remote sensing

MDPI

Article

A Novel Beam-Domain Direction-of-Arrival Tracking Algorithm for an Underwater Target

Xianghao Hou [1,2,3,*], Weisi Hua [1,2,3], Yuxuan Chen [1,2,3] and Yixin Yang [1,2,3]

[1] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; huaweisi@mail.nwpu.edu.cn (W.H.); crainx@mail.nwpu.edu.cn (Y.C.); yxyang@nwpu.edu.cn (Y.Y.)
[2] Shaanxi Key Laboratory of Underwater Information Technology, Xi'an 710072, China
[3] Hanjiang National Laboratory, Xi'an Research Center, Xi'an 710072, China
* Correspondence: houxianghao1990@nwpu.edu.cn

Abstract: Underwater direction-of-arrival (DOA) tracking using a hydrophone array is an important research subject in passive sonar signal processing. In this study, a DOA tracking algorithm based on a novel beam-domain signal processing technique is proposed to ensure robust DOA tracking of an interested underwater target under a low signal-to-noise ratio (SNR) environment. Firstly, the beam-based observation is designed and proposed, which innovatively applies beamforming after array-based observation to achieve specific spatial directivity. Next, the proportional–integral–differential (PID)-optimized Olen–Campton beamforming method (PIDBF) is designed and proposed in the beamforming process to achieve faster and more stable sidelobe control performance to enhance the SNR of the target. The adaptive dynamic beam window is designed and proposed to focusing the observation on more likely observation area. Then, by utilizing the extended Kalman filter (EKF) tracking framework, a novel PIDBF-optimized beam-domain DOA tracking algorithm (PIDBF-EKF) is proposed. Finally, simulations with different SNR scenarios and comprehensive analyses are made to verify the superior performance of the proposed DOA tracking approach.

Keywords: adaptive beamforming; extended Kalman filter; PID control method; underwater target tracking

check for updates

Citation: Hou, X.; Hua, W.; Chen, Y.; Yang, Y. A Novel Beam-Domain Direction-of-Arrival Tracking Algorithm for an Underwater Target. *Remote Sens.* **2024**, *16*, 4074. https://doi.org/10.3390/rs16214074

Academic Editor: Gabriel Vasile

Received: 29 August 2024
Revised: 22 October 2024
Accepted: 28 October 2024
Published: 31 October 2024

Copyright: © 2024 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/).

1. Introduction

The ocean is considered the largest public space on Earth's surface. Additionally, the ocean is the origin of numerous security threats. In modern naval warfare, passive bearing angle estimation of underwater non-cooperative targets is a pivotal field of study. Passive bearing angle estimation methods are classified into two categories: direction-of-arrival (DOA) estimation and DOA tracking [1–3]. DOA estimation and tracking determine the real-time bearing of underwater targets, enabling real-time monitoring of specific maritime areas.

The DOA estimation method assumes that the target's bearing angle remains unchanged or changes slightly during the observation time. This approach determines the target's bearing by processing the information contained in array-received signals. However, DOA estimation methods neglect the motion information of the target [4–7], which results in poor bearing estimation in low signal-to-noise ratio (SNR) environments. In complex underwater environments, with non-stationary, highly non-cooperative targets and low SNR, traditional DOA estimation methods struggle to provide effective results. Consequently, continuous DOA tracking is required for the accurate estimation of moving underwater targets.

To overcome the drawbacks of the traditional DOA estimation methods, DOA tracking techniques not only utilize array-received signals but also incorporate the target's motion characteristics to achieve more accurate and stable results than traditional DOA estimation methods. Researchers have extensively studied the DOA tracking field. Typically, DOA

tracking methods use signals received by sonar arrays as input and develop motion and measurement models based on typical underwater target movements. Due to the significant nonlinearity between target bearing and original array signal measurements, these models are applied with nonlinear Bayesian filtering theory to recursively estimate the target's bearing. Researchers have developed various tracking algorithms based on nonlinear Bayesian filtering for DOA tracking, a type of nonlinear parameter estimation problem. The extended Kalman filter (EKF) is the most representative of these. Kong [8] used EKF to derive a target bearing tracking method using array-received signals, achieving effective target tracking. In real underwater target tracking scenarios, in addition to the target's acoustic signals, uncertain noise from marine turbulence, distant ships, storms, and surface waves can significantly affect array measurements. Zhang [9] used an improved Sage–Husa algorithm to estimate the covariance matrix of measurement noise in real-time, obtaining robust underwater target DOA tracking results against strong disturbances. Hou [10] enhanced the extended Kalman filter with variational Bayesian methods and incorporated machine learning in the initial covariance matrix [11], conducting extensive research on DOA tracking based on raw array measurements. These studies verify the feasibility of array-based DOA tracking, which can achieve continuous high-precision tracking of underwater moving targets. However, array-based observations have not been processed to focus on specific spatial directions, leading to potential signal drowning in noise under low SNR or strong interference conditions. Additionally, neglecting the spatial sparsity of target signals and the large number of array elements can result in reduced computational efficiency in DOA tracking.

Beamforming functions as a spatial filter, suppressing signals from non-target directions and enhancing those from target directions. To achieve spatial directivity and improve accuracy and computational efficiency of DOA tracking, beam-domain DOA tracking is required. This method employs beamforming to generate weight vectors, resulting in observations with spatial directivity. By focusing on specific spatial directions, this approach effectively increases the SNR of received signals. Conventional beamforming, exemplified by the 1965 Bartlett algorithm (conventional beamforming, CBF) [12], is computationally simple but constrained by the "Rayleigh limit", resulting in poor resolution, poor array gain, and the inability to adapt. Adaptive beamforming, such as the MVDR beamformer introduced by Capon in 1969 [13], offers improved spatial resolution but fails to control the main lobe and sidelobes. In practice, received signals are often drowned in environmental noise and cluttered with interference from multiple directions. Thus, a beamforming method with low sidelobes is essential to effectively reduce interference within the sidelobe areas and enhance directional accuracy. Additionally, beam-domain tracking can introduce errors. Each observation should improve the SNR within the target angle range and suppress signals outside this range, necessitating the use of low-sidelobe beamforming techniques. In conventional beamforming, the most well-known method to reduce sidelobes is the Dolph–Chebyshev (DC) weighting [4], which achieves the lowest sidelobe levels given a set main lobe width, but which is limited to ULA and lacks adaptability. In adaptive beamforming, Ma [14] first introduced an optimization method for antenna patterns of sensors with any structural shape. This method, by placing virtual interferers of appropriate intensity in the sidelobe area, achieves desired low sidelobes. Following the same principle, Olen [15] proposed a digital synthesis method for static beam patterns. This method control sidelobes through iterative virtual noise sources, which can reduce sidelobes to a predetermined level set by us. It also allows for the pre-calculation and storage of beamforming weights. However, Olen only used simple difference control to iterative virtual noise sources. Many researchers have improved this approach [16–18], enhancing algorithm convergence and precision. However, these methods, primarily using single controls, like difference and proportional control, often face issues with oscillations, slow convergence, excessive iterations, and high computational demands.

Building on the above discussion, to address the limitations caused by traditional DOA estimation methods overlooking target motion characteristics and array-based DOA

tracking algorithms neglecting real-time spatial information of targets, we propose a new tracking algorithm. Initially, to achieve observations with spatial directivity, we innovatively design and propose beam-based observation. Beamforming is applied after element domain observation, resulting in beam-based observations with specific spatial directivity. By performing beamforming in the element domain, the SNR in suspected target directions is enhanced, thereby improving DOA tracking accuracy. Next, to obtain the beam-domain observations, a new beamforming method and an adaptive dynamic beam window are innovatively designed and proposed. To suppress signals from non-target directions and enhance signals from the target direction for better SNR, the beamforming method needs to effectively reduce the sidelobe levels of the beam. To be applicable in various array scenarios, the beamforming method needs to be adaptable to different array configurations. To reduce real-time computation, the beamforming method should allow for the pre-calculation and storage of beamforming weight vectors. The new beamforming method, namely the proportional–integral–differential (PID)-optimized Olen–Campton beamforming method (PIDBF) can meet the above three conditions; at the same time, it can also achieve faster and more stable low-sidelobe control effects. The Olen–Campton beamforming method (OBF) and its improvements adopt single control strategies, such as difference control and proportional control during the error control process of these virtual noise sources. This approach results in slow sidelobe reduction, excessive iterations, and high computational demands. To overcome these issues, the PIDBF introduces PID control techniques on the basis of OBF and its improvements, achieving faster and more stable low-sidelobe control effects. To obtain the beam-domain observations, an adaptive dynamic beam window is also designed and proposed. The adaptive dynamic beam window uses the predicted angle as the beam center direction. The root mean square of the angular error determines the beam window width. This creates dynamic feedback for the beam window, allowing real-time adjustments to its position and size. This approach improves the capture of target directional characteristics and regulates the SNR of the received signals. Then, a DOA tracking algorithm based on a beam-domain observation model is proposed. This algorithm is framed within EKF and utilizes a PID-optimized Olen–Campton beamforming method, which is named PIDBF-EKF. Firstly, we obtain measurement information from a uniform linear array (ULA) regarding the target's radiation noise and establish a typical kinematic model for underwater moving targets. Additionally, we design and propose PIDBF to achieve faster and more stable sidelobe control. We also design and propose an adaptive dynamic beam window to obtain better beam-based observations. Finally, we integrate PIDBF into the EKF tracking filter framework and propose the PIDBF-EKF for robust DOA tracking. The innovative contributions of this work are summarized as follows:

(1) To achieve observations with spatial directivity, beam-based observation is innovatively designed and proposed. Beamforming is applied after element domain observation, resulting in beam-based observations with specific spatial directivity. By performing beamforming in the element domain, the SNR in suspected target directions is enhanced, thereby improving DOA tracking accuracy.

(2) To obtain the beam-domain observations, the PIDBF is designed and proposed. By introducing the PID control technique, proportional control, integral control, and derivative control are used to iteratively adjust virtual noise sources, achieving faster and more stable sidelobe control so that the SNR of the interested target can be enhanced. The PIDBF is adaptable to different array configurations to be applicable in various array scenarios. The PIDBF also allows for pre-calculation and storage of beamforming weight vectors to reduce real-time computation. Simulation verification of the proposed beamforming algorithm under different SNR conditions shows that the PIDBF can achieve stable and rapid low sidelobe control. To obtain the beam-domain observations, an adaptive dynamic beam window is also designed and proposed. The adaptive dynamic beam window uses the predicted angle as the beam center direction. The root mean square of the angular error

determines the beam window width. This creates dynamic feedback for the beam window, allowing real-time adjustments to its position and size.

(3) The PIDBF-EKF is designed and proposed. PIDBF-EKF is a real-time iterative updating beam-domain DOA tracking framework based on "beamforming + nonlinear filtering", where EKF is incorporated based on beam-based observations. This framework overcomes the limitations of traditional DOA estimation and DOA tracking for underwater moving targets under low SNR conditions, enhancing the spatial directivity and computational efficiency of the tracking algorithm. Simulation verification of the proposed beam-domain DOA tracking algorithm under different SNR conditions and different numbers of beams demonstrates that beam-domain DOA tracking offers higher tracking accuracy than traditional DOA estimation and element-domain DOA tracking under low SNR conditions. PIDBF-EKF enhances computational efficiency while maintaining the same tracking accuracy as other beam-domain DOA tracking methods.

The remainder of this paper is organized as follows: In Section 2, the problem of underwater DOA tracking is formulated. In Section 3, the PIDBF is proposed. The beam-domain DOA tracking algorithm PIDBF-EKF is designed in Section 4. In Section 5, simulations with various conditions are conducted and the effectiveness of the proposed algorithms are verified.

2. Problem Formulation

2.1. Kinematic Model of the Underwater Target by the Bearing Angle

In passive tracking, the target is relatively far from the hydrophone array, and the target's movement speed is slow. The movement of far-field targets under the constant velocity (CV) model can be regarded as uniform angular velocity motion. The target state is characterized by the bearing angle and rate of angle change, forming the basis for the two-dimensional CV motion model, as follows:

$$\boldsymbol{X}_k = \left(\theta_k, \dot{\theta}_k\right)^T \tag{1}$$

where θ_k represents the target's bearing angle, and $\dot{\theta}_k$ denotes the angular velocity. The CV motion model can be expressed as follows:

$$\boldsymbol{X}_k = \boldsymbol{F}_{k|k-1}\boldsymbol{X}_{k-1} + \boldsymbol{G}_k\boldsymbol{w}_k \tag{2}$$

where $\boldsymbol{X}_k = \left(\theta_k, \dot{\theta}_k\right)^T$ represents the system state at tracking step k ($k = 1, 2, 3, \ldots, K$). Here, K is the total number of steps. $\boldsymbol{F}_{k|k-1}$ is the transition matrix from step $k-1$ to step. $\boldsymbol{G}_k$ is the noise driving matrix of the CV motion model at step k. $\boldsymbol{w}_k$ is the zero-mean Gaussian process noise caused by the unknown underwater environment, with a covariance matrix $\boldsymbol{Q}_k$. The matrices $\boldsymbol{F}_{k|k-1}$ and $\boldsymbol{G}_k$ are expressed as follows:

$$\boldsymbol{F}_{k|k-1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

$$\boldsymbol{G}_k = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \tag{4}$$

where T is the interval between adjacent tracking steps.

2.2. Measurement Model Based on the Received Signal of the Uniform Linear Array

The configuration of the ULA and the underwater target is shown in Figure 1.

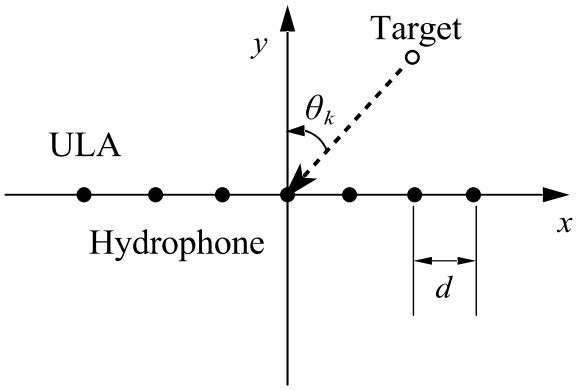


Figure 1. Configuration of the ULA-based measurement system.

The narrowband acoustic signal emitted by the target has a frequency of f, an amplitude of a, and an initial phase of ϕ_0. T represents the observation time interval. The target signal at position k is expressed as follows:

$$s(k) = a\exp(j(2\pi fkT + \phi_0)) \tag{5}$$

The array-received signal also includes a clutter model. The clutter narrowband signal has a frequency f_c, an amplitude of b, and an initial phase of ϕ_c. The clutter model at position k is expressed as follows:

$$s_c(k) = b\exp(j(2\pi f_c kT + \phi_c)) \tag{6}$$

Assuming the target's range satisfies the far-field condition, the arriving signal is considered a plane wave. With the sound speed c, the received signal at time k can be expressed as follows:

$$\boldsymbol{z}(k) = real(\boldsymbol{A}(\theta_k)s(k) + \boldsymbol{A}(\theta_c)s_c(k)) + \boldsymbol{V}_k = \boldsymbol{r}(k) + real(\boldsymbol{A}(\theta_c)s_c(k)) + \boldsymbol{V}_k \tag{7}$$

where $\boldsymbol{A}(\theta_k)$ represents the array manifold in the direction of θ_k, $\boldsymbol{A}(\theta_c)$ represents the array manifold in the direction of θ_c, θ_c represents the direction of the clutter, and $\boldsymbol{V}_k$ is the measurement noise following a Gaussian distribution. The expression for $\boldsymbol{A}(\theta_k)$ and $\boldsymbol{A}(\theta_c)$ are given by the following equations:

$$\boldsymbol{A}(\theta_k) = \begin{bmatrix} a_1(\theta_k) & a_2(\theta_k) & \cdots & a_P(\theta_k) \end{bmatrix}^T \tag{8}$$

$$\boldsymbol{A}(\theta_c) = \begin{bmatrix} a_1(\theta_c) & a_2(\theta_c) & \cdots & a_P(\theta_c) \end{bmatrix}^T \tag{9}$$

where P is the number of elements in the ULA, and $a_p(\theta_k)$ and $a_p(\theta_c)$ are defined as follows:

$$a_p(\theta_k) = \exp(-j2\pi(p-1)d\sin(\theta_k)/\lambda) \tag{10}$$

$$a_p(\theta_c) = \exp(-j2\pi(p-1)d\sin(\theta_c)/\lambda) \tag{11}$$

where $p = 1, 2, \cdots, P$, d is the spacing between elements in the ULA, and λ is the wavelength. The array received signal over an observation time (with M snapshots) is given as the following element-domain observation:

$$\boldsymbol{Z}_k = \begin{bmatrix} \boldsymbol{z}(k) \\ \boldsymbol{z}(k+\tau) \\ \cdots \\ \boldsymbol{z}(k+(M-1)\tau) \end{bmatrix} = \begin{bmatrix} real(\boldsymbol{A}(\theta_k)s(k) + \boldsymbol{A}(\theta_c)s_c(k)) + \boldsymbol{V}_k \\ real(\boldsymbol{A}(\theta_k)s(k+\tau) + \boldsymbol{A}(\theta_c)s_c(k+\tau)) + \boldsymbol{V}_k \\ \cdots \\ real(\boldsymbol{A}(\theta_k)s(k+(M-1)\tau) + \boldsymbol{A}(\theta_c)s_c(k+(M-1)\tau)) + \boldsymbol{V}_k \end{bmatrix} \tag{12}$$

where M is the number of snapshots, and τ is the reciprocal of the sampling rate, representing the duration of one snapshot.

3. Principle of the PIDBF

Array-based observations possess spatial omnidirectionality and cannot focus on specific spatial directions of interest. To achieve observations with spatial directivity, we innovatively design and propose beam-based observation. Beamforming is applied after element domain observation, resulting in beam-based observations with specific spatial directivity. We propose a new beamforming method that introduces PID control technology in the process of artificially adding virtual interference sources to control the sidelobe level in the Olen–Campton beamforming method (OBF) [15]. This approach which is called the proportional–integral–differential (PID)-optimized Olen–Campton beamforming method (PIDBF) achieves a faster and more stable sidelobe reduction in the beamforming process. The following subsections will demonstrate the principle of the OBF and PIDBF, which is designed and proposed based on the OBF.

3.1. Olen–Campton Beamforming Method and Optimization

To enhance the spatial directivity and interference suppression capabilities of beamforming, it is essential to achieve lower sidelobe levels, which effectively suppress interference within the sidelobe area and reduce the system's false alarm probability. The Olen–Campton beamforming method addresses this by generating a beam response map and artificially introducing virtual interference sources in the sidelobe region. This approach enables adaptive adjustment of the intensity of these interference sources, facilitating the reduction of sidelobes.

First, J virtual interference sources are artificially added in the sidelobe region. σ_j^2 and $\theta_j (j = 1, 2, \cdots, J)$ represent the intensity and direction of the j-th interference source, respectively. The steady-state interference covariance matrix is obtained through matrix operations, as shown in the following Equation (13):

$$\boldsymbol{R}_{i+n} = \sum_{j=1}^{J} \sigma_j^2 \boldsymbol{a}(\theta_j) \boldsymbol{a}^H(\theta_j) + \sigma^2 \boldsymbol{I} \tag{13}$$

where the subscripts i and n in index $\boldsymbol{R}_{i+n}$ are abbreviations for the words interference and noise, respectively, indicating that $\boldsymbol{R}_{i+n}$ includes two components: the noise covariance matrix and the interference source covariance matrix. $\boldsymbol{a}(\theta_j)$ is the array response vector corresponding to direction θ_j, σ^2 is the power of the additive white noise of the element, and $\boldsymbol{I}$ is the identity matrix.

Under the current conditions of the interference source's intensity and direction, the beamforming weight vector is given by the following:

$$\boldsymbol{w}_{opt} = \frac{\boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}(\theta_s)}{\boldsymbol{a}^{\mathrm{H}}(\theta_s) \boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}(\theta_s)} \tag{14}$$

where θ_s is the beam pointing angle, and $\boldsymbol{a}(\theta_\mathrm{s})$ is the array manifold of the base array. The interference sources can be uniformly distributed across the entire scanning bearing, with no interference sources located within the main lobe. Consequently, the intensity of the interference sources in the main lobe is set to zero, while the intensity of the interference sources in the sidelobe region is continuously and adaptively adjusted.

Suppose the main lobe region is $[\theta_L(n), \theta_R(n)]$ during the n-th adaptive adjustment. Then, for the next adjustment, the intensity of the interference sources σ_j^2 in the Olen–Campton beamforming method is set as follows:

$$\sigma_j^2(n+1) = \begin{cases} 0 & \theta_j \in [\theta_L(n), \theta_R(n)] \\ max[0, \Gamma_j(n)] & others \end{cases} \tag{15}$$

$$\Gamma_j(n) = \sigma_j^2(n) + K[p(\theta_j, n) - D(\theta_j)] \, , j = 1, 2 \ldots J \tag{16}$$

The obtained beam pattern needs to be compared with the desired beam pattern. When the sidelobe level in a certain direction exceeds the desired value, the intensity of the interference source in that direction is increased. Conversely, if the sidelobe level is below the desired value, the interference source's intensity is decreased. In Equation (16), $p(\theta_j, n)$ is the normalized beam response in direction θ_j obtained after the n-th adjustment, $D(\theta_j)$ is the desired beam response in direction θ_j, and K is the adaptive iterative gain, which affects the speed of iterative convergence. The initial value of the interference source intensity can be set to zero.

OBF and its optimization have three advantages. Firstly, the beam has lower sidelobe levels to effectively suppress interference within the sidelobe area and reduce the system's false alarm probability. Secondly, they are adaptable to different array configurations to be applicable in various array scenarios. Thirdly, by using the method of artificially adding virtual interference sources, OBF and its optimization can iteratively obtain the beamforming weight vectors without obtaining array signals in advance. So, OBF and its optimization allow for the pre-calculation and storage of beamforming weight vectors to reduce real-time computation. However, there is a problem still to be solved.

The iterative steps of the Olen–Campton beamforming method utilize difference regulation, which can lead to slow iteration speeds and oscillation problems. Zhang and Yang have each optimized the interference source iterative steps of the Olen–Campton beamforming method [14]. Zhang's optimization, known as the Zhang–Olen–Campton beamforming method (ZOBF), transforms the nonlinear problem into a linear one through segmentation but still uses a single Differential control method to manage the error between the current sidelobe and the desired sidelobe. The choice of coefficients relies on empirical values, which may lack theoretical significance to some extent. Optimization of the Olen–Campton beamforming method proposed by Yang (YOBF) uses a proportional control method, which can increase the stability but still has room for improvement in speed of iterations.

3.2. Design of the PIDBF Method

In addressing the problem, we design and propose PIDBF, which introduces PID control in the interference source iteration steps of the Olen–Campton beamforming method for optimization and improvement. As a widely used error-regulating controller, the PID algorithm can effectively regulate the system's desired output through end-to-end control. Its core principle involves computing the control output by weighting and summing the proportional, integral, and derivative elements of the control deviation (the gap between the set value and the actual value), thereby effectively correcting the deviation of the controlled object to achieve optimal control effect and ensuring the controlled object reaches a stable state. By introducing PID control into the process of artificially adding virtual interference sources for low-sidelobe management in beamforming, we enable rapid calibration to achieve the desired low-sidelobe beam under certain main lobe power conditions, thereby enhancing the SNR in regions where targets may exist in space. Initially, the sidelobe error between the current beam and the desired beam is calculated in real-time to obtain the PID control input. Then, using the characteristics of proportional control to increase the speed of sidelobe control, integral control is adopted to eliminate static sidelobe control errors, and derivative control is utilized to improve the dynamic performance of sidelobe control. Consequently, low-sidelobe beamforming in beam-domain specific spatial regions can be achieved, providing a measurement basis for subsequent DOA tracking.

The intensity of the interference source σ_j^2 in the beamforming method is set as follows:

$$\sigma_j^2(n+1) = \begin{cases} 0 & \theta_j \in [\theta_L(n), \theta_R(n)] \\ K_j(n+1)\sigma_j^2(n) & others \end{cases} \tag{17}$$

where the initial value of $\sigma_j^2(n)$ should be non-zero. Otherwise, the interference source intensity will be zero in each iteration.

Equation (18) is as follows:

$$error(n) = \frac{D(\theta_j)}{p(\theta_j, n)} - 1, \;\; j = 1, 2, \cdots, J \tag{18}$$

where $D(\theta_j)$ is the desired sidelobe level, and $p(\theta_j, n)$ is the sidelobe level obtained in the current iteration. Equation (18) is used to obtain the proportional error term $error(n)$ in the n-th iteration.

The iterative gain factor of the n-th interference source after the n-th adjustment $K_j(n+1)$ is set as follows:

$$K_j(n+1) = \min\left\{K_{\max}, 1 + K_p \cdot error(n) + K_i \cdot errorsum + K_d \cdot (error(n) - error(n-1))\right\} \tag{19}$$

$$errorsum = \sum_{i=1}^{n} error(i) \tag{20}$$

where $K_{\max}$ is the maximum allowable value of the iterative gain factor setting to prevent divergence in the adaptive process. $error(n)$ is the proportional error term in the n-th iteration. The three parameters K_p, K_i, K_d correspond to the three parameters of the PID controller. K_p is the proportional control parameter, K_i is the integral control parameter, and K_d is the derivative control parameter. *errorsum* is the sum of the proportional error amounts over n iterations.

4. Principle of the PIDBF-EKF

Considering the drawbacks of the traditional DOA estimation and element-domain DOA tracking, traditional DOA estimation methods neglect the kinematic information of the tracking target. Element-domain DOA tracking directly employs raw array information as measurements without considering the spatial sparsity of target information. A beam-domain DOA tracking algorithm based on low sidelobe PID-controlled beamforming with artificially added virtual interference sources is proposed, which incorporates artificially added virtual interference sources to enhance performance.

This algorithm aims to achieve more robust DOA tracking of underwater targets. The algorithm consists of four steps, as outlined below.

4.1. Target State Time Update

Based on Equation (2), the one-step state transition equation is given by the following Equation (21):

$$\hat{\boldsymbol{X}}_{k|k-1} = \boldsymbol{F}_k \hat{\boldsymbol{X}}_{k-1|k-1} \tag{21}$$

The prediction equation for the state covariance matrix $\boldsymbol{P}_{k|k-1}$ is given by the following:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k{}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{22}$$

where $\boldsymbol{P}_{k-1|k-1}$ is the state covariance matrix at time $k-1$, and $\boldsymbol{Q}_k$ is the noise covariance matrix.

4.2. Calculation of the Beamforming Matrix

To obtain the beam-domain observations, an adaptive dynamic beam window is designed and proposed. We obtain the given target predicted state $\hat{\boldsymbol{X}}_{k|k-1}$ and state covariance matrix $\boldsymbol{P}_{k-1|k-1}$ through Equation (22). The adaptive dynamic beam window uses the predicted angle $\hat{\theta}_{k|k-1}$ as the beam center direction. The root mean square of the angular error $\sqrt{\boldsymbol{P}_{k|k-1}(1,1)}$ determines the beam window width. This creates dynamic feedback for the

beam window, allowing real-time adjustments to its position and size. The beam window size $\boldsymbol{B}_k$ is created and proposed by us as follows:

$$\boldsymbol{B}_k = \left(\hat{\theta}_{k|k-1} - \sqrt{\boldsymbol{P}_{k|k-1}(1,1)},\ \hat{\theta}_{k|k-1} + \sqrt{\boldsymbol{P}_{k|k-1}(1,1)}\right) \tag{23}$$

Since the predicted angle $\hat{\theta}_{k|k-1}$ and the state covariance matrix $\boldsymbol{P}_{k|k-1}$ are updated in real-time, dynamic feedback is provided to the beamforming window, which adjusts the central angle and size of the window in real-time to regulate the SNR of the received signal. When the angular error of the state covariance matrix $\boldsymbol{P}_{k|k-1}(1,1)$ is large, the beam window size will increase accordingly to observe a larger area of space; when $\boldsymbol{P}_{k|k-1}(1,1)$ is small, the beam window size will decrease accordingly, focusing the observation on the area of interest. We set the number of beams N to be uniformly distributed within the beam window. The obtained beamforming matrix is denoted as $\boldsymbol{W}_k^N$, which can be represented as follows:

$$\boldsymbol{W}_k^N = \begin{bmatrix} \boldsymbol{w}^1 \\ \cdots \\ \boldsymbol{w}^N \end{bmatrix} \tag{24}$$

where $\boldsymbol{w}^n$ $(n = 1, 2, \cdots, N)$ represents the weight of the n-th beam. The weight matrix of the PIDBF is given by Equation (14) The iterative method for the interference source intensity σ_j^2 is shown in Equations (17)–(19).

It should be noted that as the observation time increases, $\boldsymbol{P}_{k|k-1}(1,1)$ may converge to a very small value. This can lead to an excessively small observation window, resulting in minimal changes in the beam angle and poor beamforming performance. Therefore, a threshold ε is set to prevent the window from shrinking beyond a certain point. Since the state covariance matrix $\boldsymbol{P}_{k|k-1}$ and the predicted angle $\hat{\theta}_{k|k-1}$ are updated in real time, this provides dynamic feedback to the beamforming window, allowing real-time adjustments to the window position and size, thereby improving the SNR of the received signal.

4.3. Calculation of Kalman Filter Gain

The optimal Kalman gain $\boldsymbol{K}_k$ is given by the following:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\widetilde{\boldsymbol{H}}_k^{\mathrm{T}}\left(\widetilde{\boldsymbol{H}}_k\boldsymbol{P}_{k|k-1}\widetilde{\boldsymbol{H}}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1} \tag{25}$$

where $\boldsymbol{R}_k$ is the observation noise covariance matrix, and $\widetilde{\boldsymbol{H}}_k$ is the Jacobian matrix of the observation equation. For the array observation model shown in Equation (12), under single snapshot conditions, the Jacobian matrix can be expressed as Algorithm 1.

$$\boldsymbol{h}(k) = \begin{bmatrix} \frac{\partial \boldsymbol{z}(k)}{\partial \theta} \\ \frac{\partial \boldsymbol{z}(k)}{\partial \omega} \end{bmatrix} \tag{26}$$

$$\frac{\partial \boldsymbol{z}(k)}{\partial \theta} = \begin{bmatrix} \frac{\partial \boldsymbol{A}^1(\theta_{k|k-1})}{\partial \theta} & \frac{\partial \boldsymbol{A}^2(\theta_{k|k-1})}{\partial \theta} & \cdots & \frac{\partial \boldsymbol{A}^p(\theta_{k|k-1})}{\partial \theta} \end{bmatrix} \tag{27}$$

$$\frac{\partial \boldsymbol{A}^p(\theta_{k|k-1})}{\partial \theta} = \exp\left(\left(\frac{-\mathrm{j}2\pi d(P-1)\sin(\theta_{k|k-1})}{\lambda}\right)\left(\frac{-\mathrm{j}2\pi d(P-1)\cos(\theta_{k|k-1})}{\lambda}\right)\right) \tag{28}$$

$$\frac{\partial \boldsymbol{z}(k)}{\partial \omega} = \begin{bmatrix} 0_1 & 0_2 & \cdots & 0_P \end{bmatrix} \tag{29}$$

Algorithm 1: PIDBF-EKF for DOA tracking using ULA

Inputs: $\hat{\boldsymbol{X}}_{k|k-1}$, $\boldsymbol{P}_{k-1|k-1}$, $\widetilde{\boldsymbol{Z}}_k$.
Target State Time Update
1. $\hat{\boldsymbol{X}}_{k|k-1} = \boldsymbol{F}_k \hat{\boldsymbol{X}}_{k-1|k-1}$.
2. $\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} {\boldsymbol{F}_k}^{\mathrm{T}} + \boldsymbol{Q}_k$.
Calculation of the Beamforming Matrix
3. Calculate $\boldsymbol{B}_k$ by Equation (23),
FOR i $= 1 : J$
4. $\sigma_j^2(n+1) = \begin{cases} 0 & \theta_j \in [\theta_L(n), \theta_R(n)] \\ K_j(n+1)\sigma_j^2(n) & others \end{cases}$,
5. $error(n) = \frac{D(\theta_j)}{p(\theta_j, n)} - 1, j = 1, 2, \cdots, J$,
6. $errorsum = \sum_{i=1}^{n} error(i)$,
7. $K_j(n+1) = \min\{K_{\max}, 1 + K_p \cdot error(n) + K_i \cdot errorsum + K_d \cdot (error(n) - error(n-1))\}$,
8. $\boldsymbol{R}_{i+n} = \sum_{j=1}^{J} \sigma_j^2 \boldsymbol{a}\left(\theta_j\right) \boldsymbol{a}^H\left(\theta_j\right) + \sigma^2 \boldsymbol{I}$,
END
9. $\boldsymbol{w}_{opt} = \frac{\boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}(\theta_s)}{\boldsymbol{a}^{\mathrm{H}}(\theta_s) \boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}(\theta_s)}$
10. Calculate $\boldsymbol{W}_k^N$ by Equation (24),
where $\boldsymbol{w}^n$ is calculated by Equation (14).
Calculation of Kalman Filter Gain
11. Calculate $\widetilde{\boldsymbol{H}}_k$ by $\boldsymbol{W}_k^N$ and $h(k)$,
where $h(k)$ is calculated by Equation (26),
12. $\boldsymbol{K}_k^{(i+1)} = \boldsymbol{P}_{k|k-1} (\widetilde{\boldsymbol{H}}_k^{(i)})^T (\widetilde{\boldsymbol{H}}_k^{(i)} \boldsymbol{P}_{k|k-1} (\widetilde{\boldsymbol{H}}_k^{(i)})^T + \hat{\boldsymbol{R}}_k)^{-1}$,
Target State Measurement Update
13. $\hat{\boldsymbol{X}}_{k|k}^{(i+1)} = \hat{\boldsymbol{X}}_{k|k-1} + \boldsymbol{K}_k^{(i+1)} (\widetilde{\boldsymbol{Z}}_k - \overline{\boldsymbol{Z}}_k)$,
14. $\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k \widetilde{\boldsymbol{H}}_k\right) \boldsymbol{P}_{k|k-1}$.
Output: $\hat{\boldsymbol{X}}_k$, $\hat{\boldsymbol{P}}_k$.

Therefore, the measurement Jacobian matrix over an observation interval (with M snapshots) is given by the following:

$$\widetilde{\boldsymbol{H}}_k = \begin{bmatrix} \boldsymbol{W}_k^N h(k) \\ \boldsymbol{W}_k^N h(k+\tau) \\ \cdots \\ \boldsymbol{W}_k^N h(k+(M-1)\tau) \end{bmatrix} \tag{30}$$

where $\boldsymbol{W}_k^N$ is the set of N beamforming weight matrices obtained, and k represents the k-th time step.

4.4. Target State Measurement Update

Next, we need to update the target state measurement.

$$\hat{\boldsymbol{X}}_{k|k} = \hat{\boldsymbol{X}}_{k|k-1} + \boldsymbol{K}_k (\widetilde{\boldsymbol{Z}}_k - \overline{\boldsymbol{Z}}_k) \tag{31}$$

$$\overline{\boldsymbol{Z}}_k = \begin{bmatrix} \boldsymbol{W}_k^N \left(real\left(\boldsymbol{A}\left(\theta_{k|k-1}\right) s(k) \right) \right) \\ \boldsymbol{W}_k^N \left(real\left(\boldsymbol{A}\left(\theta_{k|k-1}\right) s(k+\tau) \right) \right) \\ \cdots \\ \boldsymbol{W}_k^N \left(real\left(\boldsymbol{A}\left(\theta_{k|k-1}\right) s(k+(M-1)\tau) \right) \right) \end{bmatrix} \tag{32}$$

$$\widetilde{\boldsymbol{Z}}_k = \begin{bmatrix} \boldsymbol{W}_k^N \boldsymbol{z}(k) \\ \boldsymbol{W}_k^N \boldsymbol{z}(k+\tau) \\ \cdots \\ \boldsymbol{W}_k^N \boldsymbol{z}(k+(M-1)\tau) \end{bmatrix} \tag{33}$$

Finally, the state covariance matrix is updated using the following measurement:

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k \widetilde{\boldsymbol{H}}_k\right) \boldsymbol{P}_{k|k-1} \tag{34}$$

where $\boldsymbol{I}$ is the identity matrix.

From the above process, compared to traditional DOA estimation, PID-EKF considers the target's motion characteristics. Additionally, by spatially filtering the raw element-domain signals within a specific window, SNR is improved, and the error information in the input data is reduced. This enhancement increases the robustness of DOA tracking while also reducing computational complexity.

5. Simulation and Result Analysis

5.1. Performance Verification and Analysis of the PIDBF

5.1.1. Simulation

The bearing angle of a single underwater target relative to the sensor array is 90°. The simulation involves inputting the target element-domain signal, with the underwater sound speed set to 1500 m/s. The frequency f, amplitude α, and initial phase ψ_0 of the target radiated noise are set to 5600 Hz, 1, and 0°, respectively. A ULA with 18 elements is used, with an element spacing of half the wavelength. The number of interference sources is set to 180, uniformly distributed over 0–180°, with the initial intensity of the interference sources within the main lobe set to 2. The maximum allowable value for the iterative gain factor is 10, with proportional control parameter $K_p = 6$, integral control parameter $K_i = 0.8$, and derivative control parameter $K_d = 0.8$. The desired sidelobe level is −40 dB.

To further evaluate the sidelobe convergence speed performance, we introduce the mean square error (MSE), and root mean square error (RMSE) of the sidelobes, as represented in Equations (32) and (33). These metrics calculate the mean square error and root mean square error between the maximum sidelobe level in each iteration and the desired sidelobe height.

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (yi - f(x_i))^2$$

$$\boldsymbol{R}MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{n} (Y_i - f(x_i))^2}$$

5.1.2. Result Analysis

To validate the effectiveness of PIDBF, this method was compared with Olen series optimization methods, which include OBF [15], ZOBF, YOBF [14], and PIDBF. We compare and analyze these methods from the perspectives of sidelobe convergence speed performance and computational efficiency.

- Comparison of sidelobe convergence speed performance.

Figure 2 shows the beam response comparison of Olen series optimization methods at different iteration counts. Through the beam response plots, it can be intuitively observed that PIDBF most significantly reduces sidelobe levels, achieving sidelobe heights very close to the desired levels within just 10 iterations.

Figure 3 and Tables 1 and 2 display the mean squared error (MSE) and root mean squared error (RMSE) of the beam response for the Olen series optimization methods at different iteration counts. It can be concluded that PIDBF achieves the fastest convergence speed towards the desired sidelobe height, reaching lower MSE and RMSE values in fewer iterations. Additionally, the convergence process with PIDBF is relatively stable, without

susceptibility to oscillations, and as the number of iterations increases, the final convergence result of the PIDBF most closely matches the desired sidelobe height.

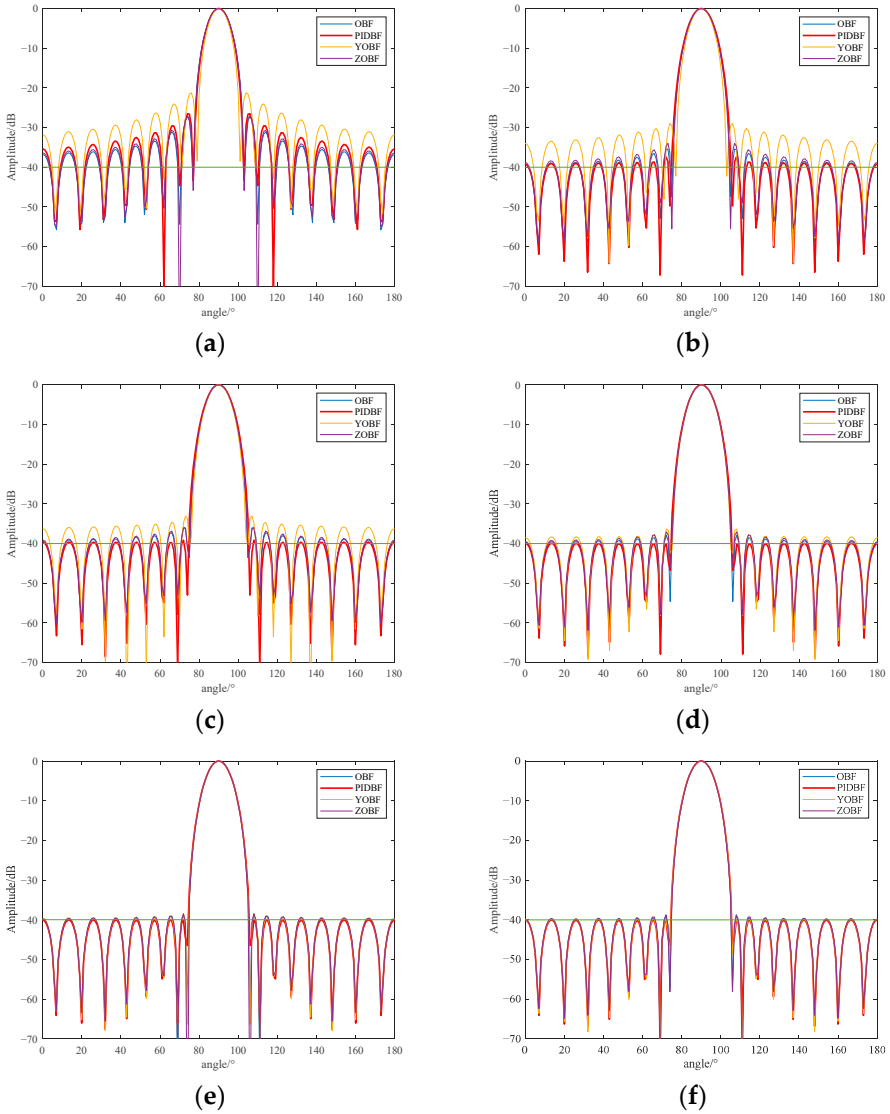


Figure 2. Comparison of beam patterns for Olen series optimization methods at different iteration counts. (**a**) 1 iteration. (**b**) 5 iterations. (**c**) 10 iterations. (**d**) 20 iterations. (**e**) 50 iterations. (**f**) 100 iterations.

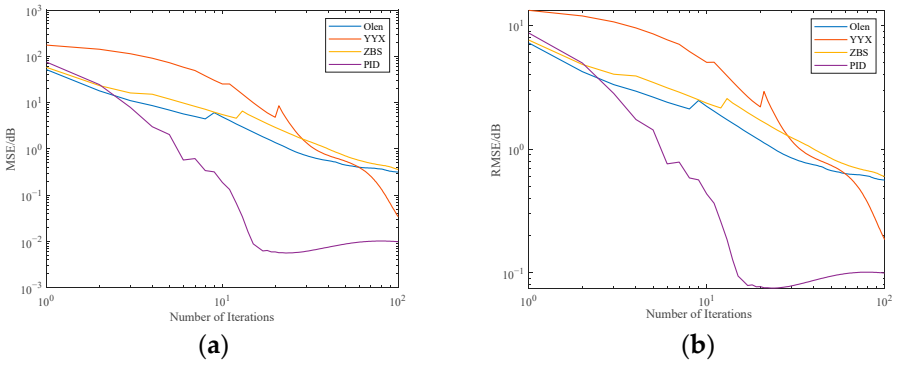


Figure 3. MSE and RMSE plots for the Olen series optimization methods. (**a**) MSE plot for the Olen series optimization methods. (**b**) RMSE plot for the Olen series optimization methods.

Table 1. The impact of different iteration counts on the MSE of Olen series optimization methods.

Optimization Method	Iteration (Iter)					
	1	**5**	**10**	**20**	**50**	**100**
OBF	51.8	6.91	4.91	1.38	0.45	0.3158
YOBF	175.08	72.79	25.23	4.28	0.54	0.04
ZOBF	57.67	11.99	5.54	2.93	0.71	0.36
PIDBF	75.15	2.04	0.19	0.006	0.008	0.009

Table 2. The impact of different iteration counts on the RMSE of Olen series optimization methods.

Optimization Method	Iteration (Iter)					
	1	**5**	**10**	**20**	**50**	**100**
OBF	7.19	42.62	2.21	1.17	0.67	0.56
YOBF	13.23	8.53	5.02	2.19	0.73	0.18
ZOBF	7.59	3.46	2.35	1.71	0.84	0.60
PIDBF	8.67	1.43	0.43	0.07	0.09	0.09

- Comparison of sidelobe convergence speed performance: comparison of computational efficiency.

The computational efficiency of Olen series optimization methods is compared by evaluating the time required by these methods to achieve the same RMSE. The tables below are formatted accordingly. Table 3 indicates the number of iterations needed for the Olen series methods to achieve RMSEs of less than 2. Table 4 indicates the computational time required by the Olen series methods to achieve RMSEs of less than 2. Table 5 indicates the number of iterations needed for the Olen series methods to achieve RMSEs of less than 1. Table 6 indicates the computational time required by the Olen series methods to achieve RMSEs of less than 1. Table 7 indicates the number of iterations needed for the Olen series methods to achieve RMSEs of less than 0.5. Table 8 indicates the computational time required by the Olen series methods to achieve RMSEs of less than 0.5. Figures 4–6 then depict the required computational time in bar chart form.

Table 3. The number of iterations required for the Olen series optimization method to achieve an RMSE of 2.

Optimization Method	Iteration (Iter)			
	4	**12**	**17**	**24**
OBF	2.93	1.86	1.35	1.00
YOBF	9.53	4.41	2.64	1.91
ZOBF	3.89	2.15	1.99	1.47
PIDBF	1.73	0.26	0.08	0.07

The numbers highlighted in red represent the RMSE values of different algorithms when they exactly meet the requirement of RMSE = 2.

Table 4. The time required for the Olen series optimization method to achieve an RMSE of 2.

Iterations (Times)	OBF	YOBF	ZOBF	PIDBF
Iteration time (cpu time)	0.1094	0.0938	0.1563	0.0313

Table 5. The number of iterations required for the Olen series optimization method to achieve an RMSE of 1.

Optimization Method	Iteration (Iter)			
	6	**24**	**34**	**41**
OBF	2.39	1.00	0.79	0.55
YOBF	7.64	1.91	0.98	0.70
ZOBF	3.13	1.47	1.13	0.98
PIDBF	0.76	0.07	0.08	0.09

The numbers highlighted in red represent the RMSE values of different algorithms when they exactly meet the requirement of RMSE = 1.

Table 6. The time required for the Olen series optimization method to achieve an RMSE of 1.

Iterations (Times)	OBF	YOBF	ZOBF	PIDBF
Iteration time (cpu time)	0.0625	0.1095	0.2031	0.0398

Table 7. The number of iterations required for the Olen series optimization method to achieve an RMSE of 0.5.

Optimization Method	Iteration (Iter)		
	10	**71**	**100**
OBF	2.21	0.62	0.50
YOBF	5.02	0.49	0.18
ZOBF	2.35	0.69	0.60
PIDBF	0.43	0.10	0.09

The numbers highlighted in red represent the RMSE values of different algorithms when they exactly meet the requirement of RMSE = 0.5.

Table 8. The time required for the Olen series optimization method to achieve an RMSE of 0.5.

Iterations (Times)	OBF	YOBF	ZOBF	PIDBF
Iteration time (cpu time)	0.1094	0.0938	0.1563	0.0313

Observations indicate that the PIDBF has a significant advantage in terms of operational efficiency. When achieving the same RMSE, the PIDBF requires fewer iterations and considerably less computational effort compared to the other methods. Moreover, as the requirement for a lower RMSE, which corresponds to higher sidelobe precision, increases, the advantages of the PIDBF become even more pronounced, relative to the other three methods.

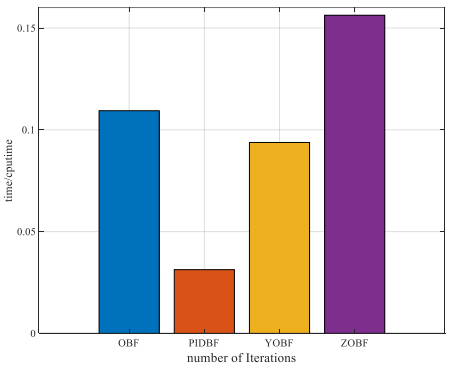


Figure 4. The time required for the Olen series optimization method to achieve an RMSE of 2.

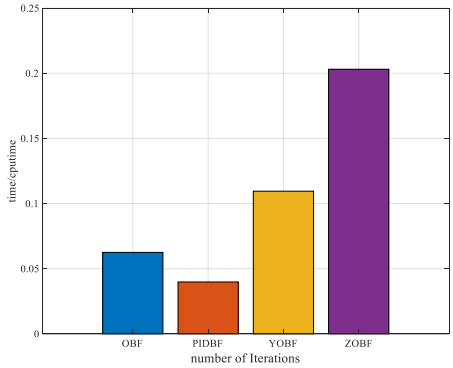


Figure 5. The time required for the Olen series optimization method to achieve an RMSE of 1.

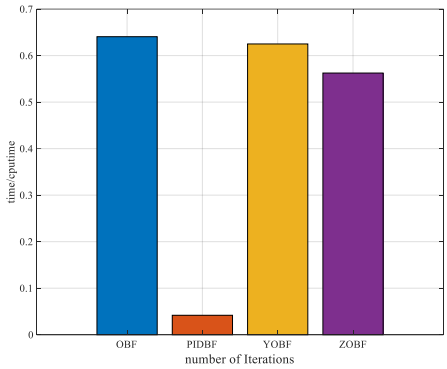


Figure 6. The time required for the Olen series optimization method to achieve an RMSE of 0.5.

5.2. Performance Verification and Analysis of the PIDBF-EKF

5.2.1. Simulation

The total simulation time is 1400 s, with a sampling interval of 1 s and with the initial state set to $\boldsymbol{X}_0 = \begin{bmatrix} -50^\circ & 0.005^\circ/\mathrm{s} \end{bmatrix}^{\mathrm{T}}$. The process noise is set to $\left(1 \times 10^{-3}\right)^\circ/\mathrm{s}$, and the frequency f, amplitude α, and initial phase ϕ_0 of the target radiation noise are, respectively, set to 170 Hz, 1, and 0°. There is also clutter with a fixed arrival direction of θ_c, a fixed frequency of f_c and initial phase ϕ_c, which are, respectively, set to 30°, 300 Hz, and 0°. A ULA of 18 elements with half-wavelength spacing is used.

Observation noise can be calculated through the SNR as follows:

$$\sigma_n = \frac{\sigma_s}{\mathrm{sqrt}(10^{\frac{\mathrm{SNR}}{10}})} / \sqrt{2}$$

where σ_n is the noise power, and σ_s is the signal power.

To numerically assess algorithm performance, the bearing estimation error (BEE) and the average bearing estimation error (ABEE) are defined as follows:

$$B(k) = \left| \hat{\theta}_k - \theta_k^{\mathrm{real}} \right|$$

$$A = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left(\hat{\theta}_k - \theta_k^{\mathrm{real}} \right)^2}$$

where $\hat{\theta}_k$ represents the target angle values obtained by different DOA estimation algorithms, θ_k^{real} is the true target angle value, and K is the total observation time.

To validate the effectiveness of the beam-domain DOA tracking, beam-domain DOA tracking is compared and analyzed with the traditional DOA estimation method CBF, and the element-domain EKF-based DOA tracking method (E-EKF) [9].

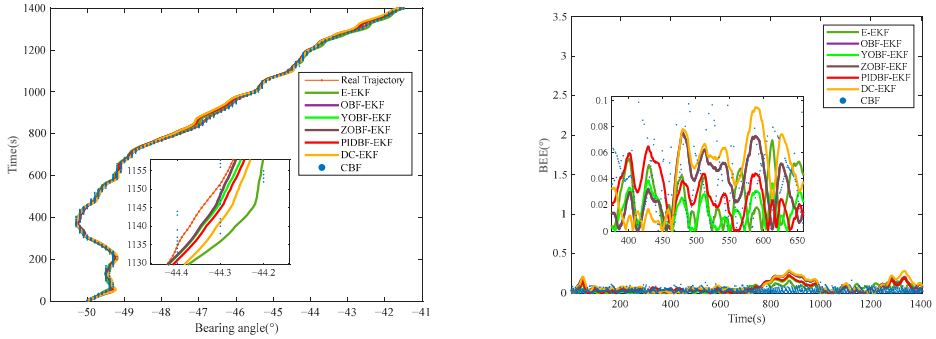
5.2.2. Result Analysis

The performance verification and analysis consist of two components. The first component is the impact of SNR on the DOA tracking results, comparing and analyzing the performance of DOA estimation and various DOA tracking methods under different SNR conditions. The second component is the impact of beam count on the DOA tracking results.

- Impact of SNR on the DOA tracking results

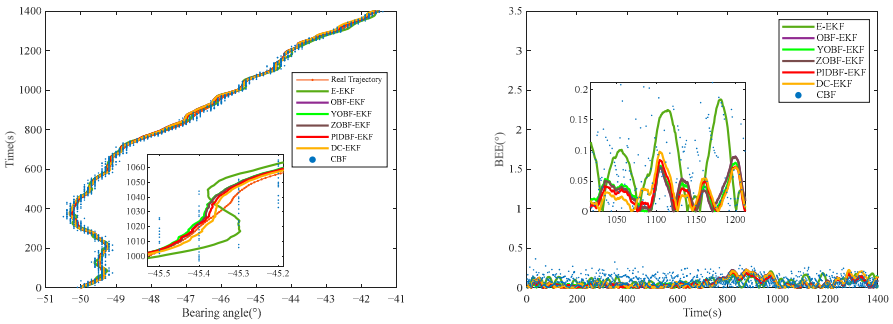
To investigate the impact of SNR on DOA tracking results, simulations are conducted under different SNR conditions. We obtained tracking results, BEE, and ABEE under different SNR conditions.

Figures 7–10 depict the tracking results and BEE comparisons of the algorithms at four different SNRs, namely 0 dB, −10 dB, −20 dB, and −30 dB, with the number of beams set to 5. Additionally, Table 9 shows the comparison of ABEE results for CBF, E-EKF, DC-EKF, Olen-EKF, YOBF-EKF, ZOBF-EKF, and PIDBF-EKF across these four different SNR levels. It can be observed that DOA estimation performs best at SNR levels of 0 dB and −10 dB. Under conditions of no observational errors, traditional DOA estimation methods demonstrate a clear advantage. However, when the SNR drops below −20 dB, the DOA estimation methods become nearly ineffective. This is because DOA estimation methods rely solely on measurement information and do not consider the motion characteristics of the target. Consequently, as the SNR decreases and measurement accuracy worsens, the DOA estimation results exhibit substantial deviations.



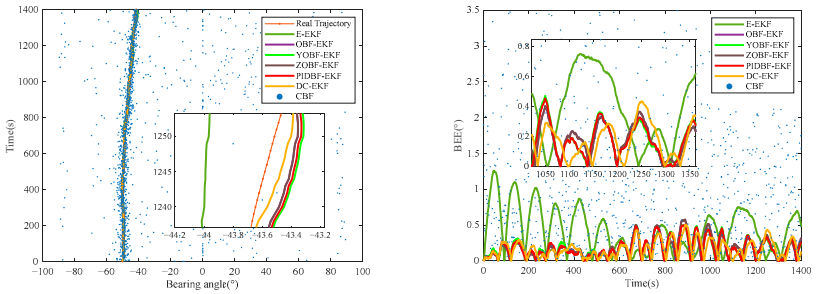
(**a**) (**b**)

Figure 7. Comparison of various methods with an SNR of 0 dB. (**a**) Comparison of bearing angle tracking result with an SNR of 0 dB. (**b**) BEEs obtained with an SNR of 0 dB.



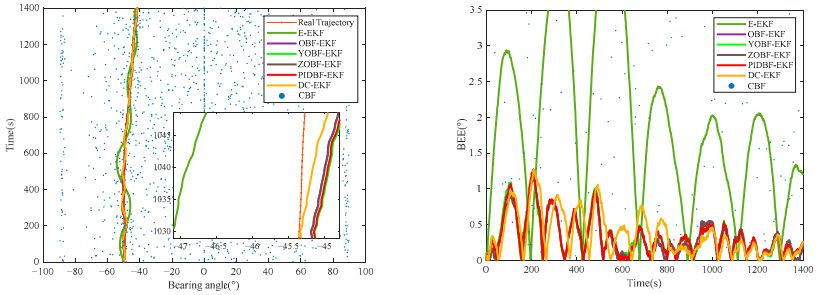
(**a**) (**b**)

Figure 8. Comparison of various methods with an SNR of −10 dB. (**a**) Comparison of bearing angle tracking result with an SNR of −10 dB. (**b**) BEEs obtained with an SNR of −10 dB.



(**a**) (**b**)

Figure 9. Comparison of various methods with an SNR of −20 dB. (**a**) Comparison of bearing angle tracking result with an SNR of −20 dB. (**b**) BEEs obtained with an SNR of −20 dB.



(**a**) (**b**)

Figure 10. Comparison of various methods with an SNR of −30 dB. (**a**) Comparison of bearing angle tracking result with an SNR of −30 dB. (**b**) BEEs obtained with an SNR of −30 dB.

Table 9. ABEE of various methods under different SNRs.

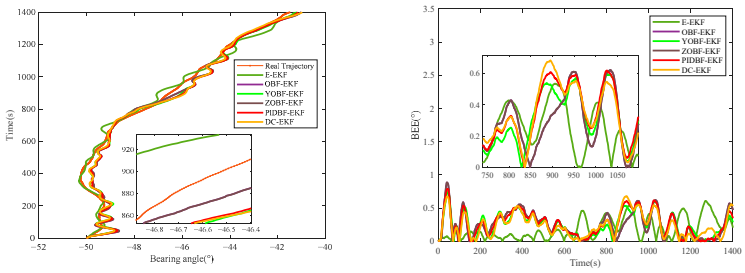
SNR/dB	ABEE/(°)						
	CBF	E-EKF	DC-EKF	OBF-EKF	YOBF-EKF	ZOBF-EKF	PIDBF-EKF
0	0.0435	0.1276	0.2030	0.1524	0.1302	0.1521	0.1417
−10	0.2941	0.5077	0.3233	0.2874	0.2635	0.2872	0.2746
−20	34.3920	0.6467	0.2945	0.3011	0.2956	0.3010	0.2956
−30	59.2715	2.1279	0.6979	0.6310	0.6436	0.6311	0.6493

From Table 9, beam-domain DOA tracking methods have a considerable advantage over element-domain tracking methods. Beam-domain tracking methods demonstrate superior DOA tracking accuracy, an,m as the SNR decreases, the tracking performance of the element-domain worsens significantly. In contrast, while beam-domain tracking results also decline, the reduction is relatively minor. The tracking accuracy differences among the various beamforming methods in beam-domain DOA tracking are minimal, as the differences between beamforming methods primarily manifest in the speed and stability of sidelobe reduction. The advantage of the PIDBF lies in its computational efficiency while maintaining the same tracking accuracy as other beam-domain DOA tracking methods. This makes PID-EKF more adaptable for array applications. When the array configuration changes (such as due to individual element failures or requiring recalculation of beamforming weights), PID-EKF exhibits a significant speed advantage.

- Impact of beam count on DOA tracking results

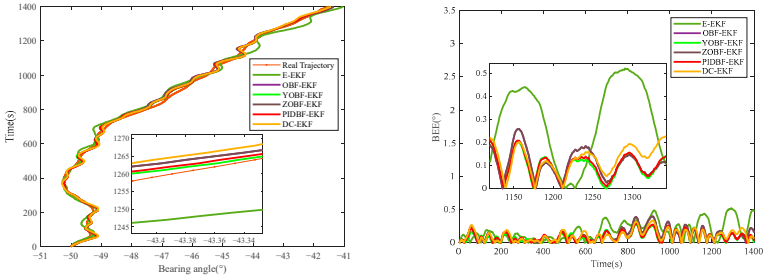
The impact of beam count on DOA tracking results is analyzed from two perspectives: computational efficiency and tracking accuracy. First, we analyze the computational efficiency. Figures 11–16 display the DOA tracking results under different beam count

conditions at SNR of −10 dB. According to Equation (12), the element-domain measurement equation results in a vector of size $PM \times 1$, while the beam-domain measurement equation provided by Equation (32) yields a vector of size $NM \times 1$. When it is time $P > N$, the beam-domain EKF can achieve measurement dimensionality reduction. This reduction in measurement dimensions significantly enhances the efficiency of the subsequent Kalman filtering computations, thereby improving computational speed. Figure 11a shows a comparison of tracking results when the observational dimension is reduced to one-sixth of that in E-EKF, and Figure 11b presents the corresponding BEE comparison. Similarly, Figures 12a, 13a, 14a, 15a and 16a, respectively, show the tracking result comparisons and BEE comparisons when the observational dimensions are reduced to 27.78%, 50%, 83.34%, and 100% of the element-domain observations.



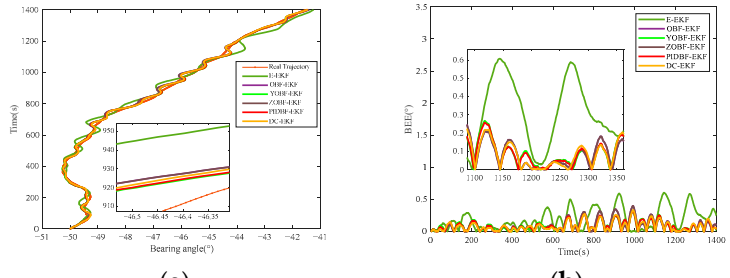
(**a**) (**b**)

Figure 11. Comparison of various methods with the number of beams set to 3. (**a**) Comparison of bearing angle tracking result with the number of beams set to 3. (**b**) BEEs obtained with the number of beams set to 3.



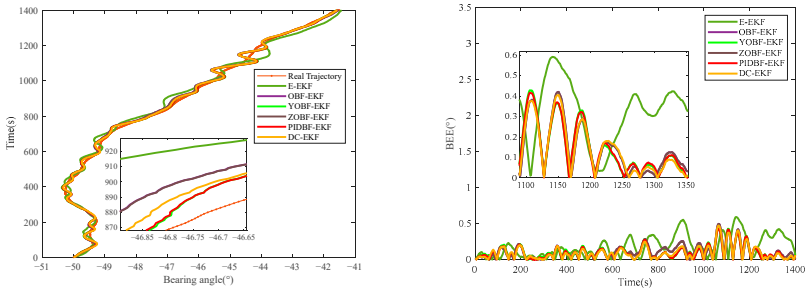
(**a**) (**b**)

Figure 12. Comparison of various methods with the number of beams set to 5. (**a**) Comparison of bearing angle tracking result with the number of beams set to 5. (**b**) BEEs obtained with the number of beams set to 5.



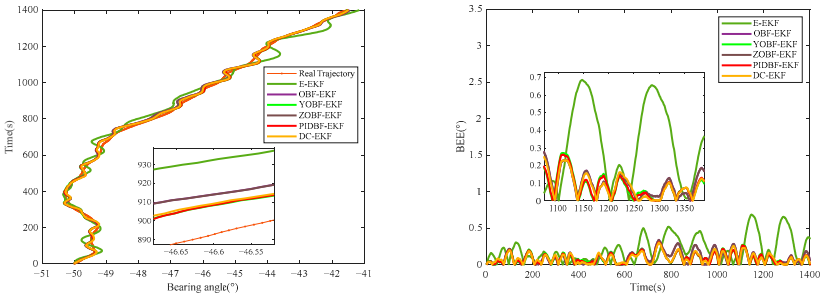
(**a**) (**b**)

Figure 13. Comparison of various methods with the number of beams set to 9. (**a**) Comparison of bearing angle tracking result with the number of beams set to 9. (**b**) BEEs obtained with the number of beams set to 9.



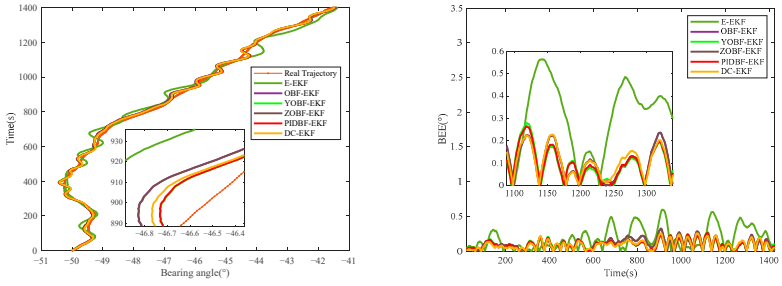
(**a**) (**b**)

Figure 14. Comparison of various methods with the number of beams set to 15. (**a**) Comparison of bearing angle tracking result with the number of beams set to 15. (**b**) BEEs obtained with the number of beams set to 15.



(**a**) (**b**)

Figure 15. Comparison of various methods with the number of beams set to 18. (**a**) Comparison of bearing angle tracking result with the number of beams set to 18. (**b**) BEEs obtained with the number of beams set to 18.



(**a**) (**b**)

Figure 16. Comparison of various methods with the number of beams set to 25. (**a**) Comparison of bearing angle tracking result with the number of beams set to 25. (**b**) BEEs obtained with the number of beams set to 25.

Next, we analyze the tracking accuracy. Table 10 displays the comparative results of ABEE for beam-domain DOA tracking under various beam counts. When the number of beams is less than 5, ABEE significantly decreases as the beam count increases. When the beam count is 9, the advantages of beam-domain tracking over element-domain tracking become quite evident, with ABEE reaching approximately 49.94–51.86% of the ABEE in element-domain DOA tracking. As the beam count continues to increase, the effectiveness of beam-domain tracking tends to stabilize within a certain range and no longer shows significant fluctuations. This is because the tracking error has a lower bound; when the number of beams reaches a certain threshold, sufficient beam-domain observations are obtained, and the resolution of beam formation is high, thus enabling the tracking error to approach very close to this lower limit without significant fluctuation.

Table 10. ABEE of various methods under different numbers of beams.

Number of Beams	ABEE/(°)					
	E-EKF	**DC-EKF**	**OBF-EKF**	**YOBF-EKF**	**ZOBF-EKF**	**PIDBF-EKF**
3	0.5053	0.3887	0.3736	0.3528	0.3736	0.3709
5	0.5077	0.3233	0.2874	0.2635	0.2872	0.2746
9	0.4984	0.2491	0.2585	0.2527	0.2583	02525
15	0.5109	0.2556	0.2626	0.2576	0.2624	0.2568
18	0.5218	0.2413	0.2504	0.2442	0.2502	0.2436
25	0.5169	0.2512	0.2591	0.2534	0.2589	0.2532

6. Conclusions

We explore robust DOA tracking technologies for underwater targets in complex environments. To enhance the target direction signal and effectively improve the SNR of the received signal, the PIDBF has been designed. This method accelerates the convergence of sidelobes to the desired height, achieving lower MSE and RMSE with fewer iterations. Furthermore, the method's convergence process is stable, with minimal oscillations, and as the iterations increase, the final convergence results closely match the desired sidelobe heights.

To address the two issues of traditional DOA estimation—namely, the lack of consideration for the kinematic information of the tracking target and the direct use of raw array information as measurements without considering the spatial sparsity of target information, beam-domain DOA tracking based on "beamforming + nonlinear filtering" is proposed. Furthermore, PIDBF-EKF is designed and proposed.

The simulation results demonstrate that under conditions of low SNR, where traditional DOA estimation methods and E-EKF are ineffective or fail, the proposed beam-domain DOA tracking method still maintains an average estimation error below 0.5°. PIDBF-EKF enhances computational efficiency while maintaining the same tracking accuracy as other beam-domain DOA tracking methods. This makes PID-EKF more adaptable for array applications. When the array changes (such as individual element failures or array configuration adjustments) and requires recalculating beamforming weights, PID-EKF shows a significant speed advantage. In the simulations, the extremely low error in the simulation algorithms is due to the ideal simulation conditions. The simulation algorithm exhibits an extremely low error due to the highly idealized conditions. The observation noise in the simulation is artificially defined as white noise that follows a Gaussian distribution. The motion model in the simulation is precise, and the beamforming accuracy is exceptionally high, achieving 0.01°. In real-world scenarios, the target is subject to unknown environmental interferences, leading to challenges, such as non-stationary and non-Gaussian observation noise, unknown target motion trajectories, and reduced beamforming accuracy. These factors collectively result in decreased estimation accuracy. Future research will aim to address these issues.

Author Contributions: Conceptualization, X.H. and W.H.; methodology, X.H. and W.H.; software, W.H.; validation, W.H. and Y.C.; Writing—review and editing, X.H. and W.H.; supervision, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 12104113, the Stable Supporting Fund of National Key Laboratory of Underwater Acoustic Technology under Grant No. JCKYS2023604SSJS010, and the Shanghai Aerospace Science and Technology Innovation Foundation under Grant No. SAST2022-011.

Data Availability Statement: No new data were created, and data are unavailable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, W.; Zhang, W.; Wu, Y. Joint Algorithm Based on Interference Suppression and Kalman Filter for Bearing-Only Weak Target Robust Tracking. *IEEE Access* **2019**, *7*, 131653–131662. [CrossRef]
2. Jin, X. The new trend of state estimation: From model-driven to hybrid-driven methods. *Sensors* **2021**, *21*, 2085. [CrossRef] [PubMed]
3. Afshari, H.H.; Gadsden, S.A.; Habibi, S. Gaussian Filters for Parameter and State Estimation: A General Review of Theory and Recent Trends. *Signal Process.* **2017**, *135*, 218–238. [CrossRef]
4. Van Veen, B.D.; Buckley, K.M. Beamforming: A Versatile Approach to Spatial Filtering. *IEEE ASSP Mag.* **1988**, *5*, 4–24. [CrossRef] [PubMed]
5. Zhang, Y.D.; Amin, M.G.; Himed, B. Sparsity-Based DOA Estimation Using Co-Prime Arrays. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013.
6. Yan, F.; Jin, M.; Qiao, X. Low-Complexity DOA Estimation Based on Compressed MUSIC and Its Performance Analysis. *IEEE Trans. Signal Process.* **2013**, *61*, 1915–1930. [CrossRef]
7. Cao, R. A Low-Complex One-Snapshot DOA Estimation Algorithm with Massive ULA. *IEEE Commun. Lett.* **2017**, *21*, 1071–1074. [CrossRef]
8. Kong, D.; Chun, J. A fast DOA tracking algorithm based on the extended Kalman filter. In Proceedings of the IEEE 2000 National Aerospace and Electronics Conference, NAECON 2000. Engineering Tomorrow (Cat. No. 00CH37093), Dayton, OH, USA, 12 October 2000.
9. Zhang, B.; Hou, X.; Yang, Y. Robust Underwater Direction-of-Arrival Tracking with Uncertain Environmental Disturbances Using a Uniform Circular Hydrophone Array. *J. Acoust. Soc. Am.* **2022**, *151*, 4101–4113. [CrossRef] [PubMed]
10. Hou, X.; Qiao, Y.; Zhang, B. Robust Underwater Direction-of-Arrival Tracking Based on Variational Bayesian Extended Kalman Filter. *JASA Express Lett.* **2023**, *3*, 014801. [CrossRef] [PubMed]
11. Hou, X.; Qiao, Y.; Zhang, B. Robust Underwater Direction-of-Arrival Tracking Based on AI-Aided Variational Bayesian Extended Kalman Filter. *Remote Sens.* **2023**, *15*, 420. [CrossRef]
12. Benesty, J.; Chen, J.; Huang, Y. Conventional Beamforming Techniques. In *Microphone Array Signal Processing*; Springer: New York, NY, USA, 2008; Volume 1, pp. 39–65.
13. Capon, J. High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE* **1969**, *57*, 1408–1418. [CrossRef]
14. Yang, Y. Beamforming and Beamspace High Resolution Bearing Estimation Techniques in Sonar Systems. Ph.D. Thesis, NPU, Xi'an, China, 2002.
15. Olen, C.A.; Compton, R.T. A Numerical Pattern Synthesis Algorithm for Arrays. In Proceedings of the ISAP, Merging Technologies for the 90's, Dallas, TX, USA, 7–11 May 1990.
16. Wu, R.; Bao, Z.; Ma, Y. Control of Peak Sidelobe Level in Adaptive Arrays. *IEEE Trans. Antennas Propag.* **1996**, *44*, 1341–1347.
17. Zhou, P.Y.; Ingram, M.A.; Anderson, P.D. Synthesis of minimax sidelobes for arbitrary arrays. *IEEE Trans. Antennas Propag.* **1998**, *46*, 1759–1760. [CrossRef]
18. Yang, Y.; Wan, C. Adaptive Beampattern Synthesis Based on Null Broadening. In Proceedings of the IEEE Antennas and Propagation Society Symposium, Monterey, CA, USA, 20–25 June 2004; Volume 4, pp. 3996–3999.

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.