



IDBD-Based Beamforming Algorithm for Improving the Performance of Phased Array Radar in Nonstationary Environments

Shihan Wang ^{1,2}, Tao Chen ¹ and Hongjian Wang ^{1,*}

- ¹ National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China
- ² School of Earth and Planetary Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
- * Correspondence: wanghongjian@mirslab.cn; Tel.: +86-010-62586454

Abstract: Adaptive array processing technology for a phased array radar is usually based on the assumption of a stationary environment; however, in real-world scenarios, nonstationary interference and noise deteriorate the performance of the traditional gradient descent algorithm, in which the learning rate of the tap weights is fixed, leading to errors in the beam pattern and a reduced output signal-to-noise ratio (SNR). In this paper, we use the incremental delta-bar-delta (IDBD) algorithm, which has been widely used for system identification problems in nonstationary environments, to control the time-varying learning rates of the tap weights. The designed iteration formula for the learning rate ensures that the tap weights adaptively track the Wiener solution. The results of numerical simulations show that in a nonstationary environment, the traditional gradient descent algorithm with a fixed learning rate has a distorted beam pattern and reduced output SNR; however, the IDBD-based beamforming algorithm, in which a secondary control mechanism is used to adaptively update the learning rates, showed a similar beam pattern and output SNR to a traditional beamformer in a Gaussian white noise background; that is, the main beam and null satisfied the pointing constraints, and the optimal output SNR was obtained. Although the proposed algorithm contains a matrix inversion operation, which has considerable computational complexity, this operation could be replaced by the Levinson-Durbin iteration due to the Toeplitz characteristic of the matrix; therefore, the computational complexity could be decreased to O(n), so additional computing resources are not required. Moreover, according to some intuitive interpretations, the reliability and stability of the algorithm are guaranteed.

Keywords: phased array radar; beamforming; IDBD algorithm; nonstationary environment

1. Introduction

Phased array radar, in which the main beam and null are pointed toward the expected direction and beam scanning is realized by weighting the amplitudes and phases of each array element [1,2], plays an important role in gain control, resolution enhancement, and anti-jamming maximization [3,4] and has been widely used in radar, navigation, sonar, and communication [5–8]. Beamforming, which is one of the key techniques for phased array radar, allows users to adjust the direction of the main beam and nulls and, therefore, perform target detection, tracking, and interference suppression [4,9–11]. Thus, beamforming has attracted considerable attention from researchers.

Several studies have been performed on adaptive signal processing. Traditional algorithms, such as the minimum variance distortionless response (MVDR), linearly constrained minimum variance (LCMV), and frequency invariant beamforming (FIB) for wideband phased array radar, are usually based on the assumption of a stationary environment with fixed error performance surfaces [12]. In this case, an acceptable performance could be obtained by using a traditional gradient descent algorithm with a fixed learning rate. However, in practice, many environments are nonstationary [12,13]. In these nonstationary



Citation: Wang, S.; Chen, T.; Wang, H. IDBD-Based Beamforming Algorithm for Improving the Performance of Phased Array Radar in Nonstationary Environments. *Sensors* **2023**, *23*, 3211. https://doi.org/10.3390/s23063211

Academic Editor: Seongjoo Lee

Received: 2 March 2023 Revised: 13 March 2023 Accepted: 14 March 2023 Published: 17 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). environments, the probability density function is non-Gaussian and the autocorrelation function and power spectra are time-varying, which might result in distortions in the beam pattern and deterioration in the output signal-to-noise ratio (SNR) if a fixed learning rate is used to update the tap weights [12].

Several studies have investigated target tracking, detection system identification, and pattern classification in nonstationary environments [14–18]. However, few studies have investigated beamforming in nonstationary environments. In [19], L. R. Zhang et al. proposed an adaptive beamforming algorithm to address beam pattern distortion due to coloured noise. In [20], fractional lower-order moment theory was applied to improve the performance of spatial and Doppler target localization. Ref. [21] studied an adaptive array processing method in a non-Gaussian environment by replacing the Gaussian data assumption with a multivariate elliptically contoured distribution. In [22], statistics and physics were used to analyse the eigenvalues of the sample covariance matrix, which might influence the performance of a beamformer.

In this study, the incremental delta-bar-delta (IDBD) algorithm, in which the learning rates of the tap weights of each element are different and time-varying [16–18] to track the continuously time-varying position of the optimal solution, is used to address the beamforming problem in a nonstationary environment. Moreover, the performance of the IDBD-based beamforming algorithm is compared with that of a fixed learning rate beamformer. Furthermore, the reliability, stability, and computational complexity of the proposed algorithm are analysed intuitively.

The remainder of this paper is organized as follows. In Section 2, the signal and system model of the narrowband beamformer are reviewed. In Section 3, the constrained cost function is established via Frost's algorithm. Then, based on the IDBD algorithm, the iteration formula for each tap weight is summarized, and the reliability, stability, and computational complexity of the proposed algorithm are analysed intuitively. In Section 4, the numerical simulation results of the IDBD-based beamforming algorithm and those of a beamformer with a fixed learning rate are compared. Finally, the conclusions and some discussions are presented in Section 5.

2. Signals and Systems Modelling

Suppose that the signals of interest are narrowband; then, these signals can be denoted by their steering vectors. Without loss of generality, consider a uniform linear array (ULA) consisting of M omnidirectional antennas followed by M tap weights. The structure of the proposed narrowband beamformer is shown in Figure 1.



Figure 1. Structure of the narrowband beamformer.

Suppose we have K signal sources spanning a column vector $\mathbf{s} = [\mathbf{s}_0(t), \mathbf{s}_1(t) \dots \mathbf{s}_{K-1}(t)]^T$ with the direction of arrival (DOA) vector $\theta = [\theta_0, \theta_1 \dots \theta_{K-1}]^T$. The steering matrix of these K signal sources can be written as $\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1 \dots \mathbf{a}_{K-1}]^T$, where

$$\mathbf{a}_{k} = \left[1, \exp\left(-j\frac{2\pi}{\lambda}d\sin\theta_{k}\right), \cdots \exp\left(-j\frac{2\pi}{\lambda}(M-1)d\sin\theta_{k}\right)\right]^{1}$$
(1)

is the steering vector of the *k*-th signal source. Then, the vector spanned by the signals received by each antenna can be written as

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} + \mathbf{v} \tag{2}$$

where $\mathbf{x} = [x_0, x_1 \dots x_{M-1}]^T$, $\mathbf{n} = [n_0, n_1 \dots n_{M-1}]^T$ and $\mathbf{v} = [v_0, v_1 \dots v_{M-1}]^T$ denote the vectors spanned by the total signals, AWGN, and nonstationary noise received by each antenna.

After the beamformer processes these signals, the output signal can be written as

$$y = \mathbf{w}^{\mathrm{H}}\mathbf{x} \tag{3}$$

where $\mathbf{w} = [w_0, w_1 \dots w_{M-1}]^T$ denotes the vector spanned by the tap weights. Thus, the total output power of the beamformer is

$$P_{\text{total}} = E|y|^2 = E\left(\mathbf{w}^{\text{H}}\mathbf{x}\mathbf{x}^{\text{H}}\mathbf{w}\right) = \mathbf{w}^{\text{H}}\mathbf{R}\mathbf{w}$$
(4)

where $\mathbf{R} = E(\mathbf{x}\mathbf{x}^{H})$ is the autocorrelation matrix of the input signal. According to the LCMV criterion, in which the direction of the main beam and null are guaranteed and the total output power is minimized, the objective function can be written as

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{W}} \mathbf{W}^{\mathrm{H}} \mathbf{R} \mathbf{w}$$

s.t. $\mathbf{A}^{\mathrm{H}} \mathbf{w} = \mathbf{f}$ (5)

According to this formula, we can determine the optimal tap weights analytically using the Lagrange multiplier method as follows:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{A} (\mathbf{A}^{\text{H}} \mathbf{R} \mathbf{A})^{-1} \mathbf{f}$$
(6)

3. IDBD-Based Beamformer

Equation (6) includes two matrix inverse operations, which are computationally complex. To reduce the complexity of this equation, which cannot be applied to higherdimensional arrays, we determine the optimal tap weights iteratively using gradient descent. In this constrained optimization problem, Frost [23] noted that the cost function in the n-th iteration should be formulated via the Lagrange multiplier, namely,

$$L(n) = \mathbf{w}^{\mathrm{H}}(n)\mathbf{R}\mathbf{w}(n) + \lambda^{\mathrm{H}}(n)\left(\mathbf{A}^{\mathrm{H}}\mathbf{w}(n) - \mathbf{f}\right) + \lambda^{\mathrm{T}}(n)\left(\mathbf{A}^{\mathrm{T}}\mathbf{w}^{*}(n) - \mathbf{f}^{*}\right)$$
(7)

where $\lambda(n)$ is an M-by-1 vector that denotes the Lagrange multiplier in the *n*-th iteration and the superscript * denotes complex conjugate. Through Wirtinger calculus, the gradient of the cost function *L* with respect to the tap weights is

$$\mathbf{g}(n) = \frac{\partial L(n)}{\partial \mathbf{w}^*(n)} = \mathbf{R}\mathbf{w}(n) + \mathbf{A}\boldsymbol{\lambda}(n)$$
(8)

In traditional algorithms, each tap weight is updated using the same time-invariant learning rate ε , namely,

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \varepsilon \mathbf{g}(n) \tag{9}$$

However, in the IDBD algorithm, each tap weight is updated according to different time-varying learning rates, which can be expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{\Xi}(n)\mathbf{g}(n) \tag{10}$$

where $\Xi(n) = \text{diag}(\varepsilon 0(n), \varepsilon 1(n) \dots \varepsilon M - 1(n))$ denotes a diagonal matrix spanned by the learning rate of each tap weight in the n-th adaptation cycle.

To satisfy the constraint described in Equation (6) in each adaptation cycle, the parameter $\lambda(n)$ needs to be adjusted as follows:

$$\mathbf{A}^{\mathrm{H}}\mathbf{w}(n+1) = \mathbf{A}^{\mathrm{H}}\mathbf{w}(n) - \mathbf{A}^{\mathrm{H}}\Xi(n)\mathbf{g}(n) = \mathbf{f}, \forall n$$
(11)

Substituting Equation (8) into Equation (11), we can analytically solve the Lagrange multiplier in each iteration cycle as follows:

$$\lambda(n) = \left(\mathbf{A}^{\mathrm{H}} \mathbf{\Xi}(n) \mathbf{A}\right)^{-1} \left(\mathbf{A}^{\mathrm{H}} \mathbf{w}(n) - \mathbf{A}^{\mathrm{H}} \mathbf{\Xi}(n) \mathbf{R} \mathbf{w}(n) - \mathbf{f}\right)$$
(12)

Thus, the relationship between the cost function and the time-varying learning rate can be obtained; therefore, $\Xi(n)$ can be updated adaptively via the block diagram shown in Figure 2.



Figure 2. Block diagram of the IDBD algorithm.

Equation (12) shows that the solution of the Lagrange multiplier contains a matrix inversion operation, which has considerable computational complexity; however, due to the Toeplitz characteristic of the matrix $\mathbf{A}^{\mathrm{H}} \Xi(n) \mathbf{A}$, the Levinson–Durbin iteration could be used to solve λ . Because this operation has a computational complexity of O(*n*), additional computing resources are not required [12].

The learning rates are controlled by an adaptable memory parameter, denoted as β , and the relationship between the two parameters in the system identification problem is formulated as [12]

$$\varepsilon_m(n) = \exp(\beta_m(n)) \tag{13}$$

However, the beamforming problem includes several complex operations; thus, to ensure that the learning rates are always real numbers, in this paper, we rewrite Equation (13) as

$$\varepsilon_m(n) = \exp(\beta_m(n) + \beta_m^*(n)) \tag{14}$$

Thus, the cost function is formulated as a function of the adaptable memory parameter, allowing us to control the learning rate matrix $\Xi(n)$ by updating β via gradient descent with a fixed learning rate κ . Next, we show the mathematical derivations.

The differentiation of the cost function with respect to the m-th tap weight in the n-th adaptation cycle is

$$g_m(n) = \frac{\partial L(n)}{\partial w_m^*(n)} = \sum_{l=0}^{M-1} (R_{ml} w_l(n) + A_{ml} \lambda_l(n))$$

= $\mathbf{R}(m,:) \mathbf{w}(n) + \mathbf{A}(m,:) \lambda(n)$ (15)

where A(m,:) and R(m,:) denote the *m*-th row of matrix A and R, respectively. Then, the updated formula for the *m*-th tap weight can be expressed as

$$w_m(n+1) = w_m(n) - \varepsilon_m(n)(\mathbf{R}(m, :)\mathbf{w}(n) + \mathbf{A}(m, :)\lambda(n))$$
(16)

Based on this formula, the gradient of the cost function with respect to the adaptable memory parameter can be determined using the chain rule for differentiation, namely,

$$\frac{\partial L(n)}{\partial \beta_m^*} = \frac{\partial L(n)}{\partial w_m^*(n)} \cdot \frac{\partial w_m^*(n)}{\partial \beta_m^*} = g_m(n)h_m(n) \tag{17}$$

where $h_m(n)$ denotes the differentiation of the *m*-th tap weight with respect to the *m*-th adaptable memory parameter in the *n*-th adaptation cycle. This equation is difficult to solve analytically. However, Equations (14) and (16) suggest the following relationship:

$$h_m(n+1) = \frac{\partial w_m^*(n+1)}{\partial \beta_m^*} = \frac{\partial}{\partial \beta_m^*} (w_m(n) - \varepsilon_m(n)g_m(n))^* = (18)$$

$$h_m(n) - \varepsilon_m(n)g_m^*(n) - \varepsilon_m(n)\frac{\partial g_m^*(n)}{\partial \beta_m^*}$$

Thus, we can determine $h_m(n)$ iteratively.

Substituting Equations (12) and (15) into Equation (18), and in order to make the mathematical form more convenient, let auxiliary variables

$$\mathbf{C}(n) = \left(\mathbf{A}^{\mathrm{T}} \Xi(n) \mathbf{A}^{*}\right)^{-1}$$

$$\mathbf{D}(n) = \mathbf{I} - \Xi(n) \mathbf{R}^{*}$$
 (19)

Then, the differentiation of $g_m(n)$ with respect to $\beta_m(n)$ can be written as

$$\frac{\partial g_m^*(n)}{\partial \beta_m^*} = \frac{\partial}{\partial \beta_m^*} (\mathbf{R}(m, :) \mathbf{w}(n) + \mathbf{A}(m, :) \lambda(n))^* = R_{mm} h_m(n) + \mathbf{A}^*(m, :) \frac{\partial}{\partial \beta_m^*} (\mathbf{C}(n) \left(\mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \mathbf{w}^*(n) - \mathbf{f} \right))$$
(20)

Moreover, the differentiation in the fourth line of Equation (20) can be expanded as three terms:

$$\frac{\partial}{\partial\beta_m} \left(\mathbf{C}(n) \left(\mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \mathbf{w}^*(n) - \mathbf{f} \right) \right) = \frac{\partial \mathbf{C}(n)}{\partial\beta_m} \left(\mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \mathbf{w}^*(n) - \mathbf{f} \right) - \mathbf{C}(n) \mathbf{A}^{\mathrm{T}} \frac{\partial\Xi(n)}{\partial\beta_m} \mathbf{R}^* \mathbf{w}^*(n) + \mathbf{C}(n) \mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \frac{\partial \mathbf{w}^*(n)}{\partial\beta_m} \right)$$
(21)

Using the matrix differential formula, we obtain

$$\frac{\partial \Xi(n)}{\partial \beta_m} = \varepsilon_m(n) \begin{bmatrix} \mathbf{O} & & \\ & \mathbf{I} \\ & & \mathbf{O} \end{bmatrix}$$
(22)

$$\frac{\partial \mathbf{C}}{\partial \beta_m} = -\mathbf{C} \mathbf{A}^{\mathrm{T}} \frac{\partial \Xi(n)}{\partial \beta_m} \mathbf{A}^* \mathbf{C} = -\varepsilon_m(n) \mathbf{C} (\mathbf{A}(m,:))^{\mathrm{T}} \mathbf{A}^*(m,:) \mathbf{C}$$
(23)

$$\frac{\partial \mathbf{w}^*(n)}{\partial \beta_m} = \left[\mathbf{0}, h_m(n), \mathbf{0}\right]^{\mathrm{T}}$$
(24)

Thus, Equation (20) can be further simplified as

$$\frac{\partial}{\partial \beta_m} \left(\mathbf{C}(n) \left(\mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \mathbf{w}^*(n) - \mathbf{f} \right) \right) = -\varepsilon_m(n) \mathbf{C} \left(\mathbf{A}(m,:) \right)^{\mathrm{T}} \mathbf{A}^*(m,:) \\ \mathbf{C} \left(\mathbf{A}^{\mathrm{T}} \mathbf{D}(n) \mathbf{w}^*(n) - \mathbf{f} \right) \\ -\varepsilon_m(n) \mathbf{C} \left(\mathbf{A}(m,:) \right)^{\mathrm{T}} \mathbf{R}^*(m,:) \mathbf{w}^*(n) \\ + h_m(n) \mathbf{C} \mathbf{A}^{\mathrm{T}} \mathbf{D}(:,m)$$
(25)

where **D**(:,*m*) denotes the *m*-th column of matrix **D**. Substituting Equations (20) and (24) into Equation (18), the gradient of the cost function with respect to the adaptable memory parameter in each iteration cycle can be obtained, and the updated formula for $\beta_m(n)$ can be written as

$$\beta_m(n+1) = \beta_m(n) + \kappa \frac{\partial L}{\partial \beta_m^*(n)}$$
(26)

where κ is a fixed learning rate for updating $\beta_m(n)$. Thus, by substituting Equation (26) into Equation (14), the time-varying learning rate for the m-th tap weight in the n-th adaptation cycle can be adaptively controlled. Moreover, Equations (18)–(26) show that by choosing this relationship between ε and β , β becomes proportional to the product of the current weight change and the trace of recent weight changes, as shown in Equation (14). Therefore, after several iterations, the overall change in β becomes proportional to the correlation between the current and recent weight changes, which suggests that the previous steps were larger if the current learning rate is positively correlated with past learning rates, and vice versa. According to this intuitive statement, the iteration of β is controlled by previous states, indicating that the update direction of the tap weights is corrected adaptively; thus, the proposed algorithm is reliable and stable. A similar analysis is provided in [18].

Summarizing the above mathematical derivations, the pseudocode for the IDBD-based beamformer is presented in Algorithm 1.

Algorithm 1 Summary of the IDBD-based beamformer.
begin initialize $\mathbf{w}(0), \boldsymbol{\Xi}(0), \mathbf{h}(0), m \leftarrow 0, n \leftarrow 0$
input A, κ , N
while $n \leq N$
compute $C(n)$, $D(n)$, $\lambda(n)$ with Equations (12) and (19)
compute $g_m(n)$ with Equation (15)
compute $\partial g_m^*(n) / \partial \beta_m^*$ with Equation (25)
compute $\partial L(n) / \partial \beta_m^*$ with Equation (17)
do
$h_m(n+1) \leftarrow h_m(n) - \varepsilon_m(n)g_m^*(n) - \varepsilon_m(n)\frac{\partial g_m^*(n)}{\partial \beta_m^*}$
$eta_m(n+1) \leftarrow eta_m(n) + \kappa rac{\partial L}{\partial eta_m^*(n)}$
$\varepsilon_m(n+1) \leftarrow \exp(2\operatorname{Re}(\beta_m(n+1)))$
$\mathbf{w}(n+1) \leftarrow \mathbf{w}(n) - \mathbf{\Xi}(n) \mathbf{g}(n)$
$m \leftarrow m+1$
until $m = M - 1$
$n \leftarrow n+1$
end

4. Numerical Simulation

In the numerical simulation of the algorithm, we chose a ULA with 16 antenna elements and an element spacing of 1/2-times the wavelength of the carrier wave. Suppose that the target signal is a linear frequency-modulated (LFM) signal with a pulse width of 10 μ s and a bandwidth of 12 MHz, and that the power and DOA of the target signal are 0 dB and 30°, respectively. We choose three interference signals with powers of 5 dB: sine waves with frequencies of 8 GHz, 9 GHz, and 9.5 GHz and incident DOAs of -60° , -30° ,

and 0° , respectively. The powers of the AWGN and nonstationary signals are 3 dB and 15 dB, respectively.

The nonstationary noise was randomly chosen from the heavy sine, bumps, and Doppler signals in each Monte Carlo experiment. The waveform and autocorrelation functions of these nonstationary noises are shown in Figure 3. The figures show that for each time delay, the autocorrelation functions of these signals vary with the initial observation time, which indicates the nonstationary property of these signals.



Figure 3. Common nonstationary signal waveforms and their autocorrelation functions (panels (**a**,**c**,**e**) are waveform of the heavy sine signal, bumps signal and Doppler signal, and panels (**b**,**d**,**f**) are autocorrelation function of the heavy sine signal, bumps signal and Doppler signal).

The autocorrelation matrix of the received signal x is computed by replacing the ensemble average with an arithmetic average. The tap weights are initialized randomly, and the diagonal matrix spanned by the time-varying learning rates is initialized to 0.005-times the unitary matrix. Furthermore, to compare the performance of the IDBD-based beamformer with that of a beamformer with a fixed learning rate of 0.005, we studied the performance of the two algorithms in a stationary environment with an AWGN signal with the same power. Figure 4 shows the average results of 100 Monte Carlo experiments, and in each of the experiments, the snapshot is 128.



Figure 4. Comparison of the performance of the two beamforming algorithms in stationary and nonstationary environments (panel (**a**) is beam pattern, panel (**b**) indicates output SNR of the IDBD-based beamformer in stationary and nonstationary environments and the fixed learning rate beamformer in a stationary environment, and panel (**c**) represents output SNR of the fixed learning rate beamformer in a nonstationary environment).

In Figure 4, the blue dotted line, cyan circle, red dotted line, and pink dotted line denote the performance of the IDBD-based beamformer in nonstationary and stationary environments and the performance of the beamformer with a fixed learning rate in non-stationary and stationary environments, respectively. We plotted the SNR curve of the traditional beamformer in a nonstationary environment in a separate graph to ensure that the changes in the three SNR curves could be shown clearly.

Figure 4 shows that in a stationary environment, the IDBD-based beamformer and fixed learning rate beamformer show approximately the same performance, namely, the main beam and null point toward the expected direction and the optimal SNRs are output. However, in a nonstationary environment, although the nulls satisfied the pointing constraint for the beamformer with a fixed learning rate, the direction of the main beam was distorted, and the output SNR decreased dramatically as the number of iterations increased. This might be because when the surface of the cost function L moves in the weight space due to the time-varying property of auto-correlation matrix R, we still update the weight vector along a fixed direction. Therefore, the algorithm cannot converge to the optimal solution and result in the distortion of the radiation pattern and deterioration of the output SNR.

In contrast, for the IDBD-based beamformer, the beam pattern and SNR showed similar performance to the IDBD-based beamformer in a stationary environment. Moreover, Figure 4b shows that the convergence rate of the IDBD-based beamforming algorithm is as fast as the convergence rates of both beamformers in a stationary environment. A similar result was discussed in [17], revealing the superiority of the IDBD-based beamforming algorithm.

In conclusion, the traditional beamforming algorithm with a fixed learning rate and the IDBD-based beamforming algorithm both showed acceptable performance in a stationary environment; that is, the main beam pointed toward the target direction, the null point pointed toward the interference, and the optimal output SNR was obtained. However, in an environment with high-power nonstationary noise, the traditional beamformer suffers from beam pattern distortions and a reduced output SNR, and an IDBD-based beamformer is needed to address these issues.

5. Conclusions and Discussions

Several studies have shown that in nonstationary environments, algorithms with fixed learning rates may fail to converge to the optimal solution, resulting in a reduced performance [20–22], which is reflected in beam pattern distortion and a decreased SNR in beamforming problems. Therefore, for a beamformer to achieve its optimal tracking capability, the beamformer must first pass from transient mode to steady-state mode, and the filter's free parameters must be continuously adjusted [12]. By introducing time-varying learning rates for each weight, the algorithm can track statistical changes in the environment in real time.

We considered an LFM target signal and compared the performance of the IDBD-based beamforming algorithm with that of a beamformer with a fixed learning rate in stationary and nonstationary environments. The results indicate that the IDBD-based beamformer has several advantages.

First, in contrast with the beamformer with a fixed learning rate, which had beam pattern distortion and a decreased SNR in nonstationary environments, the IDBD-based beamforming algorithm performs similarly to a traditional beamformer in a stationary environment; that is, the main beam and null pointed toward the expected direction, and the optimal output SNR was obtained.

Second, in a nonstationary environment, the convergence rate of the IDBD-based beamforming algorithm was as fast as that of the beamformer with a fixed learning rate in a stationary environment, which indicates that the introduction of the adaptable memory parameter does not deteriorate the rate of convergence.

Finally, Equations (12), (19) and (25) show that in the IDBD-based beamforming algorithm, although each tap weight adaptation cycle includes a matrix inversion operation, the Toeplitz characteristic of the matrix $\mathbf{A}^{\mathrm{H}} \Xi(n) \mathbf{A}$ indicates that this problem can be solved using the Levinson–Durbin iteration, which has a computational complexity of O(*n*). This complexity is the same as that of the beamforming algorithm with a fixed learning rate; thus, additional computing resources are not required.

Due to these advantages, we suggest that the IDBD-based beamformer should be implemented in extremely harsh electromagnetic environments, such as magnetic storms, in which the background electromagnetic noises showed similar characteristics of the simulated environment used in this paper, namely non-Gaussian and nonstationary characteristics [24].

Author Contributions: Conceptualization, H.W.; writing—original draft preparation, S.W.; mathematical derivation and MATLAB coding, S.W.; supervision, H.W. and T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China: 42130202, 41931073; International Partnership Program of Chinese Academy of Sciences: 183311KYSB20200003; Strategic Pioneer Program on Space Science, the Chinese Academy of Sciences: XDA15350201; Pandeng Program of National Space Science Center, Chinese Academy of Sciences; Ground-Based

Space Environment Monitoring Network (Meridian Project II); Specialized Research Fund for State Key Laboratories.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: There are no new data.

Acknowledgments: The authors would like to thank Lei Li for editing this manuscript.

Conflicts of Interest: The authors declare that they have no competing interest.

References

- 1. Schelkunoff, S.A. A mathematical theory of linear arrays. Bell Syst. Tech. J. 1943, 22, 80–107. [CrossRef]
- 2. Jaeckle, W. Antenna synthesis by weighted Fourier coefficients. IEEE Trans. Antennas Propag. 1964, 12, 369–370. [CrossRef]
- 3. Liao, G.S.; Tao, H.H.; Zeng, C. Digital Beamforming Technique of Phased Array Radar; National Defence Industry Press: Beijing, China, 2017.
- 4. Mailloux, R.J. *Phased Array Antenna Handbook*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2008.
- 5. Hamici, Z. Fast Beamforming with Fault-Tolerance in Massive Phased-Arrays using Intelligent Learning Control. *IEEE Trans. Antennas Propag.* **2019**, *67*, 4517–4527. [CrossRef]
- Iskander, M.F.; Kim, W.; Bell, J.; Celik, N.; Yun, Z.; Youn, H.S. Antenna arrays technologies for advanced wireless systems. In Proceedings of the IEEE International Conference on Microwaves, Tel Aviv, Israel, 9–11 November 2010.
- Gong, Y.; Wang, R.; Wang, P. Improved Phase-Encoding Calibration for Active Phased-Array Antennas of SAR. *IEEE Geosci. Remote Sens. Lett.* 2017, 13, 767–771. [CrossRef]
- 8. Younis, M.; Rommel, T.; Bordoni, F.; Krieger, G.; Moreira, A. On the Pulse Extension Loss in Digital Beamforming SAR. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1436–1440. [CrossRef]
- 9. Sun, C.; Hirata, A.; Ohira, T.; Karmakar, N.C. Fast beamforming of electronically steerable parasitic array radiator antennas: Theory and experiment. *IEEE Trans. Antennas Propag.* **2004**, *52*, 1819–1832. [CrossRef]
- 10. Bach, H. Directivity of basic linear arrays. IEEE Trans. Antennas Propag. 2003, 18, 107–110. [CrossRef]
- 11. Guo, L.; Deng, H.; Himed, B.; Ma, T.; Geng, Z. Waveform Optimization for Transmit Beamforming With MIMO Radar Antenna Arrays. *IEEE Trans. Antennas Propag.* 2015, 63, 543–552. [CrossRef]
- 12. Haykin, S. Adaptive Filter Theory, 5th ed.; Publishing House of Electronics Industry: Pearson, CA, USA, 2002.
- 13. de Souza, D.B.; Kuhn, E.V.; Seara, R.A.A. Time-Varying Autoregressive Model for Characterizing Nonstationary Processes. *IEEE* Signal Process. Lett. 2019, 26, 134–138. [CrossRef]
- 14. Akcakaya, M.; Sen, S.; Nehorai, A. A Novel Data-Driven Learning Method for Radar Target Detection in Nonstationary Environments. *IEEE Signal Process. Lett.* **2016**, *23*, 762–766. [CrossRef]
- Zhang, Q.; Wang, M.; Lu, Y.; Idrees, M.; Zhang, L. Fast Nonstationary Noise Tracking Based on Log-Spectral Power MMSE Estimator and Temporal Recursive Averaging. *IEEE Access* 2019, 7, 80985–80999. [CrossRef]
- Mahmood, A.R.; Sutton, R.S.; Degris, T.; Pilarski, P.M. Tuning-free step-size adaptation. In Proceedings of the 2012 IEEE International Conference on Acoustics, Kyoto, Japan, 25–30 March 2012.
- 17. Jacobs, R.A. Increased Rates of Convergence through Learning Rate Adaptation. Neural Netw. 1988, 1, 295–307. [CrossRef]
- 18. Sutton, R.S. Adapting bias by gradient descent: An incremental version of delta-bar-delta. Proc. AAII-92 1992, 171–176.
- 19. Zhang, L.R.; Liao, G.S. Adaptive beamforming in colored noise environment. *Acta Electron. Sin.* **1998**, *26*, 75–78.
- 20. Tsakalides, P.; Nikias, C.L. Robust adaptive beamforming in alpha-stable noise environments. In Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Atlanta, GA, USA, 9 May 1996.
- Richmond, C.D. Adaptive array processing in non-Gaussian environments. In Proceedings of the IEEE Signal Processing Workshop on Statistical Signal & Array Processing, Corfu, Greece, 24–26 June 1996.
- 22. Cox, H. Adaptive beamforming in non-stationary environments. Presented at the Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 3–6 November 2003.
- 23. Frost, O.L. An Algorithm for Linearly Constrained Adaptive Array Processing. Proc. IEEE 1972, 60, 926–935. [CrossRef]
- Wang, S.H.; Li, L.; Chen, T.; Ti, S.; Cai, C.L.; Li, W.; Luo, J. Statistical Characterization of the Magnetic Field in Space during Magnetic Storms. *Atmosphere* 2022, 13, 1578. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.