

# Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) (Online) Journal homepage: <https://www.tandfonline.com/loi/tssc20>

## A new model for DOA estimation and its solution by multi-target intermittent particle swarm optimization

Lizhi Cui , Peichao Zhao , Xinwei Li , Bingfeng Li , Keping Wang , Xuhui Bu , Shumin Fei & Yi Yang

To cite this article: Lizhi Cui , Peichao Zhao , Xinwei Li , Bingfeng Li , Keping Wang , Xuhui Bu , Shumin Fei & Yi Yang (2020): A new model for DOA estimation and its solution by multi-target intermittent particle swarm optimization, Systems Science & Control Engineering, DOI: [10.1080/21642583.2020.1836525](https://doi.org/10.1080/21642583.2020.1836525)

To link to this article: <https://doi.org/10.1080/21642583.2020.1836525>



© 2020 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 23 Oct 2020.



Submit your article to this journal [↗](#)



Article views: 46



View related articles [↗](#)



View Crossmark data [↗](#)

# A new model for DOA estimation and its solution by multi-target intermittent particle swarm optimization

Lizhi Cui, Peichao Zhao, Xinwei Li, Bingfeng Li, Keping Wang, Xuhui Bu, Shumin Fei and Yi Yang

School of Electrical Engineering and Automation, Henan Polytechnic University, Henan, People's Republic of China

## ABSTRACT

Currently, the widely used methods for direction of arrival (DOA) estimation were constructed based on the subspace, such as Multiple Signal Classification (MUSIC) and Estimating Signal Parameter via Rotational Invariance Techniques (ESPRIT), which required that the number of sources is known beforehand. In this paper, a new method based on the Vector Error Model (VEM) for estimating the DOAs was proposed, which do not need the sources number in advance. The comparison of the performance between the VEM and the MUSIC model for DOA problem was given to demonstrate the effectiveness of our method. The algorithm of multi-target intermittent particle swarm optimization (MIPSO) was adopted to solve the VEM, and the performance of the VEM-MIPSO method was analysed through simulations for a uniform linear array and an L-shaped array respectively. The results showed that: (1) the VEM was an effective model to solve the DOA estimation without prior knowledge of the sources number; (2) the MIPSO was an efficient algorithm to solve the DOA estimation with high precision.

## ARTICLE HISTORY

Received 3 April 2020  
Accepted 10 October 2020

## KEYWORDS

Direction of arrival; uniform linear array; vector error model; multi-target intermittent particle swarm optimization

## 1. Introduction

In the past couple of decades, the problem of direction of arrival (DOA) estimation for narrowband sources has been studied extensively because of its wide application in many fields such as radar, navigation and wireless communication (Lee et al., 2018). The subspace methods of multiple signals classification (MUSIC, proposed in Schmidt and Schmidt (1986)) and Estimating signal Parameter via Rotational Invariance Techniques (ESPRIT, proposed in Yuen and Friedlander (1996)), are the most prominent approaches for DOA estimation with high resolution and simple structure. The theory of the MUSIC methods is based on the assumption of partitioning the observation space into two orthogonal subspaces of signal subspace and noise subspace. Actually, these methods cast the DOA estimation as an optimization problem objective function which involves the empirical estimate of the signal subspace projection matrix (Zhou, Huang, et al., 2018). However, the number of sources was assumed to be known or estimated by an information theoretic criterion, which could not be satisfied usually. Some source enumeration techniques, such as the Akaike information criterion (AIC), the minimum description length (MDL), have been proposed to solve the problem faced by the approaches based on MUSIC, but the performance

degrades heavily if the signal to noise ratios (SNR) is low or the number of samples (snapshots) is small (Djuric, 1996).

In another aspect, researches for quantifying performance loss of MUSIC-based DOA estimation has been addressed, which turn out to be a function of the number ( $M$ ) of sensors, number of snapshots ( $N$ ) and signal-to-noise ratio (SNR), respectively. However, the statistical performance is only characterized in the situation where  $N$  tends to infinity while  $M$  remains finite. When  $M$  is probably close to or even larger than  $N$ , strong discrepancies between the sample eigenvectors and their population counterparts will occur (Thiergart & Habets, 2013). Therefore, a method which could finish DOA estimation without knowing the number of sources is needed.

Some DOA estimation algorithms without source number have been proposed in Qi et al. (2004), Jin et al. (2008), and Zhang and Ng (2010). The method proposed in Qi et al. (2004) works with a threshold for pre-estimation of the number of sources, which was not addressed. An extension of the reversible jump Markov chain Monte Carlo method is mentioned in Jin et al. (2008), which is very inefficient and computationally intensive. A MUSIC-like method was introduced in Zhang and Ng (2010), which was constructed based on the framework of beamforming.

In this paper, a method based on the Vector Error Model (VEM) was adopted to solve the problem of DOA estimation, and an algorithm named multi-target intermittent particle swarm optimization (MIPSO) was used to solve the model based on the VEM. The rest of this paper is organized as follows: Section 2 introduces the VEM for the DOA estimation problem and the solving algorithm of MIPSO. Several simulations were carried out in Section 3 to demonstrate the performance of the VEM-MIPSO method for DOA estimation. Finally, conclusions were drawn in Section 4.

## 2. Solving method

In this section, the MUSIC model for the DOA estimation was mentioned firstly. Then, the new algorithm based on the VEM was proposed.

### 2.1. Music model for DOA estimation (Zhou, Hu, et al., 2018)

Consider the situation as shown in Figure 1 (a), where the  $i$ th source impinging from the far-field at an angle of  $\theta_i$  onto a Uniform Linear Array (ULA) of  $M$  identical omnidirectional equidistant sensors with an inter-sensor space of  $d$ . The received signal at the  $m$ th sensor is a superposition of time-shifted versions of all the sources corrupted with noise as following

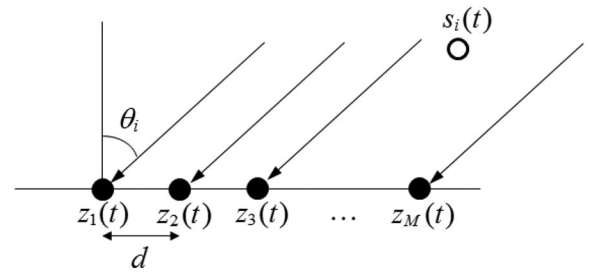
$$\begin{aligned} z_m(t) &= \sum_{i=1}^K s_i(t - (m-1) \frac{d}{c_0} \sin \theta_i) + n_m(t) \\ &= \sum_{i=1}^K s_i(t) e^{-j\omega(m-1) \frac{d}{c_0} \sin \theta_i} + n_m(t) \end{aligned} \quad (1)$$

where  $c_0$  is the speed of propagation,  $K$  is the number of the sources,  $t$  is the sampling point for the time. Rewriting (1) in vector format we get

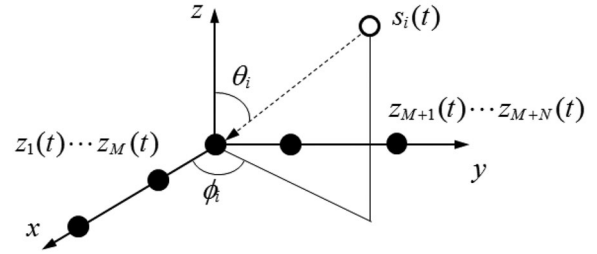
$$\mathbf{Z}(t) = \mathbf{A}(\theta) \mathbf{S}(t) + \mathbf{N}(t) \quad (2)$$

where  $\mathbf{Z}(t) = [\mathbf{z}_1(t), \mathbf{z}_2(t), \dots, \mathbf{z}_M(t)]^T$  with the superscript  $(\cdot)^T$  denoting the transpose is a complex matrix required by the sensors;  $\mathbf{S}(t) = [\mathbf{s}_1(t), \mathbf{s}_2(t), \dots, \mathbf{s}_K(t)]^T$  with the sources number of  $K$  contains all the narrow-band far-field source signals;  $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_K)]$  is the steering matrix with  $\mathbf{a}(\theta_i) = [1, a_{\theta_i}^1, \dots, a_{\theta_i}^{M-1}]^T$ ,  $a_{\theta_i} = e^{-j2\pi d/\lambda \sin(\theta_i)}$ ;  $\lambda$  is the wavelength of the source; and  $\mathbf{N}(t) = [\mathbf{n}_1(t), \mathbf{n}_2(t), \dots, \mathbf{n}_M(t)]^T$  is an unknown noise matrix.

The aim of DOA estimation is to calculate all the  $\hat{\theta}_i$ ,  $i = 1, 2, \dots, \hat{K}$  based on (2), where the top mark of  $\hat{\cdot}$  represents the estimated value. Generally,  $\{\theta_i\}_{i=1}^K$  is considered



(a) The situation for the ULA



(b) The situation for the L-shaped arrays

**Figure 1.** The  $i$ th source impinging on the sensor-array at angle  $\theta_i$ . (a) The situation for the ULA. (b) The situation for the L-shaped arrays.

as a fixed sampling grid that uniformly covers the DOA range  $[-\pi/2, \pi/2]$ .

An L-shaped array, as shown in Figure 1 (b), consists of two orthogonal  $(M + N)$ -element ULAs (denoted by x-ULA and y-ULA, respectively) in the  $xy$  plane, the narrow-band far-field uncorrelated signals impinge on the array from the distinct directions with elevation and azimuth angles  $\{(\theta_i, \phi_i)\}$ . Different from (2), the steering matrix for the L-shaped array has the form shown in (3).

$$\begin{aligned} \mathbf{A}(\{\theta, \phi\}) &= [\mathbf{A}_x(\{\theta, \phi\}); \mathbf{A}_y(\{\theta, \phi\})] \\ \mathbf{A}_x(\{\theta, \phi\}) &= \begin{bmatrix} 1 & 1 \\ e^{-j2\pi d \cos \phi_1 \sin \theta_1 / \lambda} & e^{-j2\pi d \cos \phi_2 \sin \theta_2 / \lambda} \\ \vdots & \vdots \\ e^{-j2\pi d(M-1) \cos \phi_1 \sin \theta_1 / \lambda} & e^{-j2\pi d(M-1) \cos \phi_2 \sin \theta_2 / \lambda} \\ \dots & \dots \\ \dots & 1 \\ \dots & e^{-j2\pi d \cos \phi_K \sin \theta_K / \lambda} \\ \vdots & \vdots \\ \dots & e^{-j2\pi d(M-1) \cos \phi_K \sin \theta_K / \lambda} \end{bmatrix} \\ \mathbf{A}_y(\{\theta, \phi\}) &= \begin{bmatrix} 1 & 1 \\ e^{-j2\pi d \sin \phi_1 \sin \theta_1 / \lambda} & e^{-j2\pi d \sin \phi_2 \sin \theta_2 / \lambda} \\ \vdots & \vdots \\ e^{-j2\pi d(N-1) \sin \phi_1 \sin \theta_1 / \lambda} & e^{-j2\pi d(N-1) \sin \phi_2 \sin \theta_2 / \lambda} \\ \dots & \dots \\ \dots & 1 \\ \dots & e^{-j2\pi d \sin \phi_K \sin \theta_K / \lambda} \\ \vdots & \vdots \\ \dots & e^{-j2\pi d(N-1) \sin \phi_K \sin \theta_K / \lambda} \end{bmatrix} \end{aligned} \quad (3)$$

## 2.2. DOA estimation model based on VEM

In order to avoid the calculation for the complex, we redefine (2) as follow

$$\begin{cases} \mathbf{D}(t) = \mathbf{B}\mathbf{C}(\theta) + \mathbf{V}(t) \\ \mathbf{D}(t) = [\mathbf{Z}_r(t)]^T = [\text{real}(\mathbf{Z}(t))]^T \\ \mathbf{B}(t) = [\mathbf{S}(t)]^T \\ \mathbf{C}(\theta) = [\mathbf{A}_r(\theta)]^T = [\text{real}(\mathbf{A}(\theta))]^T \\ \mathbf{V}(t) = [\mathbf{N}_r(t)]^T = [\text{real}(\mathbf{N}(t))]^T \end{cases} \quad (4)$$

where  $\mathbf{D}(t) = [\mathbf{d}_1(t), \mathbf{d}_2(t), \dots, \mathbf{d}_M(t)]$  is the real part of the complex signal matrix from  $M$  sensors' array without the superscript of  $(\cdot)^T$ ;  $\text{real}(\cdot)$  represents the real part of a complex value;  $\mathbf{B}(t) = [\mathbf{s}_1(t), \mathbf{s}_2(t), \dots, \mathbf{s}_K(t)]$  is the transpose of  $\mathbf{S}(t)$ ;  $\mathbf{C}(\theta) = [\mathbf{c}(\theta_1), \mathbf{c}(\theta_2), \dots, \mathbf{c}(\theta_K)]^T$  is the transpose of the real part of the steering matrix  $\mathbf{A}(\theta)$ ; and  $\mathbf{V}(t)$  is the transpose of the real part of the noise matrix  $\mathbf{N}(t)$ .

According to (4), the task of DOA estimation become a problem to judge whether a curve  $\mathbf{c}(\hat{\theta})$  with the parameter of  $\hat{\theta}$  exists in the matrix  $\mathbf{C}(\theta)$  as a row vector only based on the measured data  $\mathbf{D}(t)$ . In order to obtain this  $\mathbf{c}(\hat{\theta})$ , a matrix  $\mathbf{W}$  is introduced so that

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \vdots \\ \mathbf{y}_K^T \end{bmatrix} = \mathbf{W} \times \mathbf{D} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_K^T \end{bmatrix} \times \mathbf{D} \rightarrow \mathbf{C}(\theta) = \begin{bmatrix} \mathbf{c}^T(\theta_1) \\ \mathbf{c}^T(\theta_2) \\ \vdots \\ \mathbf{c}^T(\theta_K) \end{bmatrix} \quad (5)$$

where the note of  $\rightarrow$  represents that the  $\mathbf{Y}$  approximates the  $\mathbf{C}(\theta)$ . Therefore the objective function is given as

$$\min\{\varepsilon\} = \min\{\|\mathbf{y}_i^T - \mathbf{c}^T(\theta_i)\|_2^2\}, \forall i \quad (6)$$

where  $\|\cdot\|_2^2$  is the 2-norm of a vector;  $\forall i$  means for all parameters. Solving (6), a model named Vector Generated Model (VGM) described in (7) is obtained, which could generate a vector  $\mathbf{y}_i^T$  according to the known vector  $\mathbf{c}^T(\theta_i)$ . Equation (6) is called VEM, which calculated out the error between the known vector  $\mathbf{c}^T(\theta_i)$  and the generated vector  $\mathbf{y}_i^T$ .

$$\mathbf{y}_i^T = \mathbf{c}^T(\theta_i) \times \left( \frac{1}{M} \times \tilde{\mathbf{D}}^{*T} \times \tilde{\mathbf{D}}^* \right) \quad (7)$$

where  $\tilde{\mathbf{D}}^*$  is a matrix generated from the matrix  $\mathbf{D}$ . Please refer to references Cui, Poon, et al. (2014), Cui, Ling et al. (2014), Cui et al. (2015) or Appendix 1 to find the calculation process from (6) to (7). According to the definition of VEM in (6), if the known vector  $\mathbf{c}^T(\theta_i)$  exists in the matrix  $\mathbf{C}(\theta)$ , the generated vector  $\mathbf{y}_i^T$  will equal to the known vector  $\mathbf{c}^T(\theta_i)$ . The known vector of  $\mathbf{c}^T(\theta_i)$  is constructed as

follow

$$\begin{cases} \mathbf{c}^T(\theta_i) = \text{real}(\mathbf{a}(\theta_i)) \\ \mathbf{a}(\theta_i) = [1, a_{\theta_i}^1, \dots, a_{\theta_i}^{M-1}]^T \\ a_{\theta_i} = e^{-j2\pi d/\lambda \sin(\theta_i)} \end{cases} \quad (8)$$

where  $\text{real}(\cdot)$  represents the real part of a complex value. Similarly, we define the VEM shown in (9) for the L-shaped array.

$$\begin{cases} \mathbf{D}(t) = \mathbf{B}\mathbf{C}(\{\theta, \phi\}) + \mathbf{V}(t) \\ \mathbf{D}(t) = [\mathbf{Z}_r(t)]^T = [\text{real}(\mathbf{Z}(t))]^T \\ \mathbf{B}(t) = [\mathbf{S}(t)]^T \\ \mathbf{C}(\theta) = [\mathbf{A}_r(\{\theta, \phi\})]^T = [\text{real}(\mathbf{A}(\{\theta, \phi\}))]^T \\ \mathbf{V}(t) = [\mathbf{N}_r(t)]^T = [\text{real}(\mathbf{N}(t))]^T \end{cases} \quad (9)$$

Then, Equations (6), (7) and (8) are rewritten as (10), (11) and (12).

$$\min\{\|\mathbf{y}_i^T - \mathbf{c}^T(\{\theta_i, \phi_i\})\|_2^2\}, \forall i \quad (10)$$

$$\mathbf{y}_i^T = \mathbf{c}^T(\{\theta_i, \phi_i\}) \times \left( \frac{1}{t} \times \tilde{\mathbf{D}}^{*T} \times \tilde{\mathbf{D}}^* \right) \quad (11)$$

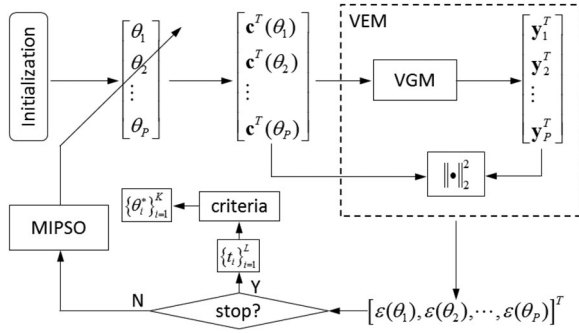
$$\begin{cases} \mathbf{c}^T(\{\theta_i, \phi_i\}) = \text{real}(\mathbf{a}(\{\theta_i, \phi_i\})) \\ \mathbf{a}(\{\theta_i, \phi_i\}) = [1, a_{\{\theta_i, \phi_i\}}^1, \dots, a_{\{\theta_i, \phi_i\}}^{M-1}, 1, b_{\{\theta_i, \phi_i\}}^1, \dots, b_{\{\theta_i, \phi_i\}}^{N-1}]^T \\ a_{\{\theta_i, \phi_i\}} = e^{-j2\pi d \cos \phi_i \sin \theta_i / \lambda} \\ b_{\{\theta_i, \phi_i\}} = e^{-j2\pi d \sin \phi_i \sin \theta_i / \lambda} \end{cases} \quad (12)$$

## 2.3. Solving algorithm

First, let us consider the problem described in (6), then (10) will be solved similarly. There are many algorithms to find the optimal solution for (6). However, there should be more than one solution for (6) needing to be obtained, and the number of the solution is unknown which corresponds to the number of the signal  $\mathbf{S}(t) = [\mathbf{s}_1(t), \mathbf{s}_2(t), \dots, \mathbf{s}_K(t)]^T$ . Here the multi-target intermittent particle swarm optimization (MIPSO) algorithm is adopted to calculate the optimal parameters  $\{\theta_i\}_{i=1}^K$  for all the signals. The MIPSO algorithm is designed as

$$\begin{cases} \Delta\theta_i^+ = \omega \times \Delta\theta_i + c_1 \times r_1 \times (b_i - \theta_i) \\ + c_2 \times r_2 \times (t_i - \theta_i) \\ \theta_i^+ = \theta_i + \Delta\theta_i^+ \end{cases} \quad (13)$$

where  $\theta_i$  is the current position (value) of the  $i$ th particle (parameter),  $\theta_i^+$  is the next position of the  $i$ th particle,  $\Delta\theta_i$  is the current velocity (increment) of the  $i$ th particle,  $\Delta\theta_i^+$  is the next velocity of the  $i$ th particle,  $b_i$  is the best position of the  $i$ th particle,  $t_i$  is a particle around the  $i$ th particle within a small scope  $\delta$ , and  $\varepsilon(t_i)$  is the smallest value in this scope  $\delta$ .  $\omega$  is the inertia coefficient,  $c_1$  and  $c_2$  are two accelerated factors which are always set as 2,  $r_1$  and  $r_2$  are two random constants whose value are between



**Figure 2.** Calculation process based on VEM and MIPSO.

0 and 1. Here,  $t_i$  replaces the global best particle in the traditional PSO algorithm. During the calculation, every particle will find its  $t_i$  and will approach this  $t_i$  dynamically. If the number of these  $\{t_i\}_{i=1}^L$  and their fitness  $\varepsilon(t_i)$  do not change during a certain step, or if the maximum step is reached, the calculation iteration will stop. In order to keep the ability of exploitation for all the particles to find their  $t_i$ , these  $t_i$  are searched intermittently (every 20 steps). After the iteration stops, all the particles will converge around several  $t_i$ . The final solution  $\{\theta_i^*\}_{i=1}^N$  are picked out from these  $\{t_i\}_{i=1}^L$ , where  $N \leq L$ , according to two criteria: (1) whether the  $\varepsilon(t_i)$  is smaller than a threshold and (2) whether the values of the  $t_i$  locate within the definition domain. Please refer to reference (Cui et al., 2015) for detailed information.

For objective function for the L-shaped array shown in (10), the MIPSO algorithm is designed as

$$\begin{cases} \Delta(\theta_i, \phi_i)^+ = \omega \times \Delta(\theta_i, \phi_i) + c_1 \times r_1 \times (b_i - (\theta_i, \phi_i)) \\ \quad + c_2 \times r_2 \times (t_i - (\theta_i, \phi_i)) \\ (\theta_i, \phi_i)^+ = (\theta_i, \phi_i) + \Delta(\theta_i, \phi_i)^+ \end{cases} \quad (14)$$

where the parameter of  $\theta_i$  is replaced by  $(\theta_i, \phi_i)$ .

#### 2.4. The VEM-MIPSO method

Figure 2 gives the process of the calculation based on VEM and MIPSO. In order to explain it simply, we use the parameter  $\theta_i$  for the ULA in Figure 1(a). Please replace the parameter of  $\theta_i$  by the parameter of  $(\theta_i, \phi_i)$  for the L-shaped array.

Firstly, the angle parameters  $\theta_i$  were initialized randomly with the number of  $P$ . Then, reference curves  $\mathbf{c}^T(\theta_i)$  were generated according to the  $\theta_i$ . Input the  $\mathbf{c}^T(\theta_i)$  to VGM,  $\mathbf{y}_i^T$  were calculated. The errors  $\varepsilon(\theta_i)$  were generated according to  $\mathbf{c}^T(\theta_i)$  and  $\mathbf{y}_i^T$ . Finally, the parameters  $\theta_i$  were adjusted based on the errors  $\varepsilon(\theta_i)$  until all the parameters  $\theta_i$  converged around several ones  $\{t_i\}_{i=1}^L$ . The criteria mentioned above is used here to pick the final solutions

$\{\theta_i^*\}_{i=1}^K$ , where  $K$  is the number of the sources, and much smaller than  $P$ .

For calculation convenience, the errors between  $\mathbf{y}_i^T$  and  $\mathbf{c}^T(\theta_i)$  are defined as

$$\begin{cases} \mathbf{e}(\theta) = [\varepsilon(\theta_1), \varepsilon(\theta_2), \dots, \varepsilon(\theta_P)]^T \\ \varepsilon(\theta_i) = \frac{\|\mathbf{y}_i^T - \mathbf{c}^T(\theta_i)\|_2^2}{\|\mathbf{c}^T(\theta_i)\|_2^2} \end{cases} \quad (15)$$

where the subscript of  $P$  is the number of the parameters calculated by the MIPSO algorithm simultaneously. According to the definition in formula (14), the  $\varepsilon(\theta_i)$  will decrease when the  $\mathbf{c}^T(\theta_i)$  and  $\mathbf{y}_i^T$  approach the same shape.

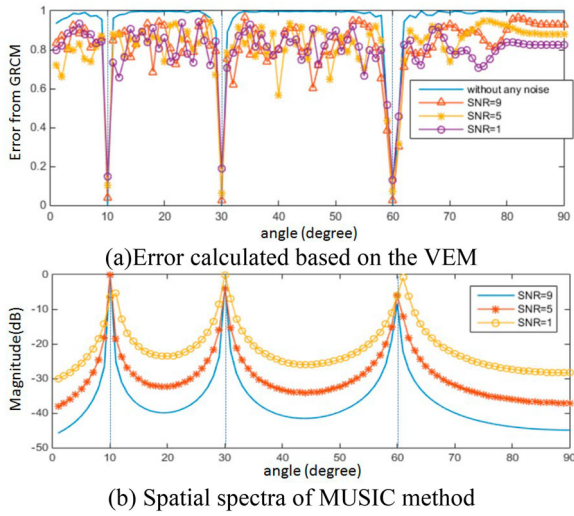
### 3. Simulation and analysis

In this section, simulations to investigate the performance of the VEM-MIPSO method proposed in this paper are conducted. The program for the DOA estimation was written by Matlab R2014b, which could be obtained through the email of clzh0308@126.com. We consider ULAs with 50 sensors, and inter-sensor spacing of  $d = \lambda/2$ . Three echo signals with DOA of  $\theta_1 = 10^\circ$ ,  $\theta_2 = 30^\circ$  and  $\theta_3 = 60^\circ$ , respectively, were generated by program. For the L-shaped array, 50 sensors are used for both x- and y-axis respectively. The pairs of elevation angle and azimuth angle are set as  $(\theta_1, \phi_1) = (10^\circ, 15^\circ)$ ,  $(\theta_2, \phi_2) = (30^\circ, 25^\circ)$  and  $(\theta_3, \phi_3) = (60^\circ, 35^\circ)$ . The snapshot for both arrays was set as 10. All the experiments are carried out in Matlab R2014b on a PC with Intel E33-1231 v3 CPU and 16GB of RAM.

#### 3.1. Analysis of the model

The performance of the VEM model for DOA estimation was firstly analysed in this section. Figure 3 gives the comparison of the performance between the VEM model and MUSIC method under different level of noise jamming for ULA. Table 1 lists the errors calculated by VEM model under different level of SNR. From Figure 3 and Table 1, the following comments could be given.

- (1) Similar to MUSIC method, the VEM could give featured values at the direction of arrival. The difference is that the VEM gives local minimum errors at the target angles, while the MUSIC method gives local maximum values in the spatial spectra at these angles.
- (2) The accuracy of the VEM is higher when the noise level becomes severe. In Figure 3(b), there is a deviation at the angle of  $60^\circ$  when the SNR equals to 1 dB, while the VEM gives all the local minimum value at correct angles. Particularly, calculation error occurs



**Figure 3.** Comparison of the model's performance for ULA. (a) Error calculated based on the VEM. (b) Spatial spectra of MUSIC method.

**Table 1.** Errors for the VEM under different SNR.

SNR	10°	30°	60°
No noise	$8.4e10^{-32}$	$1.3e10^{-31}$	$3.0e10^{-31}$
9	0.04	0.02	0.03
5	0.10	0.06	0.07
1	0.15	0.19	0.13

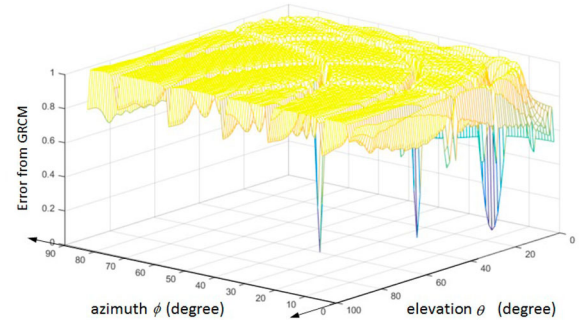
**Table 2.** Errors for the VGM under different SNR.

SNR	$(\Theta, \Phi)$			Time (s)	Time* (s)
	(10°, 15°)	(30°, 25°)	(60°, 35°)		
No noise	$1.8e10^{-31}$	$1.2e10^{-31}$	$1.8e10^{-31}$	0.06	0.1
9	0.05	0.03	0.03	0.06	0.1
5	0.1	0.05	0.1	0.06	0.1
1	0.1	0.2	0.2	0.06	0.1

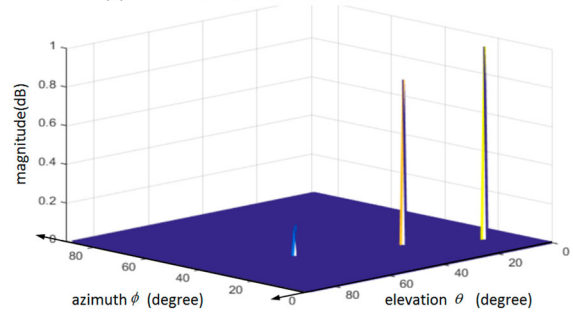
during the calculation process for the MUSIC because of the size of the snapshots. Table 1 lists all the errors for the angles under different level of noise. From Table 1, we can see the minimum values at certain angles remain small level under 0.2. Therefore, the value of 0.2 was selected as the threshold during the calculation of MIPSO algorithm.

Figure 3 gives the comparison of the performance between the VEM and MUSIC method under different level of noise jamming for L-shaped array. Table 2 lists the errors calculated by VEM model under different level of SNR. From Figure 2 and Table 1, the following comments could be given.

(1) Both of the two methods could give featured values at the direction of arrival. The difference is that the VEM gives local minimum errors at the solutions,



(a) Error calculated based on the VEM



(b) Spatial spectra of MUSIC method

**Figure 4.** Comparison of the model's performance for L-shaped array. (a) Error calculated based on the VEM. (b) Spatial spectra of MUSIC method.

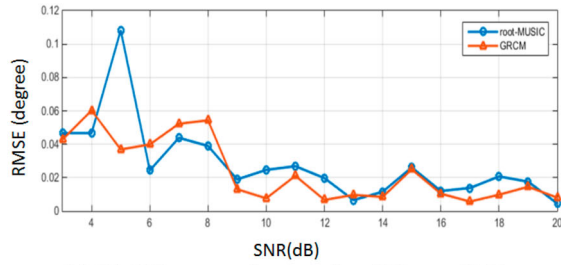
while the MUSIC method gives local maximum values in the spatial spectra at these angles.

- (2) The resolution of the MUSIC method for L-shaped array is higher than that of the VEM model. From the Figure 4 (b), we can see that only the target coordinates have a local maximum value, and the others points own little values without severe fluctuation. While the errors calculated by VEM model, as shown in Figure 4 (a), have a bigger fluctuation among the points.
- (3) The model based on the VEM has quicker speed (0.06 s), to draw the whole error map than the MUSIC method (0.1 s).
- (4) From Table 2, we can see the minimum values at certain angles remain small level under 0.3. Therefore, the value of 0.3 was selected as the threshold during the calculation of MIPSO algorithm.

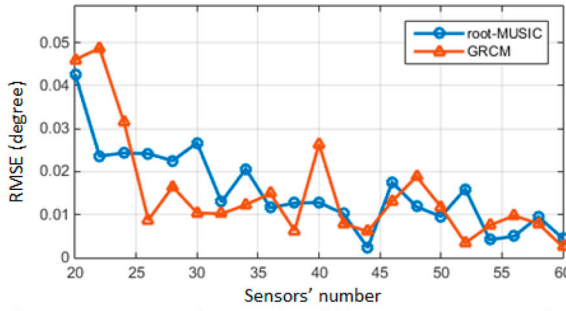
### 3.2. Analysis of calculated results

The performance was evaluated by the root mean square error (RMSE), which is calculated by formula (16).

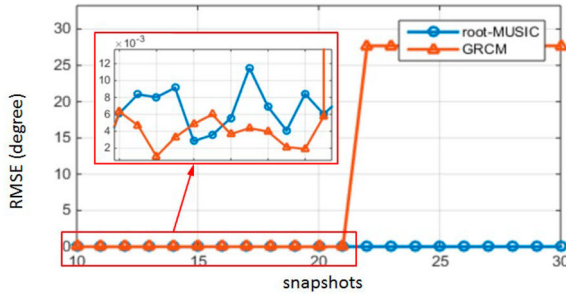
$$RMSE = \sqrt{\frac{\sum_{i=1}^K (\theta_i - \theta_i^*)^2}{K}} \quad (16)$$



(a) RMSE comparison under different SNR



(b) RMSE comparison under different sensors' number

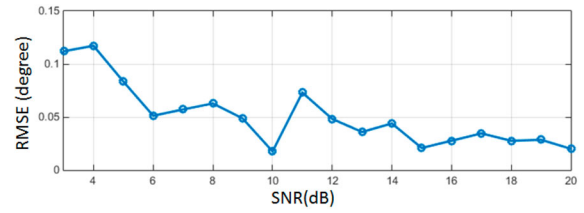


(c) RMSE comparison under different snapshots

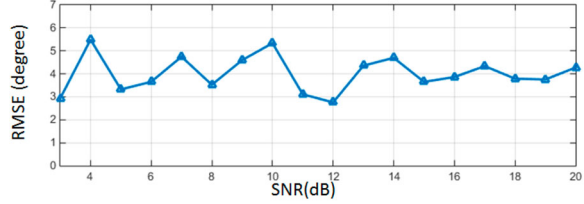
**Figure 5.** RMSE comparison for ULA. (a) RMSE comparison under different SNR. (b) RMSE comparison under different sensors' number. (c) RMSE comparison under different snapshots.

For the ULA, the comparison is carried out between VEM-MIPSO method and the root-MUSIC method. The RMSE comparisons for ULA under different SNR, sensors' number and snapshots are list in Figure 5 (a–c), respectively.

- (1) As shown in Figure 5(a), the experiment for both the two methods under different SNR situation is executed while setting the sensors' number as 50 and the snapshot as 10. Both of the errors generated by the VEM-MIPSO method and the root-MUSIC method could remain at a low level, even when the noise existing in the data is severe. But the time used by the VEM-MIPSO (with average of 0.98 s) is much shorter than that used by the root-MUSIC method (with average of 2.2 s).
- (2) Figure 5(b) gives the experimental results when increasing the sensors' number from 20 to 60 by the step of 2 under the fixing snapshots of 10 and SNR as 20. From the trend of the RMSE, the accuracy from



(a) The method based on the VEM



(b) The method of MEMP

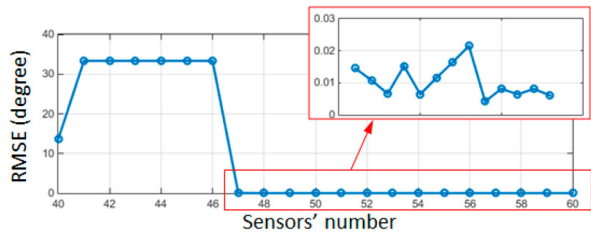
**Figure 6.** RMSE comparison for L-shaped array under different SNR. (a) The method based on the VEM. (b) The method of MEMP.

both of the root-MUSIC method and the VEM-MIPSO method are affected with the same tendency.

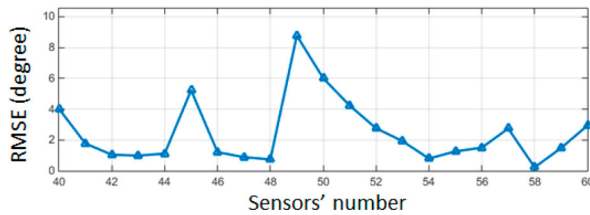
- (3) At last, the snapshot involved in the calculation was adjusted from 10 to 30 by the step of 1 under the fixing sensor's number of 50 and the SNR as 20. And the results are shown in Figure 5(c), from which we can see that the snapshots' number has little effect for the root-MUSIC method, while the RMSE for the VEM increases significantly when the snapshot goes up to a certain value.

For the L-shaped array, the comparisons between the MEMP method and the VEM were illustrated in the Figures 6–8.

- (1) The RMSE comparison for L-shaped array with different noise levels when setting the sensors' number as 50 and the snapshot as 10 is given in Figure 6, from which we can see that although the time used for the VEM-MIPSO method is longer (with average of 1.2 s) than that for the MEMP method (with average of 0.45 s), the RMSE of VEM-MIPSO model is much lesser.
- (2) In Figure 7, we changed the sensors' number of the array from 40 to 60 by step of 1 while fixing the snapshot as 10 and the SNR as 20. Different from the situation for ULA in Figure 5(b), the sensors' number adopted in the array has little effect for the MEMP method, while the RMSE for the VEM-MIPSO method become unstable when the sensors' number decreases down to a certain value.
- (3) Finally, the snapshots of the data were adjusted while fixing the sensors' number as 50 and SNR as 20. The RMSE comparison is described in Figure 8. Similar to the situation for ULA in Figure 5(c), the snapshots

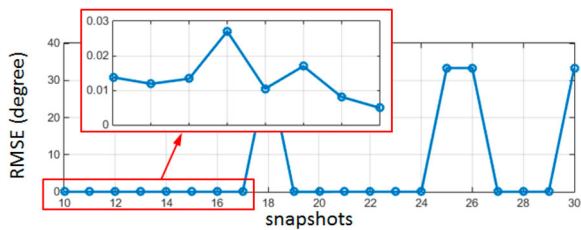


(a) The method based on the VEM

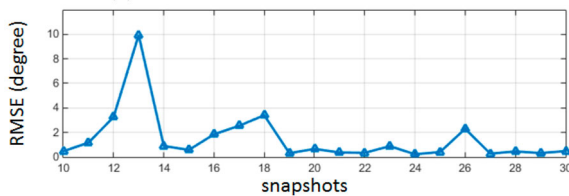


(b) The method of MEMP

**Figure 7.** RMSE for L-shaped array under different sensors' number. (a) The method based on the VEM. (b) The method of MEMP.



(a) The method based on the VEM



(b) The method of MEMP

**Figure 8.** RMSE for L-shaped array under different snapshots. (a) The method based on the VEM. (b) The method of MEMP.

involved in the calculation has little effect for the MEMP method, while the RMSE for the method based on VEM increases significantly when the snapshot goes up to a certain value under a fixed sensors' number. So it is better to set a suitable ratio between the sensors' number and the snapshot before using the VEM-MIPSO method.

## 4. Conclusion

This paper proposed a new method to estimate the DOAs based on the VEM, which had a totally different view for the previous ones. An algorithm named MIPSO was adopted to solve the model without knowing the number of the signal sources. And several simulations with deep

analysis were given to illustrate the efficiency and advantages of the proposed method. Following conclusions were drawn:

### 4.1. Efficiencies

The method based on the VEM maps the solution domain of the DOA to an error space, which is totally different from the current methods mapping the solutions to a spatial spectra space. In this error space, only several solution parameters have the local minimum values which are much lesser than the values around them. This feature of the error space provided the possibility that the minimum value could be calculated by warm intelligent optimization algorithm. Considering the fact that multi-target (multi-angle) need to be found, an algorithm of MIPSO was adopted to find all the targets simultaneously.

### 4.2. Advantages

For the ULA, the method based on VEM and MIPSO algorithm could find the solutions with similar accuracy as the root-MUSIC method but much lesser time. For the L-shaped array, the proposed method could calculate the solutions with much smaller RMSE than the MEMP method.

### 4.3. Limitation and further work

The ratio between the sensors' number and the snapshot should be set at a suitable value to guarantee the efficiency of the VEM-MIPSO method. According to the simulations in this paper, the good accuracy could be obtained with sensors' number of 47 and the snapshot of 10.

For the L-shaped array, the VEM-MIPSO method need more time to finish the calculation than the MEMP method, which was caused by the process of the MIPSO. This limitation could be solved by improving the MIPSO algorithm to accelerate the speed of the convergence of the particles.

In this paper, the situations of the ULA and the L-shaped array are analysed. And only the far-field narrow-band signals are concerned. Further experiments could be tried for different types of sensors' array and signals.

The resolution of the VEM-MIPSO method among different DOAs could be refined by adjusting the parameter  $\delta$  in the future work.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments which resulted in significant improvements to the paper. This work is supported by National Key R&D



Program of China under Grant 2018YFC0604502, National Natural Science Foundation (NNSF) of China under Grant 61573129, Innovative Scientists and Technicians Team of Henan Provincial High Education under Grant 20IRTSTHN019, the Innovative Scientists and Technicians Team of Henan Polytechnic University under Grant T2019-2, the Henan Natural Science Foundation under grant 162300410125, the Key Project of Henan Educational Committee under Grant 17A120009, and the Henan Polytechnic University Doc Fund under Grant B2016-16.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work is supported by National Key R&D Program of China [grant number 2018YFC0604502], National Natural Science Foundation (NNSF) of China [grant number 61573129], Innovative Scientists and Technicians Team of Henan Provincial High Education [grant number 20IRTSTHN019], the Innovative Scientists and Technicians Team of Henan Polytechnic University [grant number T2019-2], the Henan Natural Science Foundation [grant number 162300410125], the Key Project of Henan Educational Committee [grant number 17A120009], and the Henan Polytechnic University Doc Fund [grant number B2016-16].

## References

- Boyd, S., Vandenberghe, L., & Foybusovich, L. (2006). Convex optimization. *IEEE Transactions on Automatic Control*, 51(11), 1859–1859. <https://doi.org/10.1109/TAC.2006.884922>
- Cui, L., Ling, Z., Poon, J., Poon, S. K., Chen, H., Gao, J., Kwan, P., & Fan, K. (2014). A parallel model of independent component analysis constrained by a 5-parameter reference curve and its solution by multi-target particle swarm optimization. *Analytical Methods*, 6(8), 2679–2686. <https://doi.org/10.1039/c3ay42196a>
- Cui, L., Ling, Z., Poon, J., Poon, S. K., Chen, H., Gao, J., Kwan, P., & Fan, K. (2015). Generalized Gaussian reference curve measurement model for high-performance liquid chromatography with diode array detector separation and its solution by multi-target intermittent particle swarm optimization. *Journal of Chemometrics*, 29, 146–153. <https://doi.org/10.1002/cem.2683>
- Cui, L., Poon, J., Poon, S. K., Chen, H., Gao, J., Kwan, P., Fan, K., & Ling, Z. (2014). An improved independent component analysis model for 3D chromatogram separation and its solution by multi-areas genetic algorithm. *BMC Bioinformatics*, 15, 1–10. <https://doi.org/10.1186/1471-2105-15-S12-58>
- Djuric, P. M. (1996). A model selection rule for sinusoids in white Gaussian noise. *IEEE Transactions on Signal Processing*, 44(7), 1744–1751. <https://doi.org/10.1109/78.510621>
- Jin, M.-N., Zhao, Y.-J., & Li, D.-H. (2008). Joint source number detection and DOA estimation via reversible jump MCMC. In *Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. PACIIA '08*. IEEE. <https://doi.org/10.1109/PACIIA.2008.102>

- Lee, S.-H., Choi, I.-O., Kang, M.-S., & Kim, K.-T. (2018). Efficient sparse representation algorithm for accurate DOA estimation of multiple targets with single measurement vector. *Microwave And Optical Technology Letters*, 60(1), 31–37. <https://doi.org/10.1002/mop.30908>
- Luenberger, D. G. (1997). *Optimization by vector space methods*. Wiley.
- Qi, C.-Y., Zhang, Y.-S., Han, Y., & Chen, X.-H. (2004). An algorithm on high resolution DOA estimation with unknown number of signal sources. In *International Conference on Microwave & Millimeter Wave Technology*. IEEE. <https://doi.org/10.1109/ICMMT.2004.1411504>
- Schmidt, R., & Schmidt, R. O. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3), 276–280. <https://doi.org/10.1109/TAP.1986.1143830>
- Thiergart, O., & Habets, E. A. P. (2013). An informed LCMV filter based on multiple instantaneous direction-of-arrival estimates, ICASSP2013, (pp. 659–663).
- Yuen, N., & Friedlander, B. (1996). Asymptotic performance analysis of ESPRIT, higher order ESPRIT, and virtual ESPRIT algorithms. *IEEE Transactions on Signal Processing*, 44(10), 2537–2550. <https://doi.org/10.1109/78.539037>
- Zhang, Y., & Ng, B. P. (2010). MUSIC-like DOA estimation without estimating the number of sources. *IEEE Transactions on Signal Processing*, 58(3), 1668–1676. <https://doi.org/10.1109/TSP.2009.2037074>
- Zhou, H., Hu, G., Shi, J., & Feng, Z. (2018). Orthogonal projection method for DOA estimation in low-altitude environment based on signal subspace. *AEU – International Journal of Electronics and Communications*, 83, 317–321. <https://doi.org/10.1016/j.aeue.2017.09.021>
- Zhou, H.-F., Huang, L., So, H. C., & Li, J. (2018). Performance analysis of G-MUSIC based DOA estimator with random linear array: A single source case. *Signal Processing*, 142, 513–521. <https://doi.org/10.1016/j.sigpro.2017.08.008>

## Appendices

### Appendix 1

Considering (5), Equation (6) is rewritten as

$$\left\{ \begin{array}{l} \min\{\|\mathbf{y}_i^T - \mathbf{c}^T(\theta_i)\|_2^2 \\ = \|\mathbf{w}_i^T \times \bar{\mathbf{D}} + \mathbf{w}_i^T \times \mathbf{M}^{-1} \times \tilde{\mathbf{D}} - \mathbf{c}^T(\theta_i)\|_2^2\} \\ \mathbf{D} = \bar{\mathbf{D}} + \hat{\mathbf{D}} = \begin{bmatrix} \bar{\mathbf{d}}_{r1}^T \\ \bar{\mathbf{d}}_{r2}^T \\ \vdots \\ \bar{\mathbf{d}}_{rt}^T \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{d}}_{r1}^T \\ \hat{\mathbf{d}}_{r2}^T \\ \vdots \\ \hat{\mathbf{d}}_{rt}^T \end{bmatrix} \\ \tilde{\mathbf{D}} = [\tilde{\mathbf{d}}_{c1}, \tilde{\mathbf{d}}_{c2}, \dots, \tilde{\mathbf{d}}_{cM}] = \mathbf{M} \times \hat{\mathbf{D}} = \mathbf{M} \times [\hat{\mathbf{d}}_{c1}, \hat{\mathbf{d}}_{c2}, \dots, \hat{\mathbf{d}}_{cM}] \end{array} \right. \quad (\text{A1})$$

where  $\bar{\mathbf{d}}_{ri}^T$  ( $i = 1, 2, \dots, t$ ) are row vectors whose elements are the mean of  $\hat{\mathbf{d}}_{ri}^T$  ( $i = 1, 2, \dots, t$ );  $\hat{\mathbf{d}}_{ri}^T$  ( $i = 1, 2, \dots, t$ ) are row vectors removed  $\bar{\mathbf{d}}_{ri}^T$  from  $\mathbf{d}_{ri}^T$ . The transformation from  $\hat{\mathbf{D}}$  to  $\tilde{\mathbf{D}}$  is a linear one so that the column vectors of  $\tilde{\mathbf{d}}_{ci}$  are uncorrelated and their variances equal unity. In other words, the covariance matrix of  $\tilde{\mathbf{d}}_{ci}$  equals the identity matrix as shown in (A2). The calculation of matrix  $\mathbf{M}$  is given Appendix 2.

$$E\{\tilde{\mathbf{d}}_{ci} \times \tilde{\mathbf{d}}_{ci}^T\} = \mathbf{I}_{t \times t} \quad (\text{A2})$$

If we set

$$\begin{cases} d_i = \mathbf{w}_i^T \times \hat{\mathbf{D}} \\ \mathbf{b}_i^T = \mathbf{w}_i^T \times \mathbf{M}^{-1} \end{cases} \quad (\text{A3})$$

where  $d_i$  is a constant for a specific  $\mathbf{w}_i^T$ . So (A1) can be transformed as

$$\begin{cases} \min\{\|\bar{\mathbf{b}}_i^T \times \tilde{\mathbf{D}}^* - \mathbf{c}(\theta_i)\|_2^2\} \\ \bar{\mathbf{b}}_i^T = [\mathbf{b}_i^T, d_i] \\ \tilde{\mathbf{D}}^* = \begin{bmatrix} \tilde{\mathbf{D}} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{d}}_{c1}, \tilde{\mathbf{d}}_{c2}, \dots, \tilde{\mathbf{d}}_{cM} \\ 1 \end{bmatrix} = [\tilde{\mathbf{d}}_{c1}^*, \tilde{\mathbf{d}}_{c2}^*, \dots, \tilde{\mathbf{d}}_{cM}^*] \\ E\{\tilde{\mathbf{D}}^* \times \tilde{\mathbf{D}}^{*T}\} = M \times \mathbf{I}_{(t+1) \times (t+1)} \end{cases} \quad (\text{A4})$$

The reason why  $E\{\tilde{\mathbf{D}}^* \times \tilde{\mathbf{D}}^{*T}\} = M \times \mathbf{I}_{(t+1) \times (t+1)}$  will be explained in Appendix 3. According to the Karush-Kuhn-Tucker (KKT) conditions (Boyd et al., 2006), the solution of (A4) can be obtained at points satisfying

$$F(\bar{\mathbf{b}}_i^T) = \sum_{j=1}^M 2 \times \tilde{\mathbf{d}}_j^{*T} \times (\bar{\mathbf{b}}_i^T \times \tilde{\mathbf{d}}_j^* - \mathbf{c}(j; \theta_i)) = 0 \quad (\text{A5})$$

where  $\mathbf{c}(j; \theta_i)$  is the value of  $j$ th element under parameter  $\theta_i$ . Newton's method is used to solve (A6). The Jacobian matrix of (A5) is

$$JF(\bar{\mathbf{b}}_i^T) = \sum_{j=1}^M 2 \times \tilde{\mathbf{d}}_j^* \times \tilde{\mathbf{d}}_j^{*T} \quad (\text{A6})$$

Therefore, the following approximative Newton iteration (Luenberger 1997) is obtained as

$$\begin{aligned} \bar{\mathbf{b}}_i^+ &= \bar{\mathbf{b}}_i - \frac{\sum_{j=1}^M 2\tilde{\mathbf{d}}_j^* (\bar{\mathbf{b}}_i^T \tilde{\mathbf{d}}_j^* - \mathbf{c}(j; \theta_i))}{\sum_{j=1}^M 2\tilde{\mathbf{d}}_j^* \times \tilde{\mathbf{d}}_j^{*T}} \\ &= \frac{\sum_{j=1}^M \tilde{\mathbf{d}}_j^* \times \mathbf{r}_i(j; \theta)}{M \times \mathbf{I}} = \frac{1}{M} \times \tilde{\mathbf{D}}^* \times \mathbf{c}(j; \theta_i) \end{aligned} \quad (\text{A7})$$

The curve of  $\mathbf{y}_i^T$  can be calculated by

$$\mathbf{y}_i^T = \bar{\mathbf{b}}_i^T \times \tilde{\mathbf{D}}^* = \mathbf{c}^T(\theta_i) \times \left( \frac{1}{M} \times \tilde{\mathbf{D}}^{*T} \times \tilde{\mathbf{D}}^* \right) \quad (\text{A8})$$

where  $M$  is the number of the column in  $\tilde{\mathbf{D}}^*$ . Equation (A8) is called VGM.

## Appendix 2

In this appendix, we will introduce the method to transform  $\hat{\mathbf{d}}_{ci}/\hat{\mathbf{D}}$  to  $\tilde{\mathbf{d}}_{ci}/\tilde{\mathbf{D}}$  using the Eigen-value decomposition (EVD) of the covariance matrix  $E\{\hat{\mathbf{D}} \times \hat{\mathbf{D}}^T\} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$ , where  $\mathbf{E}$  is the orthogonal matrix of eigenvectors of  $E\{\hat{\mathbf{D}} \times \hat{\mathbf{D}}^T\}$  and  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$  is the diagonal matrix of its eigenvalues. Whitening can be done by

$$\tilde{\mathbf{D}} = \mathbf{\Lambda}^{-1/2} \mathbf{E}^T \hat{\mathbf{D}} = \mathbf{M}\hat{\mathbf{D}} \quad (\text{A9})$$

where  $\mathbf{\Lambda}^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \dots, \lambda_m^{-1/2})$ . The number of the eigenvalue kept in  $\mathbf{\Lambda}$  is always determined by a threshold, i.e.  $m$  is always less than  $t$ . However, two insufficiencies exist: (1) small signals will be missed; (2) it is difficult to fix the threshold especially when severe noises exist. In order to overcome these insufficiencies, we set  $m$  equal to  $t$  in (A9) to keep all information. Following, the procedure of transformation from  $\hat{\mathbf{D}}$  to  $\tilde{\mathbf{D}}$  will be given step by step:

**Step 1:** calculate the covariance matrix of  $\hat{\mathbf{D}}$  as

$$\mathbf{C} = \hat{\mathbf{D}} \times \hat{\mathbf{D}}^T \quad (\text{A10})$$

**Step 2:** obtain the eigenvalues and eigenvectors of  $\mathbf{C}$  by using the eigenvalue decomposition.

$$[\mathbf{E}, \mathbf{\Lambda}] = \text{EVD}(\mathbf{C}) \quad (\text{A11})$$

where  $\mathbf{E}$  is the orthogonal matrix of eigenvectors of  $\mathbf{C}$ , and  $\mathbf{\Lambda}$  is the diagonal matrix of its eigenvalues.

**Step 3:** calculate whiten matrix by

$$\mathbf{M}_{t \times t} = \text{inv}[\text{sqrt}(\mathbf{\Lambda})] \times \mathbf{E}^T \quad (\text{A12})$$

**Step 4:** calculate whitening data set by

$$\tilde{\mathbf{D}} = \mathbf{M}_{t \times t} \times \hat{\mathbf{D}} \quad (\text{A13})$$

where  $\mathbf{M}_{t \times t}$  is an invertible matrix, whose inverse matrix is

$$\mathbf{M}_{t \times t}^{-1} = \mathbf{E} \times \text{sqrt}(\mathbf{\Lambda}) \quad (\text{A14})$$

## Appendix 3

In this appendix, we will explain why

$$E\{\tilde{\mathbf{D}}^* \times \tilde{\mathbf{D}}^{*T}\} = M \times \mathbf{I}_{(t+1) \times (t+1)} \quad (\text{A15})$$

From the definition in (A4), we have

$$\tilde{\mathbf{D}}^* = \begin{bmatrix} \tilde{\mathbf{D}} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{d}}_{c1}, \tilde{\mathbf{d}}_{c2}, \dots, \tilde{\mathbf{d}}_{cM} \\ 1 \end{bmatrix} \quad (\text{A16})$$

where  $M$  is column number of the matrix  $\tilde{\mathbf{D}}$ ,  $\tilde{\mathbf{d}}_{ci}^T (i = 1, 2, \dots, M)$  are the column vectors in  $\tilde{\mathbf{D}}$ . Then, we have

$$\begin{aligned} \tilde{\mathbf{D}}^* \times \tilde{\mathbf{D}}^{*T} &= \begin{bmatrix} \tilde{\mathbf{d}}_{c1}, \tilde{\mathbf{d}}_{c2}, \dots, \tilde{\mathbf{d}}_{cM} \\ 1 \end{bmatrix} \times \begin{bmatrix} \tilde{\mathbf{d}}_{c1}^T & 1 \\ \tilde{\mathbf{d}}_{c2}^T & 1 \\ \vdots & \vdots \\ \tilde{\mathbf{d}}_{cM}^T & 1 \end{bmatrix} \\ &= \sum_{i=1}^M \begin{bmatrix} \tilde{\mathbf{d}}_{ci} \\ 1 \end{bmatrix} \times \begin{bmatrix} \tilde{\mathbf{d}}_{ci}^T & 1 \end{bmatrix} \\ &= \sum_{i=1}^M \begin{bmatrix} \tilde{\mathbf{d}}_{ci} \times \tilde{\mathbf{d}}_{ci}^T & \tilde{\mathbf{d}}_{ci} \\ \tilde{\mathbf{d}}_{ci}^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^M \tilde{\mathbf{d}}_{ci} \times \tilde{\mathbf{d}}_{ci}^T & \sum_{i=1}^M \tilde{\mathbf{d}}_{ci} \\ \sum_{i=1}^M \tilde{\mathbf{d}}_{ci}^T & \sum_{i=1}^M 1 \end{bmatrix} \end{aligned} \quad (\text{A17})$$

Because (A2), we have

$$\sum_{i=1}^M \tilde{\mathbf{d}}_{ci} \times \tilde{\mathbf{d}}_{ci}^T = M \times \mathbf{I}_{t \times t} \quad (\text{A18})$$

Because the transformation from  $\hat{\mathbf{x}}_{ci}$  to  $\tilde{\mathbf{x}}_{ci}$  did not change the original amplitude, so we have

$$\begin{cases} \sum_{i=1}^M \tilde{\mathbf{d}}_{ci} = \mathbf{0}_{t \times 1} \\ \sum_{i=1}^M \tilde{\mathbf{d}}_{ci}^T = \mathbf{0}_{1 \times t} \end{cases} \quad (\text{A19})$$

Substitute (A18) and (A19) in (A17), we have (A15).