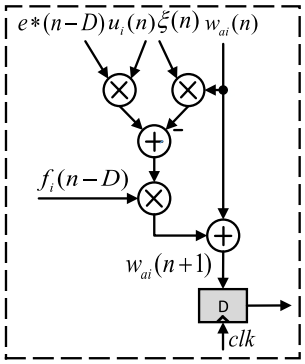


FIGURE 7. Structure of signal power and step size update unit.

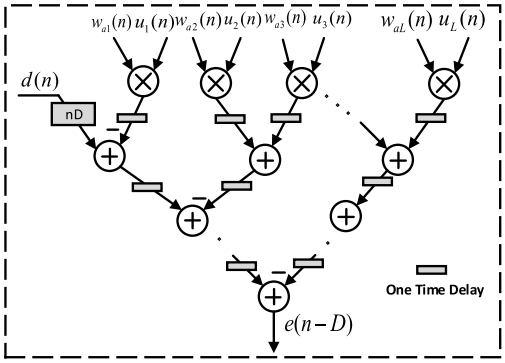


FIGURE 8. Structure of normalized step size update unit, and q = 8. D: register; >>: right shift; <<: left shift.

IV. SIMULATION AND IMPLEMENTATION RESULTS

Simulation results are now presented to evaluate the performance of the proposed D-VR-SNC-TDNLMS ABF and other state-of-the-art methods. An 8-sensor ULA with sensors separated by half wavelength with far-field scenario is considered. The SOI is located at 0°, while the two interferences are fixed at 60° and -30°. The signal to noise ratio (SNR) tested are 10 dB and 20dB, and the signal to interference (SIR) ratio is set to -10dB. All results are obtained by averaging over 100 Monte-Carlo simulations. Using the proposed architecture, a Verilog described fixed-point GSC beamformer is also designed. To evaluate performance of the hardware realization, a Matlab simulation model is studied, and the model is completely consistent with the hardware implementation. By utilizing the random search algorithm [45], the SOPOT coefficients of the parameters as shown in Table 2 are obtained. The number of quantization levels for the power normalization is $4L = 32$, and the square root operation is realized by the CORDIC IP CORE from Xilinx [46].

A. SINR PERFORMANCE COMPARISON

The signal to interference and noise (SINR)

$$\mathrm{SINR}(n) = 10\log\left(\frac{\sigma_s^2|\boldsymbol{w}^{\mathrm{H}}(n)\boldsymbol{a}(\theta_0)|^2}{\boldsymbol{w}^{\mathrm{H}}(n)\boldsymbol{R}_{\zeta\zeta}\boldsymbol{w}(n)}\right) \tag{48}$$

is used as the performance measure and σ_s^2 denotes the signal power and $\boldsymbol{R}_{\zeta\zeta}$ is the covariance matrix of the

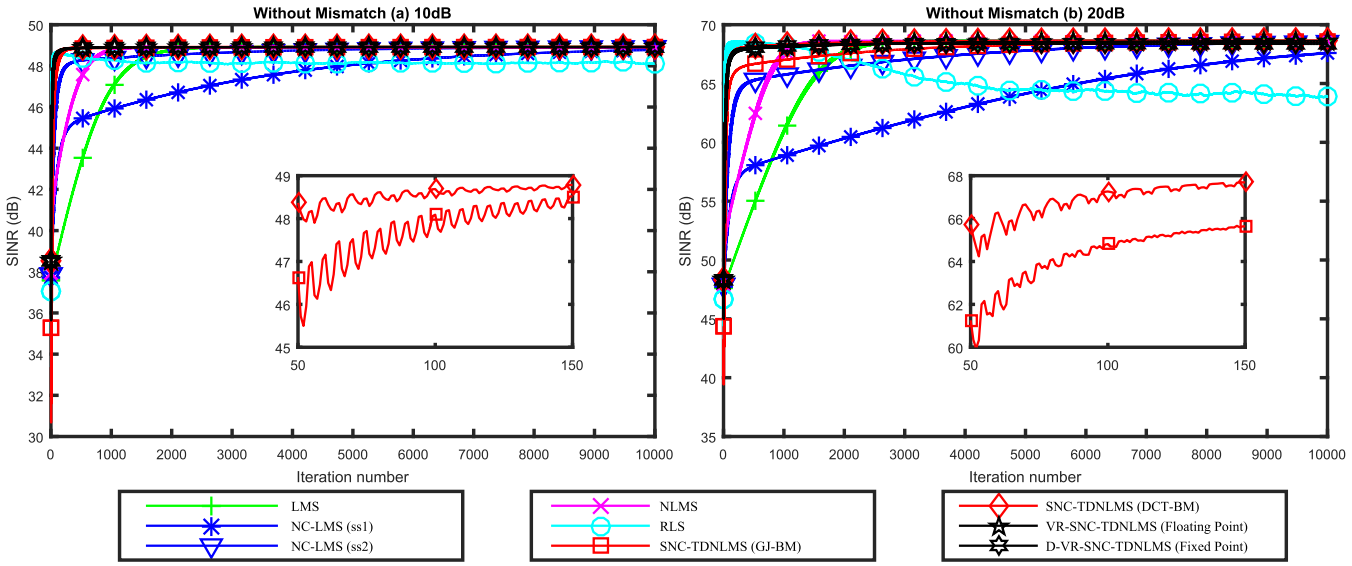


FIGURE 9. The SINR learning curves of different adaptive beamformer algorithms without steering vector mismatch. NC-LMS (ss1) and NC-LMS (ss2) correspond to the NC-LMS algorithm using initial step-sizes $1/[10(L-1)]$ and $1/[5(L-1)]$, respectively. All the algorithms are using the DCT-based blocking matrix except SNC-TDNLMS (GJ-BM) where the GJ blocking matrix is used.

interference plus noise. The algorithms tested included 1) LMS, 2) NC-LMS [11], 3) NLMS, 4) RLS, 5) SNC-TDNLMS, 6) VR-SNC-TDNLMS in floating point arithmetic and 7) VR-SNC-TDNLMS in fixed point arithmetic. The algorithm parameters are chosen as follows: the step size for the LMS and NLMS is 0.002 and $1/[4(L-1)]$ respectively so as to achieve the same SS SINR and the forgetting factor for RLS is 0.999. For the NC and SNC scheme, $\alpha, \beta, \bar{\beta}$ and γ are chosen respectively as 0.0004, 0.01, 0.05 and 20. The maximum step size for the SNC scheme is $1/(L-1)$.

a: EXPERIMENT 1

This experiment is carried out for Gaussian distributed white input in SNC-TDNLMS under different SNRs. Both the signal and interferences are of zero mean and unit variance. The EMSEs and SINRs with a noise variance mismatch factor ranging from 0.9 to 1.1 are summarized in Table 3. The theoretical SS-EMSE can be obtained from (37). By substituting the optimal beamforming vector in the SINR expression in (48), one gets the following analytical optimal SS-SINR

$$\mathrm{SINR}_{\mathrm{opt}} = 10\log(\sigma_s^2 \boldsymbol{a}^{\mathsf{H}}(\theta_0)\boldsymbol{R}_{\zeta\zeta}^{-1}\boldsymbol{a}(\theta_0)). \tag{49}$$

TABLE 3. Comparison of the steady state EMSE and SINRs for white gaussian input.

SNR		10 dB			20 dB		
Mismatch Factor		0.9	1.0	1.1	0.9	1.0	1.1
EMSE (dB)	Simulation	-46.80	-46.81	-46.83	-46.80	-46.81	-46.83
	Theory	-44.78	-44.88	-44.98	-44.77	-44.87	-44.97
SINR (dB)	Simulation	54.15	54.15	54.16	72.60	72.64	72.68
	Analytical	54.39	54.39	54.39	74.37	74.37	74.37

It can be seen that the deviation between the simulated and theoretical results in EMSE are within 2dB, which is mainly caused by the approximations used in the analysis of the last term in (22). The SINR achieved by the proposed algorithm is also within 2dB of the theoretical optimal value of the Wiener filter.

b: EXPERIMENT 2

This experiment examines the effect of using different number of discrete levels for the input power $\varepsilon_i(n)$ on the proposed VR-SNC-TDNLMS algorithm. $L/2$, L, $2L$, $4L$ and $8L$ discrete levels are evaluated and compared with the SINR performance obtained by using floating point division. Table 4 shows that the deviation decreases as the number of discrete levels increases. The situation of $L = 8$ and $L = 16$ are stimulated. Considering the complexity in hardware implementation, $4L$ discrete levels are adopted in the following experiments.

TABLE 4. Average deviations of different number of discrete levels.

L=8	Number of Levels	4	8	16	32	64
	Average Deviation	0.335	0.050	0.025	0.011	0.010
L=16	Number of Levels	8	16	32	64	128
	Average Deviation	2.896	0.376	0.067	0.036	0.013

c: EXPERIMENT 3

This experiment studies the performance of various algorithms for sinusoidal sources without mismatch at 10 dB

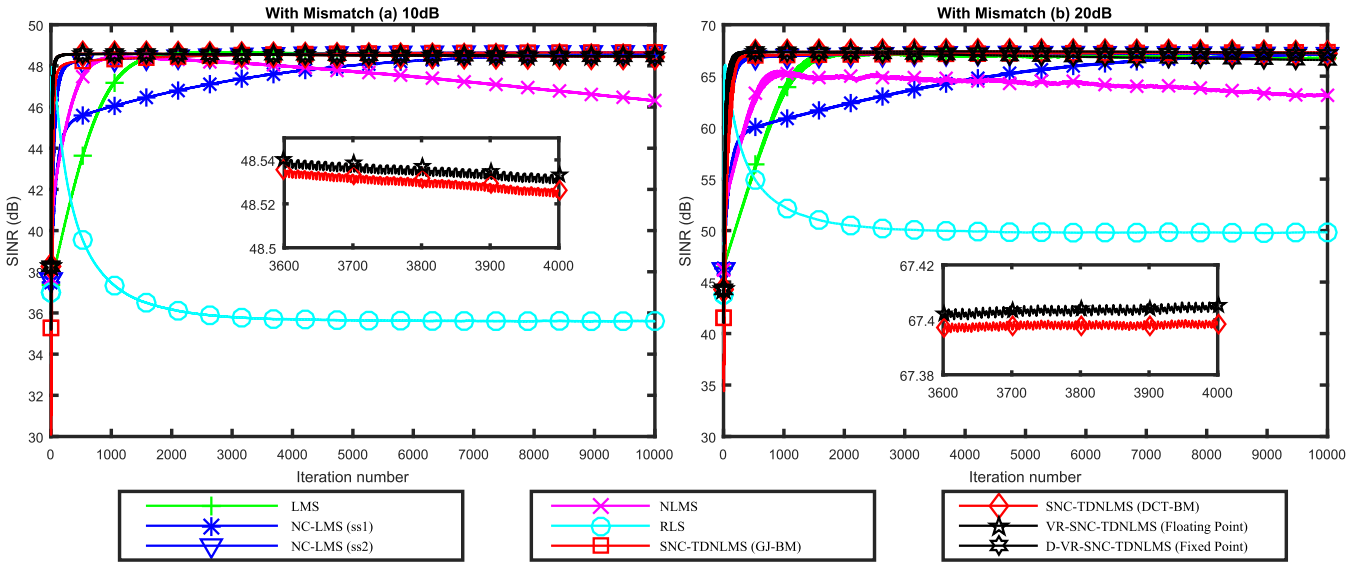


FIGURE 10. The SINR learning curves of different adaptive beamformer algorithms with steering vector mismatch. NC-LMS (ss1) and NC-LMS (ss2) correspond to the NC-LMS algorithm using initial step-sizes $1/[10(L-1)]$ and $1/[5(L-1)]$, respectively. All the algorithms are using the DCT-based blocking matrix except SNC-TDNLMS (GJ-BM) where the GJ blocking matrix is used.

and 20 dB SNR. The frequencies of the desired signal and interferences are 100 Hz, 200 Hz and 300 Hz, respectively. The learning curves of SINR are shown in Fig. 9. All the algorithms are using the DCT-based blocking matrix except SNC-TDNLMS (GJ-BM) where the GJ blocking matrix is used. To evaluate the sensitivity of the NC-LMS algorithm to the initial step-size used, two initial step-sizes denoted by (ss1: $1/[10(L-1)]$) and (ss2: $1/[5(L-1)]$) are simulated. The case ss1 corresponds to smaller step-size while ss2 corresponds to a larger step-size. It can be seen that the convergence performance of the NC-LMS is considerably degraded by the use of a smaller step-size (ss1). On the other hand, the SNC-based method has much faster convergence speed than the NC-based one due to the initial MSM and the NC-LMS method is quite sensitive to the initial step-size used. To achieve the same SINR, the LMS and NLMS algorithms are much slower than the SNC-TDNLMS algorithm. The RLS algorithm is seen to have a slightly faster speed than the SNC-based algorithm initially, its performance is rather sensitive to noise and interference, probably due to ill-conditioning. This can be improved by using the QR decomposition (QRD)-based RLS algorithm. Due to page limitation, the results are omitted. It can be seen that for the SNC-TDNLMS algorithm, using the conventional GJ-based BM is much slower than the DCT-based BM during initial convergence, probably due to the decorrelation effect of the DCT. The performances of the VR-SNC-TDNLMS and SNC-TDNLMS algorithms are almost identical.

B. PERFORMANCE UNDER DOA MISMATCH

In this experiment, the robustness of different methods with DOA mismatch is examined under different SNRs. The true desired signal is assumed to 1° while the looking direction of the array is 0°. The other settings are identical to those in the last experiment without mismatch. Fig. 10 shows the learning curves of SINR of different methods. Although the NLMS algorithm converges faster than the LMS algorithm, it is sensitive to the mismatch and degrades substantially. With appropriate initialization, NC-LMS performs comparably with the SNC-TDNLMS using GJ-BM. It should be noted that the appropriate initialization of NC-LMS requires certain experimentation and should be chosen carefully. The performance of the RLS method degrades significantly since the blocking matrix could not completely eliminate the SOI. More advanced techniques such as regularized RLS algorithm [47], [48] are necessary to mitigate the adverse effect of DOA mismatch and interested readers are referred to the literature for details. Thanks to the DCT transformation, the SNC-based algorithm using DCT-BM performs better than the GJ-BM. It can be seen from the comparison between the VR-SNC-TDNLMS and SNC-TDNLMS algorithms in the sub-plots that the VR scheme can slightly improve the steady state SINR performance.

C. HARDWARE IMPLEMENTATION

This architecture has been simulated and synthesized using Xilinx ISE 14.7 and successfully implemented on Xilinx Virtex7 (XC7VX330T) FPGA. The wordlength of all the variables are fixed at 22-bit. Eight sensors are considered in this implementation example, and the number of quantization levels for the power normalization is $4L = 32$, thus the ROM size is 5×22 bits. By utilizing the random search algorithm [45], the SOPOT coefficients of the parameters in Table 2 are obtained. In this FPGA implementation, the square root operation is realized by the CORDIC IP CORE from Xilinx [46]. The synthesized results are

TABLE 5. Proposed architecture implementation results.

	Proposed Division-free Architecture	Conventional Architecture	Improvement Percentage
Device	XC7VX330T		
Wordlength	22-bit		
LUT	19374	24953	22.36%
Slice Register	10616	17037	37.69%
DSP	276	353	21.81%
Throughput (MHz)	124.144	90.815	26.85%

summarized in Table 5. In contrast to the conventional structure using divider, our proposed structure can achieve higher working frequency. Meanwhile, the number of DSP blocks in the proposed division-free D-VR-SNC-TDNLMS ABF is reduced by around 22% over the conventional approach using hardware divider. From the simulation presented above, it is found that both methods give similar convergence and steady state performances. For the experiment 3 without mismatch, it can be seen in Fig. 9 that the performance of our proposed D-VR-SNC-TDNLMS ABF with 22-bit wordlength has a similar performance as its floating point counterpart. In terms of the performance with DOA mismatch, it is noted in Fig. 10 that the proposed fixed point VR-SNC-TDNLMS algorithm has similar performance as its floating point counterpart. Moreover, it can be seen that the proposed algorithm converges in about 150 iterations, which suggests that at a clock frequency of 124 MHz, the beam can be stabilized in around 1 microsecond. This impressive speed makes the proposed ABF and complex VR-SNC-TDNLMS algorithm valuable tools for real-time adaptive signal processing.

V. CONCLUSION

A novel ABF based on a new complex VR-SNC-TDNLMS algorithm and its efficient hardware architecture is presented. It possesses fast initial convergence speed and low steady state error due to the use of the SNC-based variable step-sizes. It also effectively mitigates signal cancellation due to steering vector mismatches through the use of variable regularization. A DCT-based BM is also introduced to improve the convergence speed in ULA. The division required in the normalization step is simplified by a new division-free approach which significantly increases the maximum operating speed and reduces hardware resources. The proposed ABF and architecture compare favorable with conventional approaches and is applicable to general adaptive filtering using LMS-like algorithms involving normalization, which makes it a valuable tool for real-time applications.

APPENDIX

Using assumption (A1) and (13), the following difference equations of the mean step-size and mean multiplier $\mathbb{E}[\lambda(n)]$ are obtained

$$\mathbb{E}[\mu(n)] = \alpha(1 + \gamma\mathbb{E}[\lambda(n)]), \tag{50}$$

$$\mathbb{E}[\lambda(n+1)] = (1-\beta)\mathbb{E}[\lambda(n)] + \beta J_b(n)/2, \tag{51}$$

where $J_b(n) = \mathbb{E}[|e(n)|^2] - a\sigma_\eta^2 = J(n) - b\sigma_\eta^2$, and $J(n) = \mathbb{E}[|e(n)|^2] - \sigma_\eta^2$ is the EMSE. Here, $b = a - 1$ is the excess noise mismatch factor and when $b = 0$, there is no noise mismatch and $J_b(n) = J(n) = \mathbb{E}[\hat{J}(n)]$.

We now evaluate the steady-state expected values of $\mu(n)$, $\mu^2(n)$, $\lambda(n)$, $\lambda^2(n)$ and get from (43) and (44) the following

$$\mathbb{E}[\mu(\infty)] = \alpha(1 + \gamma J_{*b}/2), \tag{52}$$

$$\mathbb{E}[\lambda(\infty)] = J_{*b}/2, \tag{53}$$

where J_{*b} is the steady-state value of $J_b(n)$. Similarly, $\mathbb{E}[\mu^2(n)]$ and $\mathbb{E}[\lambda^2(n)]$ are evaluated as follows

$$\mathbb{E}[\mu^2(n)] = \alpha^2(1 + 2\gamma\mathbb{E}[\lambda(n)] + \gamma^2\mathbb{E}[\lambda^2(n)]), \tag{54}$$

$$\mathbb{E}[\lambda^2(n+1)] = (1-\beta)^2\mathbb{E}[\lambda^2(n)] + \beta(1-\beta)\mathbb{E}[\lambda(n)](J(n) - b\sigma_\eta^2) + \frac{\beta^2}{4}(2J^2(n) + b_1\sigma_\eta^2 J(n) + b_2\sigma_\eta^4), \tag{55}$$

where $b_1 = 2(1-b)$, $b_2 = 2+(1+b)^2$ and we have used the fact that $\mathbb{E}[|\hat{J}(n) - b\sigma_\eta^2|^2] = \mathbb{E}[|e(n)|^4] - 2a\sigma_\eta^2 J(n) + a^2\sigma_\eta^4$
$= 2(\mathbb{E}[|\boldsymbol{v}^H(n)\boldsymbol{u}(n) + \eta|^2])^2 - 2a\sigma_\eta^2 J(n) + a^2\sigma_\eta^4$
$= 2(\mathbb{E}[|\boldsymbol{v}^H(n)\boldsymbol{u}(n)|^2] + \sigma_\eta^2)^2 - 2a\sigma_\eta^2 J(n) + a^2\sigma_\eta^4$
$= 2(J^2(n) + 2\sigma_\eta^2 J(n) + \sigma_\eta^4) - 2a\sigma_\eta^2 J(n) + a^2\sigma_\eta^4$
$= 2J^2(n) + (4-2a)\sigma_\eta^2 + (2+a^2)\sigma_\eta^4$
$= 2J^2(n) + 2(1-b)\sigma_\eta^2 + (2+(1+b)^2)\sigma_\eta^4$.
For simplicity, we have assumed that the bias due to the regularization is small so that it is ignored in the above expression. For real-valued input, the corresponding result leads

$$\mathbb{E}[\lambda^2(n+1)] = (1-\beta)^2\mathbb{E}[\lambda^2(n)] + \beta(1-\beta)\mathbb{E}[\lambda(n)](J(n) - b\sigma_\eta^2) + \frac{\beta^2}{4}(3J^2(n) + b_1\sigma_\eta^2 J(n) + b_2\sigma_\eta^4), \tag{56}$$

where $b_1 = 2(2-b)$, and $b_2 = 3 + (1+b)^2$. As the desired steady-state EMSE is usually small for VSS algorithms, we can then ignore the term $J^2(n)$ in (48). Using (47) and (48), we have

$$\mathbb{E}[\mu^2(\infty)] = \alpha^2(1 + 2\gamma\mathbb{E}[\lambda(\infty)] + \gamma^2\mathbb{E}[\lambda^2(\infty)]), \tag{57}$$

$$\mathbb{E}[\lambda^2(\infty)] \approx \frac{(1-\beta)}{2(2-\beta)}(J_* - b\sigma_\eta^2)^2 + \frac{\beta}{4(2-\beta)}(b_1\sigma_\eta^2 J_* + b_2\sigma_\eta^4), \tag{58}$$

where $J_* = \mathrm{Tr}(\boldsymbol{\Xi}_{vv}(\infty)\boldsymbol{R}_{uu})$ is the steady-state value of $J(n)$.

(53), (58), (54) and (57) correspond to (32)-(35) in the manuscript.

ACKNOWLEDGMENT

(W. Zhao, J. Q. Lin, and S. C. Chan contributed equally to this work.)

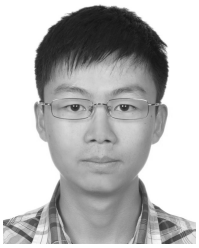
REFERENCES

[1] M. Brandstein and D. Ward, Eds., *Microphone Arrays*. Berlin, Germany: Springer, 2001.
[2] W. Liu and S. Weiss, *Wideband Beamforming: Concepts and Techniques*. Hoboken, NJ, USA: Wiley, 2010.

[3] S. M. Razavi, "Unitary beamformer designs for MIMO interference broadcast channels," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2090–2102, Apr. 2016.
[4] D. Cherkassky and S. Gannot, "New insights into the Kalman filter beamformer: Applications to speech and robustness," *IEEE Signal Process. Lett*, vol. 23, no. 3, pp. 376–380, Mar. 2016.
[5] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.
[6] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.
[7] J. Liu, A. B. Gershman, Z.-Q. Luo, and K. M. Wong, "Adaptive beamforming with sidelobe control: A second-order cone programming approach," *IEEE Signal Process. Lett.*, vol. 10, no. 11, pp. 331–334, Nov. 2003.
[8] Y.-X. Zou, B. Li, and W. Liu, "A broadband speech enhancement technique based on frequency invariant beamforming and GSC," in *Proc. Int. Conf. Green Circuits Syst. (ICGCS)*, 2010, pp. 175–179.
[9] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Autom. Control*, vol. AC-12, no. 3, pp. 282–287, Jun. 1967.
[10] S. Narayan, A. Peterson, and M. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 3, pp. 609–615, Jun. 1983.
[11] K. A. Lee and W. S. Gan, "Improving convergence of the NLMS algorithm using constrained subband updates," *IEEE Signal Process. Lett.*, vol. 11, no. 9, pp. 736–739, Sep. 2004.
[12] Y. Wei, S. B. Gelfand, and J. V. Krogmeier, "Noise-constrained least mean squares algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1961–1970, Sep. 2001.
[13] S. Zhang and J. Zhang, "New steady-state analysis results of variable step-size LMS algorithm with different noise distributions," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 653–657, Jun. 2014.
[14] S. C. Chan, Z. G. Zhang, Y. Zhou, and Y. Hu, "A new noise-constrained normalized least mean squares adaptive filtering algorithm," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Nov./Dec. 2008, pp. 197–200.
[15] H.-C. Huang and J. Lee, "A new variable step-size NLMS algorithm and its performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 2055–2060, Apr. 2012.
[16] S. Akbar *et al.*, "Novel application of FO-DPSO for 2-D parameter estimation of electromagnetic plane waves," *Neural Comput. Appl.*, 2018, doi: 10.1007/s00521-017-3318-8.
[17] F. Zaman, "Joint angle-amplitude estimation for multiple signals with l-structured arrays using bioinspired computing," *Wireless Commun. Mobile Comput.*, vol. 2017, 2017, Art. no. 9428196, doi: 10.1155/2017/9428196.
[18] M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the fractional-order Darwinian PSO," *Signal, Image Video Process.*, vol. 6, no. 3, pp. 343–350, 2012.
[19] M. S. Aslam and M. A. Z. Raja, "A new adaptive strategy to improve online secondary path modeling in active noise control systems using fractional signal processing approach," *Signal Process.*, vol. 107, pp. 433–443, Feb. 2015.
[20] N. I. Chaudhary and M. A. Z. Raja, "Identification of Hammerstein nonlinear ARMAX systems using nonlinear adaptive algorithms," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1385–1397, 2015.
[21] N. I. Chaudhary, M. A. Z. Raja, M. S. Aslam, and N. Ahmed, "Novel generalization of Volterra LMS algorithm to fractional order with application to system identification," *Neural Comput. Appl.*, vol. 29, no. 6, pp. 41–58, 2018.
[22] S. C. Chan and Y. J. Chu, "A new variable-regularized transform-domain NLMS algorithm with automatic step-size selection for adaptive system identification/filtering," *J. Signal Process. Syst.*, vol. 84, no. 2, pp. 181–196, 2016.
[23] B. Liao, S. C. Chan, and K. M. Tsui, "Recursive steering vector estimation and adaptive beamforming under uncertainties," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 1, pp. 489–501, Jan. 2013.
[24] 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.
[25] S. S. Demirsoy, I. Kale, and A. Dempster, "Reconfigurable multiplier blocks: Structures, algorithm and applications," *Circuits, Syst. Signal Process.*, vol. 26, no. 6, pp. 793–827, 2007.
[26] M. Wax and J. Sheinvald, "Direction finding of coherent signals via spatial smoothing for uniform circular arrays," *IEEE Trans. Antennas Propag.*, vol. 42, no. 5, pp. 613–620, May 1994.
[27] S. C. Chan and H. H. Chen, "Uniform concentric circular arrays with frequency-invariant characteristics—Theory, design, adaptive beamforming and DOA estimation," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 165–177, Jan. 2007.
[28] 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.
[29] S. C. Chan, K. M. Tsui, and T. I. Yuk, "Design and complexity optimization of a new digital IF for software radio receivers with prescribed output accuracy," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 351–366, Feb. 2007.
[30] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Hoboken, NJ, USA: Wiley, 1998.
[31] T. Salim, M. Trinkle, and R. Drake, "Implementation of GSC based subarray adaptive LMS algorithm using Xilinx FPGA," in *Proc. Int. Conf. Radar*, 2008, pp. 213–217.
[32] A. Tarighat *et al.*, "A low-power ASIC implementation of 2 Mbps antenna-rake combiner for WCDMA with MRC and LMS capabilities," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2005, pp. 69–72.
[33] E. H. Sarraf, M. Ahmed-Ouameur, and D. Massicotte, "FPGA implementation of beamforming receivers based on MRC and NC-LMS for DS-CDMA system," in *Proc. IEEE 17th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Sep. 2006, pp. 114–117.
[34] S. C. Chan, Z. G. Zhang, and Y. J. Chu, "A new transform-domain regularized recursive least M-estimate algorithm for a robust linear estimation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 2, pp. 120–124, Feb. 2011.
[35] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.
[36] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.
[37] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 7, pp. 1633–1642, Jul. 1992.
[38] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.
[39] S. C. Chan, Y. J. Chu, and Z. G. Zhang, "A new variable regularized transform domain NLMS adaptive filtering algorithm—Acoustic applications and performance analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 868–878, Apr. 2013.
[40] G.-H. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 9, pp. 1397–1405, Sep. 1989.
[41] L.-D. Van and W.-S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 4, pp. 359–366, Apr. 2001.
[42] L.-K. Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 1, pp. 86–95, Jan. 2005.
[43] S. Y. Park and P. K. Meher, "Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 6, pp. 346–350, Jun. 2013.
[44] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.
[45] K. S. Yeung and S. C. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 12, pp. 2444–2459, Dec. 2004.
[46] *Xilinx LogiCORE IP CORDIC v4.0*, Xilinx, San Jose, CA, USA, Mar. 2011.
[47] T. van Waterschoot, G. Rombouts, and M. Moonen, "Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement," *Signal Process.*, vol. 88, no. 3, pp. 594–611, 2008.
[48] S. C. Chan and Y. J. Chu, "A new state-regularized QRRLS algorithm with a variable forgetting factor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 3, pp. 183–187, Mar. 2012.

W. 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 the City, University of 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. His research activities and interests include array signal processing, adaptive filtering, digital hardware design, and embedded system.



J. Q. LIN received the bachelor's degree in information engineering from Beihang University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in digital signal processing with the Department of Electrical and Electronic Engineering, The University of Hong Kong.

His current research interests include array signal processing, subspace identification, adaptive filtering, and mathematics.

S. C. CHAN (S'87–M'92) received the B.Sc. (Eng.) and Ph.D. degrees from The University of 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, multirate and biomedical signal processing, communications and array signal processing, high-speed A/D converter architecture, bioinformatics, smart grid, and image-based rendering.

Dr. Chan is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was the Chair of the IEEE Hong Kong Chapter of Signal Processing from 2000 to 2002 and an Organizing Committee Member of the 2003 IEEE ICASSP and the 2010 IEEE ICIP. He served as an Associate Editor for *Digital Signal Processing*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I from 2008 to 2009, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 2012 to 2016. He is currently an Associate Editor of the *Journal of Signal Processing Systems*.

H. K.-H. SO (S'03–M'07–SM'15) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1998, 2000, and 2007, respectively. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. He received the Croucher Innovation Award in 2013 for his work in power-efficient high-performance heterogeneous computing system. He received the University Outstanding Teaching Award (Team) in 2012 and the Faculty Best Teacher Award in 2011.