Journal of
Marine Science and Engineering

MDPI

Article

Variational Bayesian Inference Time Delay Estimation for Passive Sonars

Feilong Ding [1,2,3], Cheng Chi [1,2], Yu Li [1,2,*] and Haining Huang [1,2]

[1] The Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
[2] Key Laboratory of Science and Technology on Advanced Underwater Acoustic Signal Processing, Chinese Academy of Sciences, Beijing 100190, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: ly@mail.ioa.ac.cn

Abstract: In passive sonars, distance and depth estimation of underwater targets is often limited by the accuracy of time delay estimations. The estimation accuracy of the existing methods of time delay estimation is limited by the uniform discrete grid (signal sampling rate). When a true time delay is out of the grid, the estimation accuracy deteriorates due to the mismatch between the real-time delay and the discrete grid. This paper proposes a new method for time delay estimation, which realizes the time delay estimation under the framework of variational Bayesian inference. The proposed method is grid-less, that is, continuous in the time domain. Unlike the popular grid-less compressive time delay estimation method, this method does not require parameter adjustment, and can automatically estimate the number of time delays, noise variance, and amplitude variance. The simulation results showed that the performance of the proposed method was superior to the reference state-of-the-art time delay estimation methods.

Keywords: time delay estimation; grid-less; variational Bayesian inference

1. Introduction

Time delay estimation is a basic problem in signal parameter estimation. It is required in many applications, such as speech analysis, channel estimation, radar navigation, and sonar signal processing [1–8]. In passive sonar systems, the distance and depth estimation of underwater targets is often affected by the performance of time delay estimation. In general, the accuracy of the time delay is of more concern when estimating the distance. A major time delay exists at this point without considering the multipath problem [9]. The multipath problem is very important when estimating the depth. Therefore, not only is the accuracy of the time delay estimation needed, but also the number of time delays [10].

The cross-correlation method is commonly used to estimate the time delay [11], which determines the time delay value by locating the peak of the cross-correlation function. The cross-correlation method is classical and easy to be implemented in practical systems, but its resolution is limited by the shortcomings of the wide main lobe and high sidelobes of the cross-correlation function [12]. Moreover, the estimation accuracy of this method is limited by the sampling frequency.

To mitigate the poor resolution problem of the cross-correlation method, many high-resolution methods of time delay estimation have been proposed, including look-ahead backtracking orthogonal matching pursuit (LABOMP) [4,5], multiple signal classification (MUSIC) [13], weighted l_1 norm minimization algorithm based on the cross-correlation method (CC-WL1) [14], and the deconvolution cross-correlation method (Dec-CC) [14]. These algorithms can obtain higher time delay resolution. However, there exists a common shortcoming of these methods, in that the performance of these methods is still limited by the grid. When the real-time delay is out of the grid, a grid mismatch arises, thereby reducing the performance of time delay estimation.

check for updates

Citation: Ding, F.; Chi, C.; Li, Y.; Huang, H. Variational Bayesian Inference Time Delay Estimation for Passive Sonars. *J. Mar. Sci. Eng.* **2023**, *11*, 194. https://doi.org/10.3390/jmse11010194

Academic Editors: Sergey Pereselkov, Matthias Ehrhardt and Pavel Petrov

Received: 14 November 2022
Revised: 23 December 2022
Accepted: 6 January 2023
Published: 12 January 2023

A simple way to solve the problem of grid mismatch is to increase the fineness of the grid, and solve the time delay through a denser grid. Although this can improve the accuracy of time delay estimation, dense grids often necessitate enormous data storage and computational pressure. Another method is to adaptively adjust the fineness of the grid to match the true time delay, which is known as off-grid time delay estimation. As described in Reference [15], sparse Bayesian inference can be used to iteratively update the fineness of the grid from the coarse grid to obtain a more accurate time delay estimation. However, this method still requires a coarse grid to be provided in advance.

Solving the time delay estimation problem in the continuous time domain, rather than the discrete one, may be the best solution to completely overcome the mismatch problem, which is referred to as grid-less time delay estimation. In reference [16], inspired by grid-free compressed beamforming [17], the time delay estimation problem was transformed into a sparse reconstruction problem, then the atomic-norm minimization (ANM) method was used to achieve the time delay estimation in the continuous time domain. A major limitation of the ANM method is that it requires sufficient separation between two adjacent time delays to be recovered. The reweighted atomic-norm minimization (RAM) method can avoid this limitation. According to reference [18], the RAM method has a higher resolution than the ANM method. Nevertheless, both the RAM and ANM methods meet a common issue, that is, when the noise variance is not known, it is necessary to manually adjust the parameters to obtain the best estimation result, a process which is often difficult and complex.

A VAriational Time Delay Estimation (VATDE) method is proposed in this paper. The VATDE method estimates the posterior probability density functions (PDFs) of time delays using variational Bayesian inference and calculates their expectations, rather than simply keeping the point estimates of time delays. The proposed method is grid-less, and, unlike the existing work in references [16,18], the VATDE does not need parameter adjustment, and can automatically estimate the number of time delays, noise variance, and amplitude variance. A simulation demonstrated that, even compared with other state-of-the-art methods, this method still had better time delay estimation performance.

2. Signal Model

Consider two spatially separated sensors in a passive system, which receive signals $x_1(t)$ and $x_2(t)$ from the same target source $s(t)$. Assume that there are no other interference sources, and the target source signal $s(t)$ is affected by Gaussian white noise during the propagation process. Then, the cross-correlation function $R_{x_1x_2}$ of the received signals $x_1(t)$ and $x_2(t)$ can be obtained as follows:

$$R_{x_1x_2}(\tau) = \sum_{k=1}^{K} \alpha_k R_{ss}(\tau - \tau_k) + R_{ee}(\tau) \tag{1}$$

where R_{ss} is the autocorrelation function of the source signal $s(t)$; K is the number of time delays; α_k is the amplitude of each time delay; τ_k represents the kth time delay; and R_{ee} is the noise-related term. Ideally, the received signal and noise are independent of each other, at which point $R_{ee} = 0$.

Assume that the source signal $s(t)$ is a broadband signal. Then, the Fourier transform is performed on the cross-correlation function $R_{x_1x_2}$, and the frequency form of Formula (1) can be obtained as follows:

$$G_{x_1x_2}(f_m) = G_{ss}(f_m) \sum_{k=1}^{K} \alpha_k e^{-j2\pi f_m \tau_k} + G_{ee}(f_m),\ m = 1, 2, \ldots M \tag{2}$$

where $G_{x_1x_2}(f_m)$ is the cross-power spectrum between the received signal $x_1(t)$ and $x_2(t)$; $G_{ss}(f_m)$ is the auto-power spectrum of the source signal $s(t)$; $G_{ee}(f_m)$ is the cross-power spectrum of the noise; $f_m = mf_s/N$ is a sampling of the frequency band of the source signal $s(t)$; f_s is the sampling rate; and N is the signal sampling length. The aim of this paper was to estimate exact time delays from the cross-power spectrum $G_{x_1x_2}(f_m)$ of the received signal.

Before using the VATDE algorithm proposed in this paper, according to the definition of Formula (3), the model in Formula (2) was further parameterized. Let

$$\omega_k = \frac{2\pi f_s \tau_k}{N} \tag{3}$$

where $\omega_k \in [-\pi, \pi]$. Note that ω_k and τ_k are in a one-to-one correspondence: when ω_k is evaluated correctly, τ_k can be obtained by $\omega_k N/2\pi f_s$. Therefore, Formula (2) can be further expressed as follows:

$$G_{x_1x_2}(m) = G_{ss}(m)\sum_{k=1}^{K} \alpha_k e^{-j\omega_k m} + G_{ee}(m),\ m = 1, 2, \ldots M \tag{4}$$

where $G_{ss}(m) = |G_{ss}(m)|$, because the phase component of the auto-power spectrum is 0.

In fact, the value of K is difficult to obtain, so this paper used dimension M of the source signal frequency band as the number of time delays in the overcomplete model. Of course, it is necessary to ensure $K << M$. At this point, the final modeling result of the VATDE algorithm can be given as follows:

$$\mathbf{G}_{x_1x_2} = \sum_{i=1}^{M} W_i \mathbf{a}(\omega_i) + \mathbf{G}_{ee} \tag{5}$$

where $W_i = G_{ss}(i)\alpha_i$ and $\mathbf{W} = [W_1, W_2, \ldots, W_M]^{\mathrm{T}}$ represent the weight of the time delay; and $\mathbf{a}(\omega_i) = e^{-j\omega_i m},\ m \in \{1, 2, \ldots, M\}$ represents the Fourier transform of the time delay value. Since the goal of this paper was to obtain K non-zero weights, sparse prior information was introduced for the weights. Ideally, the model shown in Formula (5) would finally obtain K real-time delays and the corresponding non-zero weights and $M - K$ zero weights. Specifically, this paper introduces binary latent variables $\mathbf{S} = [S_1, S_2, \ldots, S_M]^{\mathrm{T}}, S_i \in [0, 1]$ to control the weight $\mathbf{W}$ to promote sparsity. When $S_i = 1$, then W_i is active, that is, W_i is not equal to 0; otherwise, W_i is equal to 0. This can be expressed as the following formula:

$$p(W_i|S_i; \varepsilon) = (1 - S_i)\delta(W_i) + S_i f_{\mathrm{CN}}(W_i; 0; \varepsilon) \tag{6}$$

where $W_i|S_i$ follows the Bernoulli Gaussian distribution [19,20]; $p(S_i) = \rho^{S_i}(1-\rho)^{(1-S_i)}$; ρ is the probability that the ith weight component is active; and ε is the variance of the zero-mean gaussian distribution. In this paper, $f_{\mathrm{CN}}(\cdot;\ \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the complex multivariate Gaussian PDFs with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. For the ith time delay, let its prior distribution be $p(\omega_i)$. This is a von Mises distribution; it is extremely important in the distribution on the unit circle [21], and in most cases let $p(\omega_i) = 1/2\pi$, which represents a lack of prior knowledge [22,23].

According to the overcomplete model (Formula (5)), the likelihood function $p(\mathbf{G}_{x_1x_2}|\omega, \mathbf{W}; \nu)$ can be represented as follows:

$$p(\mathbf{G}_{x_1x_2}|\omega, \mathbf{W}; \nu) = f_{\mathrm{CN}}\left(\mathbf{G}_{x_1x_2}; \sum_i W_i \mathbf{a}(\omega_i), \nu\mathbf{I}\right) \tag{7}$$

where ν represents the variance of background noise. The maximum likelihood (ML) estimation of the core parameters in this paper can then be represented as follows:

$$(\hat{\rho}_{\mathrm{ML}}, \hat{\varepsilon}_{\mathrm{ML}}, \hat{\nu}_{\mathrm{ML}}) = \underset{\rho,\varepsilon,\nu}{\operatorname{argmax}} \int p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu)\ d\mathbf{S}d\mathbf{W}d\boldsymbol{\omega} \tag{8}$$

where $p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu)$ is the joint PDFs, which is the product of the likelihood function (Formula (7)) and the prior PDFs:

$$p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu) = p(\mathbf{G}_{x_1x_2} | \boldsymbol{\omega}, \mathbf{W}; \nu) \prod_{i=1}^{M} p(\omega_i) p(W_i | S_i) p(S_i) \tag{9}$$

However, the calculation of Formula (8) is extremely complex and intractable; thus, in the next section an iterative algorithm is proposed to solve it.

3. Variational Bayesian Inference Time Delay Estimation Algorithm

In this section, the proposed VATDE algorithm relies on the mean-field theory of variational Bayesian inference. This paper wished to determine the approximation PDFs $q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2})$ that minimize the Kullback–Leibler (KL) divergence $\mathrm{KL}(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2}) \| p(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2}))$, which means that the function $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2}))$ is a maximization [24] (pp. 732–733),

$$L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2})) = \mathrm{E}_{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2})}\left[\ln \frac{p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu)}{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2})}\right] \tag{10}$$

where $\mathrm{E}[\cdot]$ represents the expectation operator. Furthermore, all time delays and other variables are assumed to be independent of each other, and $q(\mathbf{S} | \mathbf{G}_{x_1x_2}) = \delta(\mathbf{S} - \hat{\mathbf{S}})$, so $q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2})$ can be represented as follows:

$$q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S} | \mathbf{G}_{x_1x_2}) = \prod_{i=1}^{M} q(\omega_i | \mathbf{G}_{x_1x_2}) q(\mathbf{W} | \mathbf{G}_{x_1x_2}, \mathbf{S}) \delta(\mathbf{S} - \hat{\mathbf{S}}) \tag{11}$$

According to Formula (11), the following estimation can be obtained:

$$\hat{\omega}_i = \arg\left(\mathrm{E}_{q(\omega_i | \mathbf{G}_{x_1x_2})}\left[e^{j\omega_i}\right]\right) \tag{12}$$

$$\hat{\mathbf{a}}_i = \mathrm{E}_{q(\omega_i | \mathbf{G}_{x_1x_2})}[\mathbf{a}(\omega_i)] \tag{13}$$

For the detailed derivation process, please refer to reference [21] (p. 26).

Due to $q(\mathbf{S} | \mathbf{G}_{x_1x_2}) = \delta\left(\mathbf{S} - \hat{\mathbf{S}}\right)$, the approximate posterior PDFs of the weights **W** can be represented as follows:

$$q(\mathbf{W} | \mathbf{G}_{x_1x_2}) = \int q(\mathbf{W} | \mathbf{G}_{x_1x_2}, \mathbf{S}) \delta\left(\mathbf{S} - \hat{\mathbf{S}}\right) d\mathbf{S} = q\left(\mathbf{W} \middle| \mathbf{G}_{x_1x_2}, \hat{\mathbf{S}}\right) \tag{14}$$

According to Formula (14), the mean and variance of the weights **W** can be estimated as follows:

$$\hat{\mathbf{W}} = \mathrm{E}_{q(\mathbf{W} | \mathbf{G}_{x_1x_2})}[\mathbf{W}],\ \hat{\mathbf{C}} = \mathrm{E}_{q(\mathbf{W} | \mathbf{G}_{x_1x_2})}\left[\mathbf{W}\mathbf{W}^{\mathrm{H}}\right] - \hat{\mathbf{W}}\hat{\mathbf{W}}^{\mathrm{H}} \tag{15}$$

Then, this paper defines S as the estimation of the number of time delays $\hat{K}$, which is a set of non-zero parameters in the variable $\mathbf{S}$:

$$S = \{i | 1 \le i \le M, S_i = 1\} \tag{16}$$

Therefore, $\hat{K}$ is equivalent to the cardinality of S, $\hat{K} = |\hat{S}|$. Finally, the estimation of the cross-power spectrum $\mathbf{G}_{x_1x_2}$ is given as follows:

$$\hat{\mathbf{G}}_{x_1x_2} = \sum_{i=1}^{\hat{S}} \hat{W}_i \hat{\mathbf{a}}(\omega_i) \tag{17}$$

Compared with Formula (5), Formula (17) represents the cross-power spectrum without noise.

Now, let us consider maximizing the function $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}))$. Naturally, it is difficult to maximize the function $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}))$ on all variables simultaneously. Therefore, this section adopts the alternative optimization scheme, that is, the function $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}))$ is maximized for each variable, while other variables are fixed. Let $\mathbf{z} = (\omega_1, \omega_2, \ldots, \omega_M, (\mathbf{W}, \mathbf{S}))$ be the set of all variables, and $\mathbf{z}_t, t = 1, 2, \ldots, M+1$ represents each variable in the set, then the above alternative optimization scheme can be expressed as follows [24] (p. 735), Equations (21) and (25).:

$$\ln q(\mathbf{z}_t|\mathbf{G}_{x_1x_2}) = \mathrm{E}_{q(\sim\mathbf{z}_t|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}, \mathbf{z})] + \mathrm{const} \tag{18}$$

where $\mathrm{E}_{q(\sim\mathbf{z}_t|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}, \mathbf{z})]$ represents the expectation of all the variables except variable $\mathbf{z}_t$. In the following sections, more optimization details are presented.

3.1. Inference Parameter $\boldsymbol{\omega}$

According to Formula (18), for each variable in the parameter $\boldsymbol{\omega}$, the optimization factor $\ln q(\omega_i|\mathbf{G}_{x_1x_2})$ can be represented as follows [25] (Chapter 10, p. 466):

$$\ln q(\omega_i|\mathbf{G}_{x_1x_2}) = \mathrm{E}_{q(\sim\omega_i|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}, \mathbf{z}; \rho, \varepsilon, \nu)] + \mathrm{const} \tag{19}$$

By substituting Formula (9), the following can be obtained:

$$\begin{aligned}\ln q(\omega_i|\mathbf{G}_{x_1x_2}) &= \mathrm{E}_{q(\sim\omega_i|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}|\boldsymbol{\omega}, \mathbf{W}; \nu)p(\boldsymbol{\omega})p(\mathbf{W}|\mathbf{S})p(\mathbf{S})] + \mathrm{const} \\ &= \mathrm{E}_{q(\sim\omega_i|\mathbf{G}_{x_1x_2})}\left[\sum_{d=1}^{M} \ln p(\omega_d) + \sum_{d=1}^{M} \ln p(S_d) + \ln p(\mathbf{W}|\mathbf{S})\right] \\ &\quad + \mathrm{E}_{q(\sim\omega_i|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}|\boldsymbol{\omega}, \mathbf{W}; \nu)] + \mathrm{const}\end{aligned} \tag{20}$$

When $i \in S$, Formula (20) is further expressed as follows:

$$\begin{aligned}\ln q(\omega_i|\mathbf{G}_{x_1x_2}) = \mathrm{E}_{q(\sim\omega_i|\mathbf{G}_{x_1x_2})}&\left[\left(\mathbf{G}_{x_1x_2} - \sum_{k\in S} \mathbf{a}(\omega_k)W_k\right)^{\mathrm{H}}\left(\mathbf{G}_{x_1x_2} - \sum_{k\in S} \mathbf{a}(\omega_k)W_k\right)/\nu\right] \\ &+ \ln p(\omega_i) + \mathrm{const}\end{aligned} \tag{21}$$

Then, the following can be obtained using Formula (15):

$$\ln q(\omega_i|\mathbf{G}_{x_1x_2}) = \ln p(\omega_i) + \Re\left\{\boldsymbol{\eta}_i^{\mathrm{H}}\mathbf{a}(\omega_i)\right\} + \mathrm{const} \tag{22}$$

where

$$\boldsymbol{\eta}_i = \frac{2}{\nu}\left(\mathbf{G}_{x_1x_2} - \sum_{k\in\hat{S}\backslash\{i\}} \hat{\mathbf{a}}_k\hat{W}_k\right)W_i^* - \frac{2}{\nu}\sum_{k\in\hat{S}\backslash\{i\}} \hat{C}_{k.i}\hat{\mathbf{a}}_k \tag{23}$$

When $i \notin \hat{S}$, then $\boldsymbol{\eta}_i = 0$, which means that only the prior information plays a role in Formula (22).

It is difficult to obtain the analytical results of Formula (22) directly. Fortunately, according to reference [22] (Heuristic 2), the $q(\omega_i|\mathbf{G}_{x_1x_2})$ can be approximated as a von Mises distribution:

$$q(\omega_i|\mathbf{G}_{x_1x_2}) \approx VM(\omega; \hat{\mu}_i, \hat{\kappa}_i) \tag{24}$$

The von Mises distribution $VM(\omega; \mu, \kappa)$ is expressed as follows:

$$VM(\omega; \mu, \kappa) = \frac{1}{2\pi \mathrm{I}_0(\kappa)}e^{\kappa\cos(\omega-\mu)} \tag{25}$$

where μ is the mean direction, κ is the concentration parameter, and $\mathrm{I}_p(\kappa)$ is the modified Bessel function of the first kind of order p [21] (p. 348). According to Formula (25), the following can be obtained [21] (p. 26):

$$\arg\left(\mathrm{E}_{VM(\omega;\mu,\kappa)}\left[e^{j\omega}\right]\right) = \arg\left(e^{j\mu}\frac{\mathrm{I}_1(\kappa)}{\mathrm{I}_0(\kappa)}\right) = \mu = \mathrm{E}_{VM(\omega;\mu,\kappa)}[\omega] \tag{26}$$

$$\mathrm{E}\left[e^{j\omega}\right] = e^{jm\mu}\frac{\mathrm{I}_m(\kappa)}{\mathrm{I}_0(\kappa)} \tag{27}$$

Then, parameters $\hat{\omega}_i$ and $\hat{\mathbf{a}}_i$ in Formulas (12) and (13) can be accordingly calculated.

3.2. Inference Parameters **W** and **S**

Then, fix $q(\omega_i|\mathbf{G}_{x_1x_2}), i = 1, 2, \ldots, M$, and give the maximization of the function of $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}))$ with respect to $q(\mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})$. We define matrix **J** and **H** as follows:

$$J_{ij} = \begin{cases} M, & i = j \\ \hat{\mathbf{a}}_i^{\mathrm{H}}\hat{\mathbf{a}}_j, & i \neq j \end{cases} \quad i, j \in \{1, 2, \cdots, M\} \tag{28}$$

$$\mathbf{H} = \left[\hat{\mathbf{a}}_1^{\mathrm{H}}\mathbf{G}_{x_1x_2}, \hat{\mathbf{a}}_2^{\mathrm{H}}\mathbf{G}_{x_1x_2}, \ldots, \hat{\mathbf{a}}_M^{\mathrm{H}}\mathbf{G}_{x_1x_2}\right]^{\mathrm{T}} \tag{29}$$

According to Formula (18), $\ln q(\mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})$ can be calculated with the following formula:

$$\begin{aligned} \ln q(\mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}) \quad &= \mathrm{E}_{q(\sim(\mathbf{W},\mathbf{S})|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}, \mathbf{z})] + \mathrm{const} \\ &= \mathrm{E}_{q(\sim(\mathbf{W},\mathbf{S})|\mathbf{G}_{x_1x_2})}\left[\sum_{d=1}^{M}\ln p(S_d) + \ln p(\mathbf{W}|\mathbf{S}) + \ln p(\mathbf{G}_{x_1x_2}|\boldsymbol{\omega}, \mathbf{W})\right] \\ &\quad +\mathrm{const} \\ &= \|\mathbf{S}\|_0 \ln\tfrac{\rho}{1-\rho} + \|\mathbf{S}\|_0 \ln\tfrac{1}{\pi\varepsilon} - \tfrac{1}{\varepsilon}\mathbf{W}_S\mathbf{W}_S^{\mathrm{H}} \\ &\quad +\left[\tfrac{2}{\nu}\Re\left\{\mathbf{W}_S^{\mathrm{H}}\mathbf{H}_S\right\} - \tfrac{1}{\nu}\mathbf{W}_S^{\mathrm{H}}\mathbf{J}_S\mathbf{W}_S\right] + \mathrm{const} \\ &= -(\mathbf{W}_S - \hat{\mathbf{W}}_S)^{\mathrm{H}}\hat{\mathbf{C}}_S^{-1}(\mathbf{W}_S - \hat{\mathbf{W}}_S) + \mathrm{const} \end{aligned} \tag{30}$$

where $\mathbf{W}_S$ represents the element of parameter **W** indexed by S; $\mathbf{J}_S$ and $\hat{\mathbf{C}}_S^{-1}$ represent the submatrix with S as the row and column index; and $\|\mathbf{S}\|_0$ represents the number of non-zero elements in **S**. Then, it has the following formula:

$$\hat{\mathbf{W}}_S = \frac{\hat{\mathbf{C}}_S\mathbf{H}_S}{\nu}, \hat{\mathbf{C}}_S = \nu\left(\mathbf{J}_S + \frac{\nu}{\varepsilon}\mathbf{I}\right)^{-1} \tag{31}$$

According to Formula (14), the approximate posterior PDFs $q(\mathbf{W}|\mathbf{G}_{x_1x_2})$ of the weight **W** can be represented as the product of $q(\mathbf{W}|\mathbf{G}_{x_1x_2}, \mathbf{S})$ and $\delta\left(\mathbf{S} - \hat{\mathbf{S}}\right)$. Therefore, to calculate $q(\mathbf{W}|\mathbf{G}_{x_1x_2})$, the value of $\hat{\mathbf{S}}$ must be estimated. By substituting the approximate PDFs $q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})$, put Formula (11) into Formula (10), and Formula (32) can be obtained as follows:

$$\begin{aligned} \ln Z\left(\hat{\mathbf{S}}\right) \quad &= L\left(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}); \hat{\mathbf{S}}\right) = \mathrm{E}_{q(\boldsymbol{\omega},\mathbf{W},\mathbf{S}|\mathbf{G}_{x_1x_2})}\left[\ln\frac{p\left(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \hat{\mathbf{S}}\right)}{q\left(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}\middle|\mathbf{G}_{x_1x_2}; \hat{\mathbf{S}}\right)}\right] \\ &= \mathrm{E}_{q(\boldsymbol{\omega},\mathbf{W},\mathbf{S}|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{S}) + \ln p(\mathbf{W}|\mathbf{S}) + \ln p(\mathbf{G}_{x_1x_2}|\boldsymbol{\omega}, \mathbf{W})] \\ &\quad -\mathrm{E}_{q(\boldsymbol{\omega},\mathbf{W},\mathbf{S}|\mathbf{G}_{x_1x_2})}[\ln q(\mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})] + \mathrm{const} \\ &= \|\hat{\mathbf{S}}\|_0 \ln\tfrac{\rho\nu}{(1-\rho)\varepsilon} + \ln\det\left(\mathbf{J}_S + \tfrac{\nu}{\varepsilon}\mathbf{I}\right)^{-1} + \tfrac{1}{\nu}\mathbf{H}_S^{\mathrm{H}}\left(\mathbf{J}_S + \tfrac{\nu}{\varepsilon}\mathbf{I}\right)^{-1}\mathbf{H}_S + \mathrm{const} \end{aligned} \tag{32}$$

Consequently, when we maximize $\ln Z(\hat{\mathbf{S}})$, it is possible to obtain $\hat{\mathbf{S}}$:

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmax}} \ln Z(\mathbf{S}) \tag{33}$$

The global optimal Bernoulli sequence $\mathbf{S}$ of the problem (Formula (33)) can be determined using an enumeration algorithm. However, for M-dimensional sequences, the computational cost of this operation is $O(2^M)$ and its computational complexity will increase exponentially as the M increases. For a given $\hat{\mathbf{S}}$, to reduce the complexity, a greedy iterative search strategy is proposed to determine the local optimal value. The specific update policy is as follows: For every $k = 1, 2, \ldots, M$, calculate $\Delta_k = \ln Z(\hat{\mathbf{S}}^k) - \ln Z(\hat{\mathbf{S}})$, where $\hat{\mathbf{S}}^k$ is the same as $\hat{\mathbf{S}}$, except that the kth element is flipped. Let $k_{\max} = \underset{k}{\operatorname{argmax}} \Delta_k$, if $\Delta_{k_{\max}} > 0$, update $\hat{\mathbf{S}}$ by flipping the $k_{\max}$th element; otherwise, if $\hat{\mathbf{S}}$ remains unchanged, then terminate the algorithm. As for the computation of Δ_k, a detailed method has been given in references [22,23], and will not be repeated here. If the algorithm does not stop halfway through, and $\hat{\mathbf{S}}^0$ is initialized to $\mathbf{0}_M$, it usually only takes $O(\hat{K})$ steps to determine the local maximum of $\ln Z(\hat{\mathbf{S}})$, in which $\hat{K}$ is the estimate of the number of time delays.

3.3. Inference Parameters $\hat{\rho}$, $\hat{\varepsilon}$, and $\hat{\nu}$

After updating parameters $\boldsymbol{\omega}$ and $\mathbf{W}$, parameters $\{\rho, \varepsilon, \nu\}$ are then estimated by maximizing the lower bound of function $L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}); \rho, \varepsilon, \nu)$. First, we bring the approximate PDFs $q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})$ from Formula (11) into Formula (10):

$$L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}); \rho, \varepsilon, \nu) = \mathrm{E}_{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})}\left[\ln \frac{p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu)}{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})}\right] \tag{34}$$

When only terms that depend on parameters $\{\rho, \varepsilon, \nu\}$ are considered, Formula (34) can be further expressed as follows:

$$\begin{aligned} &L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}); \rho, \varepsilon, \nu) \\ &= \mathrm{E}_{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}, \boldsymbol{\omega}, \mathbf{W}, \mathbf{S}; \rho, \varepsilon, \nu)] + \text{const} \\ &= \mathrm{E}_{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{S}) + \ln p(\mathbf{W}|\mathbf{S})] \\ &+ \mathrm{E}_{q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})}[\ln p(\mathbf{G}_{x_1x_2}|\boldsymbol{\omega}, \mathbf{W})] + \text{const} \end{aligned} \tag{35}$$

Based on the likelihood function (Formula (7)) defined in Section 2 and the form of the prior PDFs, Formula (36) can be obtained as follows:

$$\begin{aligned} &L(q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2}); \rho, \varepsilon, \nu) \\ &= \tfrac{1}{\nu}\left[2\Re\left\{\hat{\mathbf{W}}_{\hat{S}}^{\mathrm{H}} \mathbf{H}_{\hat{S}}\right\} - \hat{\mathbf{W}}_{\hat{S}}^{\mathrm{H}} \mathbf{J}_{\hat{S}} \hat{\mathbf{W}}_{\hat{S}} - \mathbf{G}_{x_1x_2}^{\mathrm{H}} \mathbf{G}_{x_1x_2} - \mathrm{tr}\left(\mathbf{J}_{\hat{S}} \hat{\mathbf{C}}_{\hat{S}}\right)\right] \\ &- M \ln \nu - \tfrac{1}{\varepsilon}\left[\hat{\mathbf{W}}_{\hat{S}}^{\mathrm{H}} \hat{\mathbf{W}}_{\hat{S}} + \mathrm{tr}\left(\hat{\mathbf{C}}_{\hat{S}}\right)\right] + \|\hat{\mathbf{S}}\|_0 \ln \rho \\ &- \|\hat{\mathbf{S}}\|_0 \ln \varepsilon + \left(M - \|\hat{\mathbf{S}}\|_0\right) \ln(1 - \rho) + \text{ const} \end{aligned} \tag{36}$$

If we set $\frac{\partial L}{\partial \rho} = 0$ and $\frac{\partial L}{\partial \varepsilon} = 0$, then Formula (37) can be obtained as follows:

$$\hat{\rho} = \frac{\|\hat{\mathbf{S}}\|_0}{M}, \hat{\varepsilon} = \frac{\hat{\mathbf{W}}_{\hat{S}}^{\mathrm{H}} \hat{\mathbf{W}}_{\hat{S}} + \mathrm{tr}\left(\hat{\mathbf{C}}_{\hat{S}}\right)}{\|\hat{\mathbf{S}}\|_0} \tag{37}$$

Clearly, $\hat{\rho}$ is given by the number of non-zero elements of $\hat{\mathbf{S}}$, and $\hat{\varepsilon}$ is the variance of the corresponding weights of these non-zero elements. Similarly, set $\frac{\partial L}{\partial \nu} = 0$, and Formula (38) can be obtained as follows:

$$\hat{\nu} = \frac{1}{M}\|\mathbf{G}_{x_1x_2} - \sum_{k\in\hat{S}} \hat{\mathbf{a}}_k \hat{W}_k\|_2^2 + \frac{1}{M}\mathrm{tr}\left(\mathbf{J}_{\hat{S}}\hat{\mathbf{C}}_{\hat{S}}\right) + \sum_{k\in\hat{S}} |\hat{W}_k|^2 \left(1 - \frac{\|\hat{\mathbf{a}}_k\|_2^2}{M}\right) \tag{38}$$

In summary, $\hat{\nu}$ is composed of three parts: the fitting of the residual, the error of the weight estimation, and the uncertainty of the parameter $\boldsymbol{\omega}$ estimation. The better $q(\boldsymbol{\omega}, \mathbf{W}, \mathbf{S}|\mathbf{G}_{x_1x_2})$ fits, the smaller the values of the above three parts will be.

3.4. The VATDE Algorithm

This section gives the details of the update process of the VATDE algorithm, as shown in Algorithm 1.

Algorithm 1. Description of the VATDE algorithm.

```
Input: The cross-power spectrum $\mathbf{G}_{x_1x_2}$ of the received signal.
Output: Estimate the number of delays $\hat{K}$, time delay estimation $\hat{\tau}$, weight estimate $\hat{\mathbf{W}}$, and
reconstruct the cross-power spectrum $\hat{\mathbf{G}}_{x_1x_2}$.
(1) initialize $\hat{\nu}$, $\hat{\rho}$, $\hat{\varepsilon}$ and $q(\omega_i|\mathbf{G}_{x_1x_2}), i = 1, 2, \ldots, M$, calculate $\hat{\mathbf{a}}_i$
(2) repeat
(3)     update $\hat{\mathbf{S}}$, $\hat{\mathbf{W}}_S$ and $\hat{\mathbf{C}}_S$ (see Section 3.2)
(4)     update $\hat{\nu}$, $\hat{\rho}$, and $\hat{\varepsilon}$ (as shown in Formulas (37) and (38))
(5)     When $i \in \hat{S}$, update $\boldsymbol{\eta}_i$, and $\hat{\mathbf{a}}_i$ (see Section 3.1)
(6) until the stopping criterion
(7) according to $\hat{\tau}_i = \hat{\omega}_i N / 2\pi f_s$, obtain $\hat{\tau}$
(8) return $\hat{K}$,$\hat{\tau}$,$\hat{\mathbf{W}}$ and $\hat{\mathbf{G}}_{x_1x_2}$
```

For Algorithm 1, each step will increase the value of function L, and eventually converge to some local maximum of function L; thus, initialization is very important for Algorithm 1. Next, a specific initialization scheme is described.

First, initialize $q(\omega_1|\mathbf{G}_{x_1x_2})$ as $q(\omega_1|\mathbf{G}_{x_1x_2}) \propto \exp\left(\|\mathbf{G}_{x_1x_2}^{\mathbf{H}}\mathbf{a}(\omega_1)\|_2^2/(\nu M)\right)$ by defining the following:

$$\begin{gathered} M' = \{m - n | m, n \in \{0, 1, \ldots, M-1\}, m > n\} \\ \mathbf{a}' : [-\pi, \pi) \to \mathbb{C}^{M'}, \omega \to \mathbf{a}'(\omega) \triangleq \left(\mathrm{e}^{\mathrm{j}\omega m} \mid m \in M'\right)^{\mathrm{T}}, \mathbf{a}(\omega) = [1; \mathbf{a}'(\omega)] \end{gathered} \tag{39}$$

Then, $q(\omega_1|\mathbf{G}_{x_1x_2})$ can be simplified to the form of Formula (22). By defining $\gamma = \frac{1}{M}\mathbf{G}_{x_1x_2}\mathbf{G}_{x_1x_2}^{\mathbf{H}}$, $q(\omega_1|\mathbf{G}_{x_1x_2})$ will be re-expressed as follows:

$$q(\omega_1|\mathbf{G}_{x_1x_2}) \propto \exp\left(\Re\left\{\frac{2}{\nu}\gamma^H \mathbf{a}'(\omega_1)\right\}\right) \tag{40}$$

Second, $\hat{\mathbf{a}}_1 = \mathrm{E}(\mathbf{a}(\omega_1))$ can be calculated. from Formula (28) $J_1 = M$, and $\mathbf{H}_1$ can be calculated using Formula (29). According to Formula (31), $\hat{\omega}_1$ can be calculated. Then, update $q(\omega_2|\mathbf{G}_{x_1x_2}) \propto \exp\left(\|\mathbf{G}^{\mathbf{H}}_{1x_1x_2}\mathbf{a}(\omega_2)\|_2^2/(\nu M)\right)$; at this time, $\mathbf{G}_{1x_1x_2} = \mathbf{G}_{x_1x_2} - \hat{\mathbf{a}}_1\hat{\omega}_1^{\mathrm{T}}$. According to the above description, $q(\omega_i|\mathbf{G}_{x_1x_2})$, $\hat{\mathbf{a}}_i$, and $\hat{\omega}_i$ are fully initialized.

Finally, we give the initialization of the parameters $\hat{\nu}$, $\hat{\rho}$ and, $\hat{\varepsilon}$. Let $\hat{\nu}$ be initialized to the average of the final quarter of the eigenvalues of matrix $\mathrm{E}\left(\mathbf{G}_{x_1x_2}\mathbf{G}^{\mathbf{H}}_{x_1x_2}\right)$; $\hat{\rho} = 0.5$; and $\hat{\varepsilon}$ is initialized to $\hat{\varepsilon} = \left(\mathrm{tr}\left[\mathbf{G}^{\mathbf{H}}_{x_1x_2}\mathbf{G}_{x_1x_2}\right]/M - \hat{\nu}\right)/(M\hat{\rho})$.

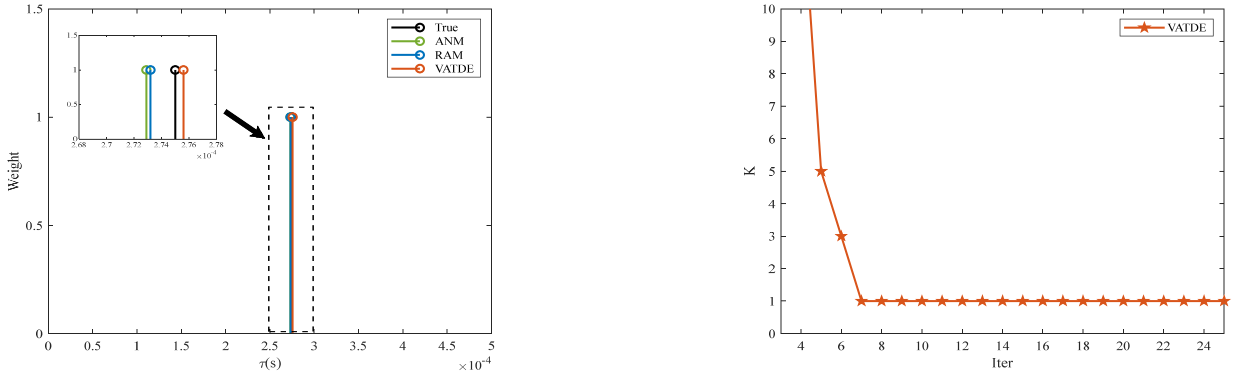
4. Simulation

Since the proposed algorithm is a grid-less time delay estimation algorithm, the simulation in this section is mainly compared with the proposed grid-less time delay estimation algorithm. It was assumed that the target source signal was a Gaussian white noise signal with a frequency of 1–4 kHz, and the signal length was 0.1 s. Considering a passive system, the sampling rate of the system was 10 kHz, the sampling interval was 1×10^{-4} s, two spatially separated synchronous sensors received the signal from the target source at the same time, and there was background noise in the space, which was Gaussian white noise with a mean of 0.

First, the ideal state was considered, that is, there was only one time delay between the received signals of the two sensors, the time delay value was 2.75×10^{-4} s, and the normalized amplitude was 1. The signal-to-noise ratio (SNR) was set to 10 dB, and the end condition of the VATDE algorithm was $\|\hat{\mathbf{G}}^{i-1}_{x_1x_2} - \hat{\mathbf{G}}^{i}_{x_1x_2}\|_2/\|\hat{\mathbf{G}}^{i-1}_{x_1x_2}\|_2 < 10^{-6}$, then $i = 3000$ represents the maximum number of iterations. Next, we compared the time delay estimation performance of the VATDE algorithm proposed in this paper with the ANM and RAM algorithms.

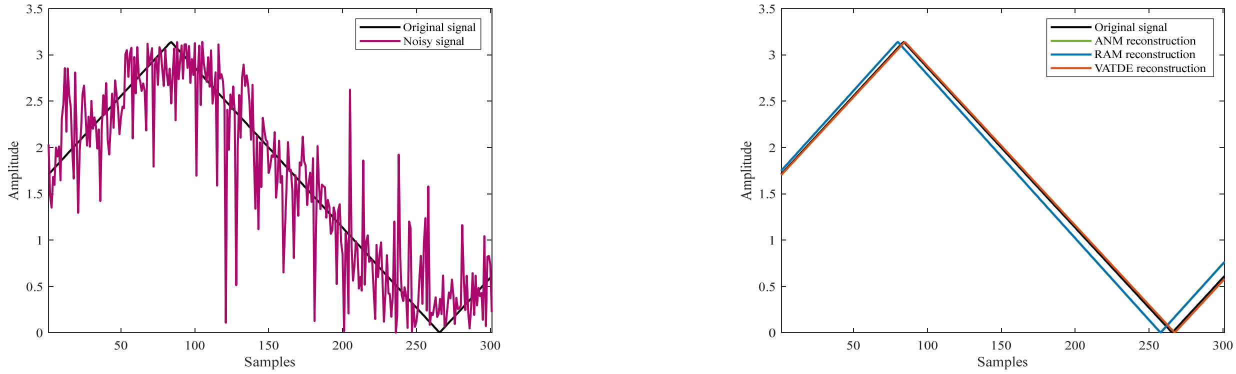
It can be seen from Figure 1a that all three algorithms, i.e., the ANM and RAM methods, and the VATDE proposed in this paper, could obtain better time delay estimation results in the case of a single time delay. Meanwhile, the enlarged window in Figure 1a further shows that the VATDE algorithm was closer to the real-time delay than the RAM and ANM algorithms. Therefore, the VATDE algorithm's time delay estimation accuracy was higher, while the time delay estimation accuracy of the RAM algorithm was also slightly better than that of the ANM algorithm. Figure 1b shows the estimation of the number of time delays existing using the VATDE algorithm. However, since the ANM and RAM algorithms cannot provide an estimate of the number of time delays, the number of time delays can only be manually determined. It can be seen from Figure 1b that the VATDE algorithm provided an accurate estimation of the number of time delays with only a few simple iterations, which avoids the interference caused by false peaks when determining the true time delay estimation.

Figure 2a shows a comparison of the phase of the cross-power spectrum between the received signals in the case of no noise and the phase of the cross-power spectrum when the SNR was 10 dB. Figure 2b shows the performance of different algorithms for reconstructing the original cross-power spectrum phase from the noisy cross-power spectrum phase. It can be clearly seen that the reconstructed cross-power spectrum phase obtained by the proposed VATDE algorithm fitted the original cross-power spectrum phase perfectly. In addition, the reconstruction performance of the ANM algorithm was similar to that of the RAM algorithm. Although the trend of the reconstructed cross-power spectrum phase was roughly the same as that of the original cross-power spectrum phase, there was still a certain deviation. Clearly, the time delay estimation performance of the algorithm is was to the reconstruction performance of the cross-power spectrum. When the time delay estimation of the algorithm was more accurate, then the reconstructed cross-power spectrum phase becomes more suitable for the original cross-power spectrum phase.



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

Figure 1. Single time delay estimation performance of different algorithms in the case of SNR 10 dB. (**a**) Time delay estimation result, (**b**) time delay number estimation.



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

Figure 2. Phase reconstruction performance of different algorithms in the case of SNR 10 dB. (**a**) Original signal phase and noisy, (**b**) reconstruction results of signal phase by different algorithms.

Second, assuming the presence of a multipath influence, there were two time delays between the received signals of the two sensors, which signifies that there will be three time delays in the cross-power spectrum. In this paper, the magnitudes of the three time delays were set to $[0.45 \times 10^{-4}\ \text{s}, 4.75 \times 10^{-4}\ \text{s}, 12.79 \times 10^{-4}\ \text{s}]$, and the amplitudes were $[1, 0.9, 0.7]$, respectively, while the SNR was also set as 10 dB. This paper compared the time delay estimation performance of the VATDE algorithm proposed in this paper with those of the ANM and RAM algorithms.

Figure 3a shows the time delay estimation results of the VATDE, ANM, and RAM algorithms in the presence of three time delay values. It can be seen that, compared with the ANM and RAM algorithms, the VATDE algorithm not only provided a more accurate time delay estimation result, it also yielded a more accurate amplitude estimate. The time delay estimation results of the RAM algorithm were similar to those of the ANM algorithm. However, under the same parameter selection, the RAM algorithm had fewer false peaks than the ANM algorithm, which reduced the interference of false peaks to the time delay estimation results. Figure 3b once again shows the estimated performance of the VATDE algorithm for the number of time delays present. It still only took a few simple iterations to estimate the real number of time delays, which is very meaningful for time delay estimation, as it avoids the problem of distinguishing the real delay from the obtained time delay estimation results.

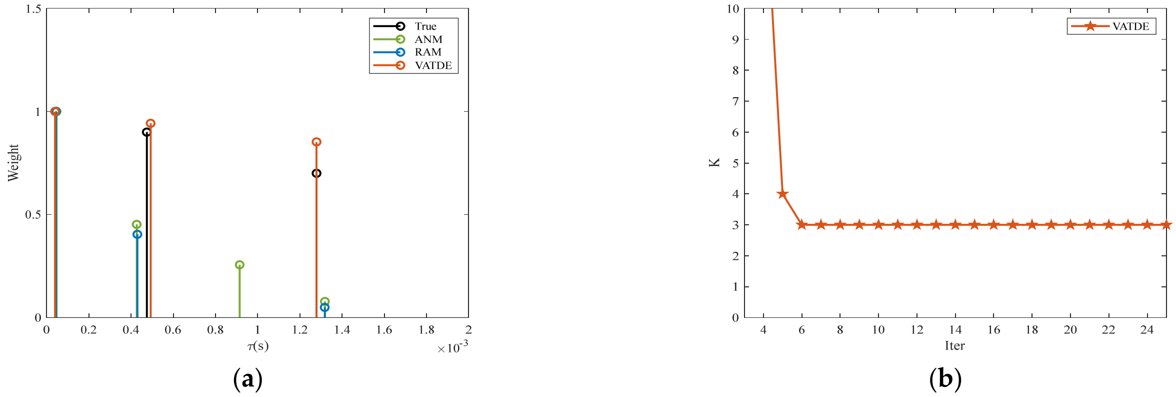


Figure 3. Multiple time delay estimation performance of different algorithms with an SNR of 10 dB. (**a**) Time delay estimation result, (**b**) time delay number estimation.

Figure 4a shows a comparison of the cross-power spectrum phase between the received signals in the case of no noise and the cross-power spectrum phase in the case of SNR 10 dB when there were three time delays. Compared with Figure 2a, the multiple time delays made the original cross-power spectrum phase more irregular; thus, it was more difficult to reconstruct the original cross-power spectrum phase from the noisy cross-power spectrum phase. Figure 4b shows the performance of different algorithms in reconstructing the original cross-power spectrum phase in the presence of three time delays. It is clear that the reconstructed cross-power spectrum phases of the ANM and RAM algorithms exhibited significant deviations from the original cross-power spectrum phase, due to their inaccurate time delay estimation and poor time delay amplitude estimation. Compared with the ANM algorithm, the reconstructed cross-power spectrum phase of the RAM algorithm could still reflect the original cross-power spectrum phase trend to a certain extent. This was because the reconstructed results of the ANM algorithm were greatly influenced by false peaks, and, thus, could not reflect the general trend of the original cross-power spectrum phase. Compared with the ANM and RAM algorithms, the reconstructed cross-power spectrum phase of the VATDE algorithm proposed in this paper still perfectly fitted the original cross-power spectrum phase, which once again reflects the superiority of the VATDE algorithm.

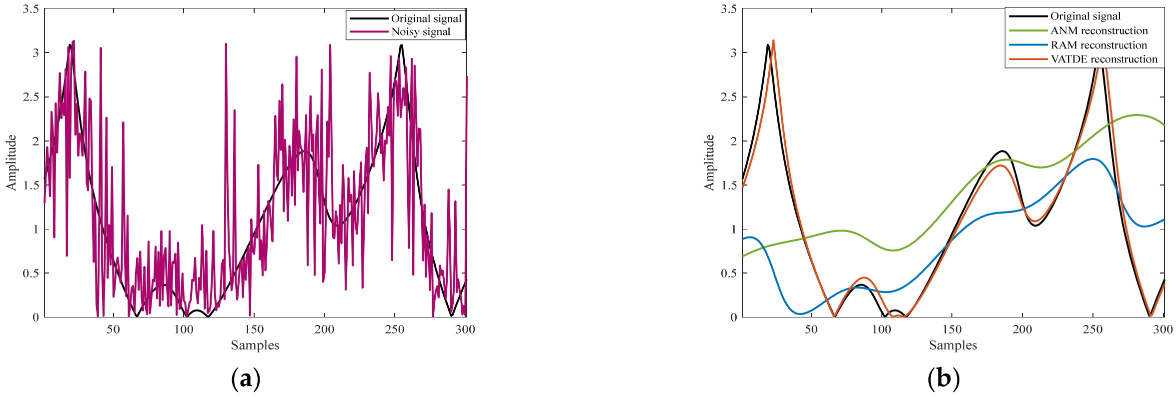


Figure 4. Signal reconstruction performance of different algorithms with an SNR of 10 dB. (**a**) Original signal phase and noise, (**b**) reconstruction results of signal phase by different algorithms.

Next, the computational efficiency of the different algorithms was compared. In this study, 100 Monte Carlo simulations were carried out under the multipath assumption. The ANM and RAM algorithms in this paper were solved using the SDPT3 [26] solver in the CVX toolbox [27]. Based on reference [28], the time complexity of the CVX toolbox for solving the semi-positive definite problem was approximately $O(N^3)$ Thus, the time complexity of

ANM was approximately $O(N^3)$. Since the estimation result of the RAM algorithm was obtained using multiple iterations of the ANM algorithm, the time complexity of RAM was approximately $O(LN^3)$, where L denotes the number of iterations. The time complexity of VATDE was approximately $O(N^2)$. Obviously, VATDE was the most computationally efficient in theory.

The average running times of the ANM, RAM, and VATDE algorithms in MATLAB 2020a are shown in Table 1. Compared with the ANM and RAM algorithms, VATDE clearly had the shortest running time. The running times of the ANM and RAM algorithms were much longer than that of the VATDE algorithm. The running time of the RAM algorithm was a multiple of the running time of the ANM algorithm, and the specific value depended on the number of iterations. In this paper, the number of iterations was set to three and it can be seen that RAM took three times as long as ANM to complete. The simulation results were consistent with the theory and prove the high computational efficiency of the VATDE algorithm.

Table 1. RUNNING TIME (INTEL^{R} CORE^{TM} i7-10510U CPU @ 1.80 GHz).

Algorithm	**ANM**	**RAM**	**VATDE**
Times (s)	148.1816	444.5448	7.8203

Of course, for the ANM, RAM, and VATDE algorithms, the running time was affected by the number of sampling points. When the number of sampling points is higher, the running time will be longer. However, as shown by the results in Figure 5, when the number of sampling point varied from 30 to 300, the running time and sampling points did not change linearly, and the multiplier of the increase in running time was greater than the multiplier of the increase in sampling points. Therefore, when the running time is long due to a large number of sampling points, the calculation efficiency can be improved by dividing the dataset into multiple small sets of sampling points. In addition, the results in Figure 5 show that the VALSE algorithm possessed computational advantages compared with the ANM and RAM algorithms at different numbers of sampling points, and the computational efficiency was more than 10 times that of the ANM algorithm.

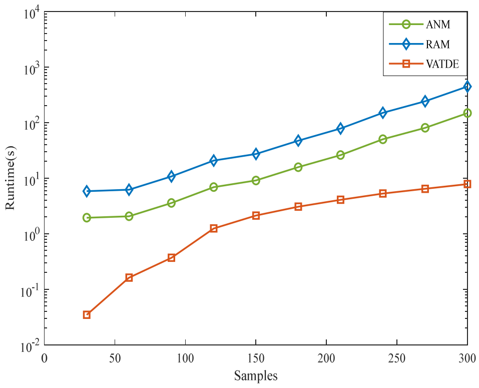
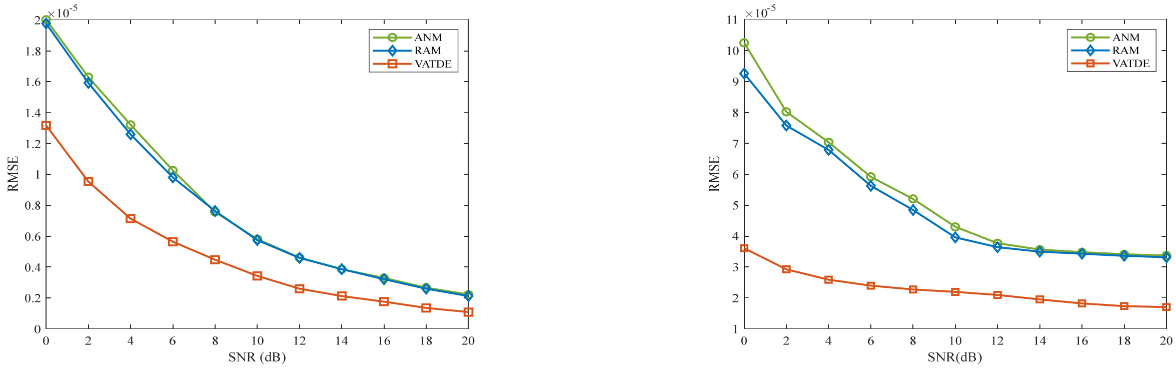


Figure 5. Running time under different numbers of sampling points.

Based on the above calculation efficiency comparison, it can be concluded that the VATDE algorithm proposed in this paper is more likely to be applied to real-time data processing.

Finally, the root-mean-squared error (RMSE) and SNR curves were plotted to show the time delay estimation accuracy of the three algorithms. Considering the different uses of time delay estimation, this paper divided the accuracy of delay estimation into two categories. One is the time delay estimation accuracy in the case of a single time delay without considering the multipath effect, which is usually applied to the geometric positioning problem in passive sonars [9]. The second category is the time delay estimation accuracy with multiple time delays in the multipath environment. The multipath effect will lead to more than three peaks in the cross-correlation function. The depth estimation problem of underwater targets can be solved by estimating multiple time delays [10].

Meanwhile, the signal simulation remained unchanged. The settings of single time delay and multiple time delays were the same as those in the above simulation. When SNR changed from 0 dB to 20 dB, then the RMSE of the ANM, RAM, and VATDE algorithms was compared, and each result was obtained using 500 Monte Carlo simulations. The simulation results are shown in Figure 6. Figure 6a shows the RMSE and SNR curves under a single time delay, while Figure 6b shows the RMSE and SNR curves under multiple time delays. It can be seen that the RMSE of the VATDE algorithm was lower than that of the ANM and RAM algorithms in the cases of both single and multiple time delays. Especially in a low SNR environment, the time delay estimation accuracy of the ANM and RAM algorithms deteriorated significantly, while VATDE still had high time delay estimation accuracy. In addition, the RMSE of the ANM and RAM algorithms was approximately equal with a single time delay, but with multiple time delays, the RMSE of the RAM algorithm was lower than that of the ANM algorithm. This may be because the RAM algorithm had fewer false peaks than the ANM algorithm. Moreover, it can be seen from the RMSE results in Figure 6a,b that the RMSE curves of the ANM, RAM, and VATDE algorithms in Figure 6b were much more than three times those in Figure 6a, which may have been due to the mutual interference between multiple time delays, which reduced the accuracy of time delay estimation. If the post-processing method of the cross-correlation function is adopted, that is, only the peak region of the cross-correlation function is searched separately, avoiding the search of the whole continuous time region, then the accuracy of time delay estimation with multiple time delays may be further improved. This is the next step of this study, and will not be discussed here.



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

Figure 6. RMSE of time delay estimation. (**a**) Single time delay, (**b**) multiple time delays.

5. Conclusions

In this paper, a time delay estimation algorithm based on variational Bayesian inference was proposed. All parameters in this algorithm were obtained by approximating its posterior probability distribution through variational Bayesian inference. Therefore, the advantages of the algorithm in this paper are that, firstly, it does not require any parameter adjustment and avoids the delay estimation error caused by improper parameter selection. Secondly, the algorithm presented in this paper is a grid-less time delay estimation method, which avoids the problem of mismatch between the grid and the real-time delay, and breaks the limitation of sampling rate, so that a time delay of arbitrary size can be estimated. Thirdly, the performance of the delay estimation was significantly improved by accounting for the uncertainty of the time delay estimation rather than calculating only the point estimate. Finally, the simulation results showed that, even compared with other current state-of-the-art grid-less time delay estimation methods, the proposed algorithm had higher time delay estimation accuracy in the case of a single time delay and multiple time delays.

In particular, in a low SNR environment, the time delay results could still be accurately estimated, which makes it suitable for passive sonar signal processing.

As an outlook, the next research direction following on from this paper is to find a better variational approximation to improve the performance of the algorithm, especially in situations where the time delays are very closely spaced. Moreover, this paper hopes to further improve the computational efficiency of the algorithm for use in large-scale scenarios.

Author Contributions: Conceptualization, Y.L.; methodology, C.C.; resources, H.H.; data curation, F.D.; writing—original draft preparation, F.D.; writing—review and editing, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62001469).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data in this study has been included in this paper.

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

References

1. Narayana Murthy, B.H.V.S.; Yegnanarayana, B.; Kadiri, S.R. Time Delay Estimation from Mixed Multispeaker Speech Signals Using Single Frequency Filtering. *Circuits Syst. Signal Process.* **2019**, *39*, 1988–2005. [CrossRef]
2. Jakubisin, D.J.; Buehrer, R.M. Performance, Complexity, and Receiver Design for Code-Aided Frame Synchronization in Multipath Channels. *IEEE Trans. Commun.* **2015**, *63*, 3363–3376. [CrossRef]
3. Quazi, A. An overview on the time delay estimate in active and passive systems for target localization. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 527–533. [CrossRef]
4. Junejo, N.U.R.; Esmaiel, H.; Zhou, M.; Sun, H.; Qi, J.; Wang, J. Sparse Channel Estimation of Underwater TDS-OFDM System Using Look-Ahead Backtracking Orthogonal Matching Pursuit. *IEEE Access* **2018**, *6*, 74389–74399. [CrossRef]
5. Junejo, N.U.R.; Esmaiel, H.; Sattar, M.; Sun, H.; Khalil, M.A.; Ullah, I. Sea Experimental for Compressive Sensing-Based Sparse Channel Estimation of Underwater Acoustic TDS-OFDM System. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 2523196. [CrossRef]
6. Zhang, T.; Liu, S.; He, X.; Huang, H.; Hao, K. Underwater Target Tracking Using Forward-Looking Sonar for Autonomous Underwater Vehicles. *Sensors* **2019**, *20*, 102. [CrossRef]
7. Wawrzyniak, N.; Stateczny, A. MSIS Image Postioning in Port Areas with the Aid of Comparative Navigation Methods. *Pol. Marit. Res.* **2017**, *24*, 32–41. [CrossRef]
8. Kazimierski, W.; Zaniewicz, G. Determination of Process Noise for Underwater Target Tracking with Forward Looking Sonar. *Remote Sens.* **2021**, *13*, 1014. [CrossRef]
9. Shao, Y.; Liu, F.; Zheng, G.; Jiang, F. A Method for Improving Three-Point Ranging Accuracy under Low SNR. In Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Dalian, China, 1–5 September 2019. [CrossRef]
10. Li, H.; Xu, Z.; Yang, K.; Duan, R. Use of multipath time-delay ratio for source depth estimation with a vertical line array in deep water. *J. Acoust. Soc. Am.* **2021**, *149*, 524–539. [CrossRef]
11. Knapp, C.; Carter, G. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process.* **1976**, *24*, 320–327. [CrossRef]
12. Park, H.-R.; Li, J. A Frequency-Domain SPICE Approach to High-Resolution Time Delay Estimation. *IEEE Wirel. Commun. Lett.* **2017**, *7*, 360–363. [CrossRef]
13. Ge, F.-X.; Shen, D.; Peng, Y.; Li, V.O.K. Super-Resolution Time Delay Estimation in Multipath Environments. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2007**, *54*, 1977–1986. [CrossRef]
14. Song, Q.; Ma, X. High-Resolution Time Delay Estimation Algorithms Through Cross-Correlation Post-Processing. *IEEE Signal Process. Lett.* **2021**, *28*, 479–483. [CrossRef]
15. Wei, S.; Li, W.; Su, Y.; Liu, R. Off-Grid Sparse Bayesian Inference with Biased Total Grids for Dense Time Delay Estimation. *J. Shanghai Jiaotong Univ.* **2022**, *27*, 1–9. [CrossRef]
16. Park, Y.; Seong, W.; Choo, Y. Compressive time delay estimation off the grid. *J. Acoust. Soc. Am.* **2017**, *141*, EL585–EL591. [CrossRef]
17. Xenaki, A.; Gerstoft, P. Grid-free compressive beamforming. *J. Acoust. Soc. Am.* **2015**, *137*, 1923–1935. [CrossRef]
18. Yang, Z.; Xie, L. Enhancing Sparsity and Resolution via Reweighted Atomic Norm Minimization. *IEEE Trans. Signal Process.* **2015**, *64*, 995–1006. [CrossRef]
19. Kormylo, J.; Mendel, J. Maximum likelihood detection and estimation of Bernoulli—Gaussian processes. *IEEE Trans. Inf. Theory* **1982**, *28*, 482–488. [CrossRef]

20. Soussen, C.; Idier, J.; Brie, D.; Duan, J. From Bernoulli–Gaussian Deconvolution to Sparse Signal Restoration. *IEEE Trans. Signal Process.* **2011**, *59*, 4572–4584. [CrossRef]
21. Mardia, K.V.; Jupp, P.E. *Directional Statistics*; Wiley: New York, NY, USA, 2000.
22. Badiu, M.-A.; Hansen, T.L.; Fleury, B.H. Variational Bayesian Inference of Line Spectra. *IEEE Trans. Signal Process.* **2017**, *65*, 2247–2261. [CrossRef]
23. Zhang, Q.; Zhu, J.; Gu, Y.; Xu, Z. Gridless Variational Direction-of-Arrival Estimation in Heteroscedastic Noise Environment. *IEEE J. Ocean. Eng.* **2021**, *46*, 1313–1329. [CrossRef]
24. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
25. Bishop, C.M.; Recognition, P. *Machine Learning (Information Science and Statistics)*; Springer: Secaucus, NJ, USA, 2006.
26. Toh, K.-C.; Michael, J.T.; Reha, H.T. SDPT3—A MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.* **1999**, *11*, 545–581. [CrossRef]
27. Grant, M.; Boyd, S. CVX: MATLAB Software for Disciplined Convex Programming, Version 2. March 2014. Available online: http://cvxr.com/cvx (accessed on 12 January 2022).
28. Boyd, S.; Stephen, P.; Lieven, V. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.

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.