

Multiple Signal Classification (MUSIC) Algorithm for Angular Estimation

Multiple Signal Classification (MUSIC) is a classic algorithm for angular information estimation in wireless communications. In this article, I first model the far-field wireless channel in the view of angular-domain, and then describe the MUSIC algorithm for angular information extraction in the modeled angular-domain channel. The simulation codes will be attached for check.

I. FAR-FIELD CHANNEL MODEL

We consider a single-input-multiple-output (SIMO) system where a single-antenna transmitter sends pilot signal x to the receiver with N_r antennas. As shown in Fig. 1(a), each of the N_r antennas receives a copy of the signal x . When the distance between the transmitter and receiver is large enough, the signal x can be assumed to arrive at each receiver antenna in a parallel way, as illustrated in Fig. 1(b). Due to the separation d between two receiver antenna, length of the parallel path from the transmitter to each receiver antenna is different. As shown in Fig. 1(b), the distance d_n between transmitter and receiver antenna n is given by

$$d_n = d_1 + (n - 1)d \sin(\theta), \quad n = 1, 2, \dots, N, \quad (1)$$

where d_1 is the distance from the transmitter to the 1st antenna of the receiver. On this basis, the received signal at the n th antenna is computed as

$$y_n = a(d_n)e^{j2\pi f_c \frac{d_n}{c}} x = a(d_n)e^{j\frac{2\pi}{\lambda} f_c d_1} e^{j\frac{2\pi}{\lambda} (n-1)d \sin(\theta)} x \stackrel{(a)}{\approx} a(d_1)e^{j\frac{2\pi}{\lambda} f_c d_1} e^{j\frac{2\pi}{\lambda} (n-1)d \sin(\theta)} x. \quad (2)$$

where $a(d_n)$ is the large-scale attenuation (e.g., pathloss), $e^{j2\pi f_c \frac{d_n}{c}}$ denotes the phase change of the signal due to propagation time delay, (a) is because of the fact that $a(d_1) \approx a(d_2) \approx \dots \approx a(d_N)$ for large-scale attenuation. Based on (2), the received signal at all the N antennas are given by

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = a(d_1)e^{j\frac{2\pi}{\lambda} f_c d_1} x \underbrace{\begin{bmatrix} 1 \\ e^{j\frac{2\pi}{\lambda} d \sin(\theta)} \\ \vdots \\ e^{j\frac{2\pi}{\lambda} (N-1)d \sin(\theta)} \end{bmatrix}}_{\mathbf{e}(\theta)} = a(d_1)e^{j\frac{2\pi}{\lambda} f_c d_1} x \mathbf{e}(\theta) \quad (3)$$

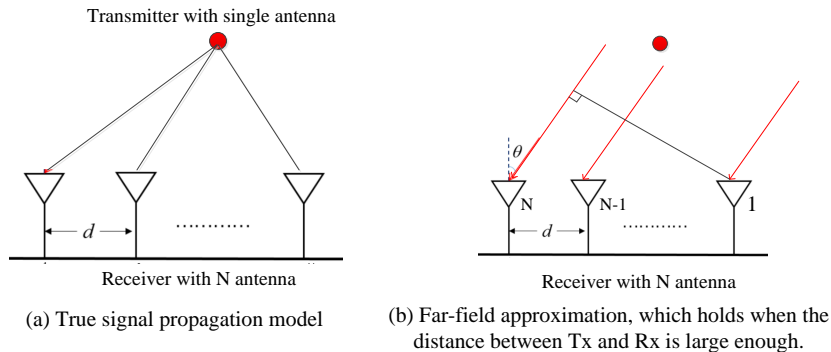


Fig. 1: Far field channel model in free space with only line-of-site (LoS) path.

Note that (3) represents the received signals along with a certain path with length d_1 and angle-of-arrival (AoA) θ . In practise, the signal from transmitter can arrive at the receiver along with many different paths under different length distance and AoA. An extension of the (3) to the multi-path scenario is given by

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{e}(\theta_1) & \mathbf{e}(\theta_2) & \dots & \mathbf{e}(\theta_L) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_L \end{bmatrix}}_{\mathbf{x}} + \mathbf{n}, \quad (4)$$

where θ_l is the AoA of the l th path, and $x_l = a(d_{1,l})e^{j\frac{2\pi}{\lambda}f_c d_{1,l}}x$ is for the l the path, and $\mathbf{n} \in \mathcal{C}^{N_r \times 1}$ is the noise with norm distribution $\mathcal{CN}(\mathbf{0}, \mathbf{I})$.

II. MUSIC ALGORITHM THEORY

In this section, we theoretically describe the MUSIC algorithm. Note that the MUSIC can only work in the scenario with $N_r > L$, i.e., the number of receive antenna is larger than the number of paths in the channel \mathbf{A} .

According to (4), the received signal can be expressed as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}. \quad (5)$$

Due to the randomness of noise \mathbf{n} , the \mathbf{y} is a random vector with covariance matrix

$$\begin{aligned} \mathbf{R}_y = \mathbb{E}[\mathbf{y}\mathbf{y}^*] &= \mathbb{E}[(\mathbf{A}\mathbf{x} + \mathbf{n})(\mathbf{A}\mathbf{x} + \mathbf{n})^*] \\ &\stackrel{(b)}{=} \mathbf{A}\mathbb{E}[\mathbf{x}\mathbf{x}^*]\mathbf{A}^* + \mathbb{E}[\mathbf{n}\mathbf{n}^*] \\ &= \mathbf{A}\mathbf{R}_x\mathbf{A}^* + \mathbf{R}_n, \end{aligned} \quad (6)$$

where (b) is due to the fact that $\mathbb{E}[\mathbf{x}\mathbf{n}^*] = 0$. Since the $\mathbf{R}_y \in \mathcal{C}^{N_r \times N_r}$ in (6) is symmetric, we conduct the eigenvalue decomposition (EVD) for the \mathbf{R}_y given by

$$\mathbf{R}_y = [\mathbf{U}_x \quad \mathbf{U}_n] \begin{bmatrix} \Sigma_x & \\ & \Sigma_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_x^* \\ \mathbf{U}_n^* \end{bmatrix} = \mathbf{U}_x \Sigma_x \mathbf{U}_x^* + \mathbf{U}_n \Sigma_n \mathbf{U}_n^*, \quad (7)$$

where $\mathbf{U}_x \in \mathcal{C}^{N_r \times L}$, $\mathbf{U}_n \in \mathcal{C}^{N_r \times (N_r - L)}$, $\Sigma_x \in \mathcal{C}^{L \times L}$, and $\Sigma_n \in \mathcal{C}^{(N_r - L) \times (N_r - L)}$. It is clear that $\mathbf{U}_x \Sigma_x \mathbf{U}_x^*$ is the EVD of the $\mathbf{A}\mathbf{R}_x\mathbf{A}^*$ and $\mathbf{U}_n \Sigma_n \mathbf{U}_n^*$ is the EVD of \mathbf{R}_n in (6). Based on the properties of EVD, we have

$$\mathbf{A}\mathbf{R}_x\mathbf{A}^*\mathbf{U}_n = \mathbf{U}_x \Sigma_x \mathbf{U}_x^* \mathbf{U}_n = \mathbf{0},$$

which further yields¹

$$(\mathbf{R}_x^{-1}(\mathbf{A}\mathbf{A}^*)^{-1}\mathbf{A}^*)\mathbf{A}\mathbf{R}_x\mathbf{A}^*\mathbf{U}_n = \mathbf{0} \Rightarrow \mathbf{A}^*\mathbf{U}_n = \mathbf{0},$$

which implied

$$\mathbf{e}(\theta_l)\mathbf{U}_n = \mathbf{0} \quad (8)$$

because of the expression of \mathbf{A} in (4). It means a path angle θ_l appears when the

$$P(\theta) = \frac{1}{|\mathbf{e}(\theta_l)\mathbf{U}_n|^2} \rightarrow +\infty. \quad (9)$$

¹Since different paths will independently affect the transmit signal x , the \mathbf{R}_x is full rank and its inverse exists.

III. MUSIC ALGORITHM IMPLEMENTATION

In real system, one can sample the received signal \mathbf{y} at different time $t \in \{1, 2, \dots, T\}$ yielding $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(T)$. Using these samples, the covariance matrix is computed as

$$\hat{\mathbf{R}}_{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^T \mathbf{y}(t)\mathbf{y}(t)^*. \quad (10)$$

Taking the EVD for the $\hat{\mathbf{R}}_{\mathbf{y}}$ yields

$$\hat{\mathbf{R}}_{\mathbf{y}} = \hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{U}}^*. \quad (11)$$

We sort the eigenvalues in $\hat{\Sigma}$ at descending order and accordingly adjust the corresponding eigenvectors in $\hat{\mathbf{U}}$. In this way, we obtain the $\hat{\mathbf{U}}_{\mathbf{n}}$ as in (7) by taking the last $N_r - L$ columns of the $\hat{\mathbf{U}}$. For different $\theta \in (0, \pi]$ for the function $P(\theta)$ in (9), we can obtain different value. According to (9), the θ yielding the locally maximum $P(\theta)$ is a target AoA. Summaries of the MUSIC algorithm for AoA extraction is given in Algorithm 1.

Algorithm 1 MUSIC algorithm for AoA Extraction

Require: $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(T)$, resolution for $\theta \in (0, \pi]$

1. Compute the covariance matrix $\mathbf{R}_{\mathbf{y}}$ in (10).
 2. Find the noise space matrix $\hat{\mathbf{U}}_{\mathbf{n}}$.
 3. Based on the resolution for $\theta \in (0, \pi]$, test each θ and find the ones yielding the locally maximum $P(\theta)$ in (9). These θ s are the expected AoAs.
-