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 ith 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 mth sensor is a superposition of time-shifted versions of all the sources corrupted with noise as following

$$z_m(t) = \sum_{i=1}^{K} s_i(t - (m - 1)\frac{d}{c_0} \sin \theta_i) + n_m(t)$$

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

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

$$Z(t) = A(\theta)S(t) + N(t)$$  \hspace{1cm} (2)

where $Z(t) = [z_1(t), z_2(t), \ldots, z_M(t)]^T$ with the superscript ()$^T$ denoting the transpose is a complex matrix required by the sensors; $S(t) = [s_1(t), s_2(t), \ldots, s_K(t)]^T$ with the sources number of $K$ contains all the narrow-band far-field source signals; $A(\theta) = [a(\theta_1), a(\theta_2), \ldots, a(\theta_K)]$ is the steering matrix with $a(\theta) = [1, a_1^j, \ldots, a_{K}^{j\lambda}]$, $a_{ij} = e^{-j2\pi d/j \sin(\theta_i)}$; $\lambda$ is the wavelength of the source; and $N(t) = [n_1(t), n_2(t), \ldots, n_M(t)]^T$ is an unknown noise matrix.

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

\[
A_x([\theta, \phi]) = [A_x([\theta, \phi]) \; A_y([\theta, \phi])]
\]

\[
A_x([\theta, \phi]) = \begin{bmatrix}
1 & e^{-j2\pi d \cos \phi \sin \theta_1/\lambda} & 1 \\
\vdots & \vdots & \vdots \\
e^{-j2\pi d(M-1) \cos \phi \sin \theta_{M}/\lambda} & \cdots & e^{-j2\pi d(M-1) \cos \phi \sin \theta_L/\lambda} \\
\end{bmatrix}
\]

\[
A_y([\theta, \phi]) = \begin{bmatrix}
1 & e^{-j2\pi d \sin \phi \sin \theta_1/\lambda} & 1 \\
\vdots & \vdots & \vdots \\
e^{-j2\pi d(N-1) \sin \phi \sin \theta_{N}/\lambda} & \cdots & e^{-j2\pi d(N-1) \sin \phi \sin \theta_L/\lambda} \\
\end{bmatrix}
\]
2.2. DOA estimation model based on VEM

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

\[
\begin{align*}
\mathbf{D}(t) &= \mathbf{B}(\theta) + \mathbf{V}(t) \\
\mathbf{D}(t) &= [\mathbf{Z}(t)]^T = [\text{real}(\mathbf{Z}(t))]^T \\
\mathbf{B}(t) &= [\mathbf{s}(t)]^T \\
\mathbf{C}(\theta) &= [\mathbf{a}(\theta)]^T = [\text{real}(\mathbf{a}(\theta))]^T \\
\mathbf{V}(t) &= [\mathbf{n}(t)]^T = [\text{real}(\mathbf{n}(t))]^T
\end{align*}
\]

where \( \mathbf{D}(t) = [d_1(t), d_2(t), \ldots, d_M(t)] \) 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; \( \mathbf{B}(t) = [\mathbf{s}_1(t), \mathbf{s}_2(t), \ldots, \mathbf{s}_K(t)] \) is the transpose of \( \mathbf{S}(t); \mathbf{C}(\theta) = [\mathbf{c}(\theta_1), \mathbf{c}(\theta_2), \ldots, \mathbf{c}(\theta_K)]^T \) 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 \( \gamma(\theta) \) is obtained, which \( \mathbf{C}(\theta) \) is the transpose of the real part of the noise matrix \( \mathbf{N}(t) \).

Equation (6) is called VEM, which calculated out the error between the known vector \( \mathbf{c}(\theta_i) \) and the generated vector \( \mathbf{y}_i^T \).

\[
\mathbf{y}_i^T = \mathbf{c}^T(\theta_i) \times \left( \frac{1}{M} \times \mathbf{D}^* \times \mathbf{D}^* \right)
\]

where \( \mathbf{D}^* \) 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

\[
\begin{align*}
\mathbf{c}^T(\theta_i) &= \text{real}(\mathbf{a}(\theta_i)) \\
\mathbf{a}(\theta_i) &= [a_{\theta_1}, a_{\theta_2}, \ldots, a_{\theta_M}]^T \\
a_{\theta_i} &= e^{-j2\pi d_i/\lambda \sin(\theta_i)}
\end{align*}
\]

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.

\[
\begin{align*}
\mathbf{D}(t) &= \mathbf{B}(\theta^1, \phi^1) + \mathbf{V}(t) \\
\mathbf{D}(t) &= [\mathbf{Z}(t)]^T = [\text{real}(\mathbf{Z}(t))]^T \\
\mathbf{B}(t) &= [\mathbf{s}(t)]^T \\
\mathbf{C}(\theta) &= [\mathbf{a}(\theta_1, \phi_1)]^T = [\text{real}(\mathbf{a}(\theta_1, \phi_1))]^T \\
\mathbf{V}(t) &= [\mathbf{n}(t)]^T = [\text{real}(\mathbf{n}(t))]^T
\end{align*}
\]

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

\[
\begin{align*}
\min \| \mathbf{y}_i^T - \mathbf{c}^T(\theta_i, \phi_i) \|^2_2, \forall i & \\
\mathbf{y}_i^T = \mathbf{c}^T(\theta_i, \phi_i) \times \left( \frac{1}{r} \times \mathbf{D}^* \times \mathbf{D}^* \right)
\end{align*}
\]
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 \( \{t_i\}_{i=1}^{K} \) 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 \( \{t_i\}_{i=1}^{K} \) is picked out from these \( \{t_i\}_{i=1}^{K} \), where \( K \) is the number of the sources, and much smaller than \( P \).

For calculation convenience, the errors between \( y_i^T \) and \( c^T(\theta_i) \) are defined as

\[
\varepsilon(\theta) = \left[ \varepsilon(\theta_1), \varepsilon(\theta_2), \ldots, \varepsilon(\theta_p) \right]^T
\]

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 \( c^T(\theta_i) \) and \( y_i^T \) 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 ULA 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° 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    |

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.

Table 2. Errors for the VGM under different SNR.

| SNR     | (10°, 15°) | (30°, 25°) | (60°, 35°) | Time (s) | Time* (s) |
|---------|------------|------------|------------|----------|-----------|
| 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       | 0.10       | 0.05       | 0.1        | 0.06     | 0.1       |
| 1       | 0.10       | 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, 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).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{K} \left( \theta_i - \theta_i^* \right)^2}{K}}$$ (16)
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 listed 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 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 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
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 201RTSTHN019, 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 201RTSTHN019], the Innovative Scientists and Technicians Team of Henan Polytechnic University [grant number T2019-2], the Henan Natural Science Foundation [grant number 162300410125], the Henan Polytechnic University Doc Fund [grant number B2016-16], National Key R&D Program of China [grant number B2016-16], National Natural Science Foundation (NNSF) of China [grant number 17A120009], the Henan Educational Committee [grant number 162300410125], the Key Project of Henan Education [grant number 20IRTSTHN019], the Henan Natural Science Foundation [grant number 2019010000125], the Innovative Scientists and Technicians Team of Henan Province under grant 2019020000125, National Natural Science Foundation of China under Grant 2016M590601, 2017M590758, 2018M630707, the Key Project of Henan Education [grant number T2019-2, the Henan Natural Science Foundation under Grant T2019-2, the Henan Natural Science Foundation under Grant 2019020000125]. The research is also supported by the National Natural Science Foundation of China under Grant 61725103, the National Natural Science Foundation of China under Grant 61703177. The research is also supported by the National Natural Science Foundation of China under Grant 61734011, the National Natural Science Foundation of China under Grant 61773001, the Key Project of Henan Education [grant number T2019-2, the Henan Natural Science Foundation under Grant T2019-2, the Henan Natural Science Foundation under Grant 2019020000125]. The research is also supported by the National Natural Science Foundation of China under Grant 61734011, the National Natural Science Foundation of China under Grant 61773001, the Key Project of Henan Education [grant number T2019-2, the Henan Natural Science Foundation under Grant T2019-2, the Henan Natural Science Foundation under Grant 2019020000125].

References

Boyd, S., Vandenberghe, L., & Faybusovich, 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 Chromatography, 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-S8

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. PACIAI 08. IEEE. https://doi.org/10.1109/PACIAI.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 of high resolution DOA estimation with unknown number of signal sources. In *International Conference on Microwave & Millimeter Wave Technology*. IEEE. https://doi.org/10.1109/ICMWM.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

Thiengtara, O., & Habets, E. A. P. (2013). An informed LCMV filter based on multiple instantaneous direction-of-arrival estimates, ICASSP 2013, (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

\[
\begin{align*}
\min & \left\{ \| y^T - c^T(\theta) \|_2^2 \right\} \\
= & \| w^T x + w^T x M^{-1} x D - c^T(\theta) \|_2^2 \\
D & = D + \mathbf{D} \\
D & = [d_{1}, d_{2}, \ldots, d_{n}] M \times D = M \times [d_{1}, d_{2}, \ldots, d_{n}] \\
(A1)
\end{align*}
\]

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

\[
E[\mathbf{d}_{i}^T \times \mathbf{d}_{j}^T] = I_{t \times t} \\
(A2)
\]
If we set
\[
\begin{align*}
d_i &= w_i^T \times \tilde{D} \\
b_i^T &= w_i^T \times M^{-1}
\end{align*}
\] (A3)
where \(d_i\) is a constant for a specific \(w_i^T\). So (A1) can be transformed as
\[
\begin{align*}
\min\{||b_i^T \times \tilde{D}^* - c(\theta)||^2_2\} \\
b_i^T = [b_i^T, d_i] \\
\tilde{D}^* = [\hat{d}_1^*, \hat{d}_2^*, \ldots, \hat{d}_M^*] = [d_1^*, d_2^*, \ldots, d_M^*] \\
E[\tilde{D}^* \times b_i^T] = M \times I_{(t+1) \times (t+1)}
\end{align*}
\] (A4)

The reason why \(E(\tilde{D}^* \times \tilde{D}^{*T}) = M \times I_{(t+1) \times (t+1)}\) 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
\[
F(b_i^T) = \sum_{j=1}^{M} 2 \times \hat{d}_j^T \times (b_i^T \times \hat{d}_j^* - c(j; \theta)) = 0
\] (A5)
where \(c(j; \theta)\) is the value of \(j\)th element under parameter \(\theta\). Newton’s method is used to solve (A6). The Jacobian matrix of (A5) is
\[
JF(b_i^T) = \sum_{j=1}^{M} 2 \times \hat{d}_j^T \times \hat{d}_j^T
\] (A6)

Therefore, the following approximative Newton iteration (Luenberger 1997) is obtained as
\[
\begin{align*}
b_i^T &= b_i^T - \frac{\sum_{j=1}^{M} 2 \times \hat{d}_j^T \times (b_i^T \times \hat{d}_j^* - c(j; \theta))}{\sum_{j=1}^{M} 2 \times \hat{d}_j^T \times \hat{d}_j^T} \\
&= \frac{\sum_{j=1}^{M} \hat{d}_j^T \times r(j; \theta)}{M \times I} = \frac{1}{M} \times \tilde{D}^* \times c(j; \theta)
\end{align*}
\] (A7)

The curve of \(y_i^T\) can be calculated by
\[
y_i^T = b_i^T \times \tilde{D}^* = c^T(\theta) \times \left(\frac{1}{M} \times \tilde{D}^{*T} \times \tilde{D}^*\right)
\] (A8)
where \(M\) is the number of the column in \(\tilde{D}^*\). Equation (A8) is called VGM.

**Appendix 2**

In this appendix, we will introduce the method to transform \(\hat{d}_i/\tilde{D}\) to \(\hat{d}_i/\hat{D}\) using the Eigen-value decomposition (EVD) of the covariance matrix \(E(\tilde{D} \times \tilde{D}^{*T}) = \mathbf{E} \Lambda \mathbf{E}^T\), where \(\mathbf{E}\) is the orthogonal matrix of eigenvectors of \(E(\tilde{D} \times \tilde{D}^{*T})\) and \(\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)\) is the diagonal matrix of its eigenvalues. Whitening can be done by
\[
\hat{D} = \Lambda^{-1/2} \mathbf{E}^{T} \tilde{D} = M \hat{D}
\] (A9)
where \(\Lambda^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_m^{-1/2})\). The number of the eigenvalue kept in \(\Lambda\) 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 \(\hat{D}\) to \(\hat{D}\) will be given step by step:

**Step 1:** calculate the covariance matrix of \(\hat{D}\) as
\[
C = \hat{D} \times \hat{D}^T
\] (A10)

**Step 2:** obtain the eigenvalues and eigenvectors of \(C\) by using the eigenvalue decomposition.
\[
[E, \Lambda] = \text{EVD}(C)
\] (A11)
where \(E\) is the orthogonal matrix of eigenvectors of \(C\), and \(\Lambda\) is the diagonal matrix of its eigenvalues.

**Step 3:** calculate whitening matrix by
\[
M_{t \times t} = \text{inv}(\sqrt{\Lambda}) \times \mathbf{E}^T
\] (A12)

**Step 4:** calculate whitening data set by
\[
\hat{D} = M_{t \times t} \times \hat{D}
\] (A13)
where \(M_{t \times t}\) is an invertible matrix, whose inverse matrix is
\[
M_{t \times t}^{-1} = E \times \text{sqrt}(\Lambda)
\] (A14)

**Appendix 3**

In this appendix, we will explain why
\[
E(\tilde{D}^* \times \tilde{D}^{*T}) = M \times I_{(t+1) \times (t+1)}
\] (A15)

From the definition in (A4), we have
\[
\tilde{D}^* = [\hat{d}_1^*, \hat{d}_2^*, \ldots, \hat{d}_M^*] = \begin{pmatrix}
\hat{d}_1^* \\
\hat{d}_2^* \\
\vdots \\
\hat{d}_M^*
\end{pmatrix}
\] (A16)
where \(M\) is column number of the matrix \(\tilde{D}\), \(\hat{d}_i^* (i = 1, 2, \ldots, M)\) are the column vectors in \(\hat{D}\). Then, we have
\[
\tilde{D}^* \times \tilde{D}^{*T} = \begin{pmatrix}
\hat{d}_1^* \\
\hat{d}_2^* \\
\vdots \\
\hat{d}_M^*
\end{pmatrix} \times \begin{pmatrix}
\hat{d}_1^* \\
\hat{d}_2^* \\
\vdots \\
\hat{d}_M^*
\end{pmatrix} = \sum_{i=1}^{M} \hat{d}_i^T \times \hat{d}_i^T
\] (A17)

Because (A2), we have
\[
\sum_{i=1}^{M} \hat{d}_i^T \times \hat{d}_i^T = M \times I_{t \times t}
\] (A18)

Because the transformation from \(\hat{x}_i\) to \(\tilde{x}_i\) did not change the original amplitude, so we have
\[
\sum_{i=1}^{M} \hat{d}_i^T = 0_{t \times 1}
\] (A19)

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