Delft University of Technology

EE4715 Array processing

Array Parameter Estimation

Authors:
Weijia Yi (5511127) & Xuan Gao (5232481)

April 28, 2023

1 Estimation of directions and frequencies

Signal processing is the theory of converting acquired sensor measurements into useful data. The derivation process consists of deriving the data model (simplifies the physics to capture the essentials) and extracting the information by methods and algorithms. In this section, firstly the signal model is derived, then ESPRIT algorithm and its extension joint diagonalization technique are applied to estimation (joint) directions and frequencies.

1.1 Signal Model

1.1.1 Data generator

In this project, M number of multiple antennas sampling in space are considered to receive d number of source signals. This data generator function aims to generate the received signal matrix $\mathbf{X}$ with the size $M \times N$. Table 1 shows all symbols and parameters used in this data generator, and corresponding explanations.

Table 1: Nomenclature

M	number of uniform linear array antennas
N	number of samples
Δ	$\Delta = D/\lambda$: spacing in wavelengths
θ	$\theta = [\theta_1, ..., \theta_d]^T$: directions of the sources in degrees $-90 \leq \theta_d \leq 90$
f	$f = [f_1, ..., f_d]$: normalized frequencies of the sources $0 \leq f_i \leq 1$
d	number of complex sinusoidal signals
SNR	signal-to-noise ratio per source
Noise	temporally and spatially white, complex Gaussian
$\mathbf{A}$	array response matrix: $M \times d$
$\mathbf{S}$	source matrix: $d \times N$
$\mathbf{X}$	received signal matrix: $M \times N$

Equation 1 shows the signal model. It is a linear model and output of antennas are stacked into a vector $\boldsymbol{x}(t)$.

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) \tag{1}$$

As there are M number of antennas and N numbers of samples. Equation 1 can be rewritten as a matrix form shown in Equation 2, where $\mathbf{X}$ has the size $M \times N$, $\mathbf{A}$ has the size $M \times d$, $\mathbf{S}$ has the size $d \times N$, and $\mathbf{N}$ has the size $M \times N$.

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{N} \tag{2}$$

In the noiseless case, the received signals stacked in a vector $\boldsymbol{x}(t)$ is written as Equation 3.

$$\boldsymbol{x}(t) = \begin{bmatrix} x_0(t) \\ x_1(t) \\ ... \\ x_{M-1}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{j\omega_0\tau_1} \\ ... \\ e^{j\omega_0\tau_{M-1}} \end{bmatrix} a_0(\theta)s(t) \tag{3}$$

For a uniform linear array, all antennas have the same spacing D, thus the delay $\phi_m = -\omega_0 \tau_m = -2\pi m \Delta sin(\theta)$. Then $\boldsymbol{x}(t)$ can be represented as Equation 4, where $\boldsymbol{a}(\theta)$ is called the array response vector.

$$\boldsymbol{x}(t) = \boldsymbol{a}(\theta)s(t) = \begin{bmatrix} 1 \\ e^{j\phi_1} \\ ... \\ e^{j\phi_{M-1}} \end{bmatrix} a_0(\theta)s(t) = \begin{bmatrix} 1 \\ e^{j2\pi\Delta sin(\theta)} \\ ... \\ e^{j2\pi(M-1)\Delta sin(\theta)} \end{bmatrix} a_0(\theta)s(t) \tag{4}$$

For the source $s(t)$, the dth source is a complex sinusoid with normalized frequency f_d and given by equation 5.

$$\boldsymbol{s}_{d,n} = exp(j2\pi f_d n) \tag{5}$$

For the noise, it is complex white Gaussian and scaled to obtain to desired SNR per source.

1.1.2 Singular Value Plots

Now, $d = 2$ sources, $M = 5$, $N = 20$, $\Theta = [-20, 30]^T$, $f = [0.1, 0.3]^T$, and $SNR = 20dB$. Singular values of $\boldsymbol{X}$ are plotted and compared by changing the numbers of antennas, the number of samples, the angle difference, and the frequency difference.

Number of samples N and antenna M doubles
Figure 1 shows the comparison of singular values between N samples and $2N$ samples and between M antennas and $2M$ antennas. When N doubles, all singular values become bigger. We assume a_1 as the singular value of the first source, a_2 as the singular value of the second source, and b as the mean singular value of the noise. After N doubles, the gap between the source and the noise gets much bigger as $gap = \frac{a_{1,2}}{b}$. Therefore, it is much easier to distinguish two sources. When M doubles, the number of singular values doubles and all value of singular values become bigger. As the source singular values become much larger and the gap difference between the sources and noises get bigger, distinguishing the sources becomes easier.

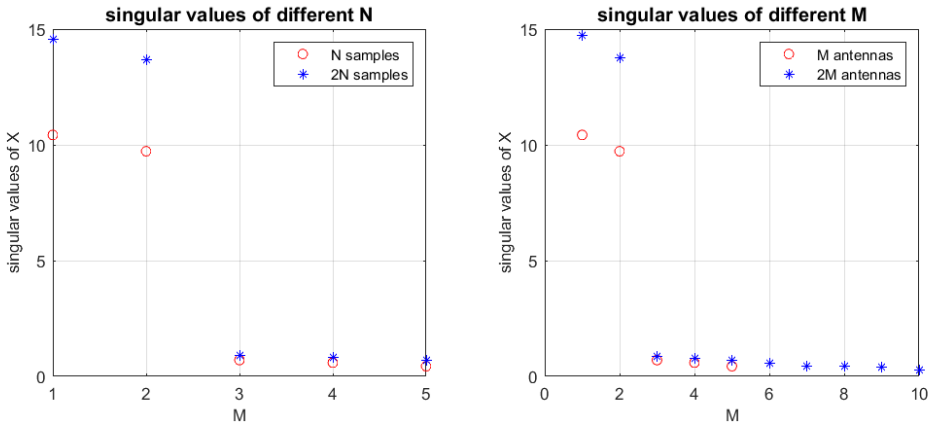

Figure 1: Effect on singular values of doubling the number of M and N

Smaller angles θ and frequency f difference between the sources
Figure 2 shows the comparison of singular values between different angles $\theta = [-20, 30]$, $\theta = [-5, 5]$ and singular values between different frequencies $f = [0.1, 0.3]$, $f = [0.1, 0.12]$. When $\theta = [-20, 30]$, there are two principle singular values which can be distinguished among all values as the gap between these values and noise values are big. For example, all noise singular values fall below 1 but the first two singular values are both above 9. When $\theta = [-5, 5]$, the first singular value gets bigger and the second singular value get smaller. The gap between the second singular value and the noise singular value becomes much smaller. Therefore, it is hard to distinguish the second as the source or the noise and the threshold for dividing the

source and noise will significantly affect our judgement. Because the angle difference is so small, it may be confusing whether there is one source or two sources. The same analysis can be carried on the effect of smaller frequency differences. When $f = [0.1, 0.3]$, two principle singular values can be distinguished among all singular values as the gap between these values and noise values are big. However, when the frequency difference becomes small as $f = [0.1, 0.12]$, it is hard to tell if there is one source or two sources as the gap between the second singular value and noise singular values gets much smaller.

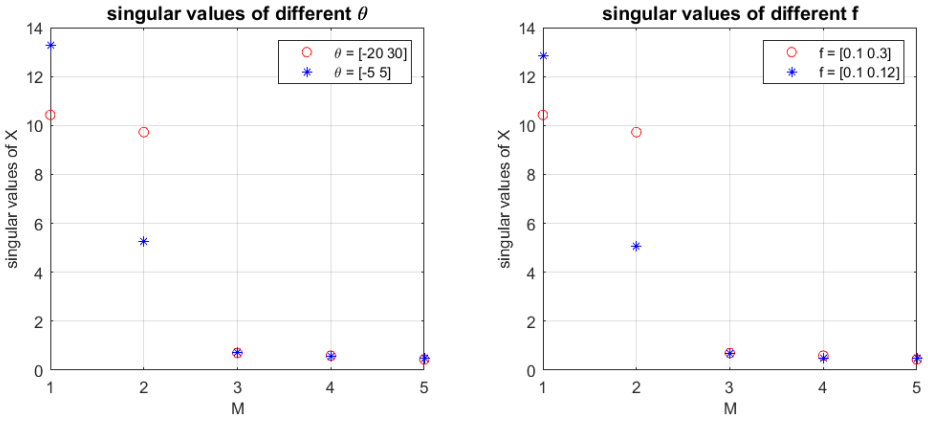

Figure 2: Effect on singular values of decreasing the difference of θ and f

1.2 Estimations of directions

ESPRIT algorithm is used to estimate the angles of arrival. This algorithm does not require a search or calibration data, but assumes a special array configuration that allows to solve for the direction-of-arrival by solving an eigenvalue problem. All signals are assumed narrowband with respect to the propagation delay across the array, so that the delay can be translated to a phase shift. There is a shift-invariance property for uniform linear array as Equation 6.

$$a_x := \begin{bmatrix} 1 \\ \phi \\ \vdots \\ \phi^{M-2} \end{bmatrix}, \quad a_y := \begin{bmatrix} \phi \\ \phi^2 \\ \vdots \\ \phi^{M-1} \end{bmatrix}, \quad \text{so that} \quad a_y = a_x\phi \tag{6}$$

We group the first and last $M-1$ antennas as Equation 7. Due to shift-invariance, the data model is Equation 8 Equation 9:

$$\boldsymbol{x}(t) = \begin{bmatrix} z_1(t) \\ \vdots \\ z_{M-1}(t) \end{bmatrix}, \quad \boldsymbol{y}(t) = \begin{bmatrix} z_2(t) \\ \vdots \\ z_M(t) \end{bmatrix} \tag{7}$$

$$\boldsymbol{X} = \boldsymbol{AS} \tag{8}$$

$$\boldsymbol{Y} = \boldsymbol{A\Theta S} \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(\alpha_1) & \cdots & \mathbf{a}(\alpha_d) \end{bmatrix}, \quad \boldsymbol{\Theta} = \begin{bmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_d \end{bmatrix}, \quad \theta_i = e^{j2\pi\Delta\sin(\alpha_i)}$$

Given a data matrix $\boldsymbol{X}$ and the number of sources d. ESPRIT algorithm follows these steps:
Step 1: Stack the data from $\boldsymbol{X}$ and $\boldsymbol{Y}$ into a single matrix $\boldsymbol{Z}$. $\boldsymbol{Z}$ has size of $2M \times N$:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \mathbf{A}_z\mathbf{S}, \quad \mathbf{A}_z = \begin{bmatrix} \mathbf{A} \\ \mathbf{A\Theta} \end{bmatrix} \tag{10}$$

Step 2: The second step is to compute singular value decomposition (SVD) of $\boldsymbol{Z}$:

$$\mathbf{Z} = \hat{\mathbf{U}}_z\hat{\mathbf{\Sigma}}_z\hat{\mathbf{V}}_z^{\mathrm{H}} \tag{11}$$

$\hat{\mathbf{U}}_z$ has the size of $2M \times d$, It spans the column space of $\mathbf{Z}$ and $\mathbf{A}_z$, so that there must exist a $d \times d$ invertible matrix T such that

$$\hat{\mathbf{U}}_z = \mathbf{A}_z\mathbf{T} = \begin{bmatrix} \mathbf{AT} \\ \mathbf{A\Theta T} \end{bmatrix} \tag{12}$$

Step 3: Next, $\hat{\mathbf{U}}_z$ is split into two $M \times d$ matrices:

$$\hat{\mathbf{U}}_z = \begin{bmatrix} \hat{\mathbf{U}}_x \\ \hat{\mathbf{U}}_y \end{bmatrix}, \quad \begin{cases} \hat{\mathbf{U}}_x = \mathbf{AT} \\ \hat{\mathbf{U}}_y = \mathbf{A\Theta T} \end{cases} \tag{13}$$

Step 4: Compute the eigenvalue decomposition of $\hat{\mathbf{U}}_x^{\dagger}\hat{\mathbf{U}}_y$. Since $\mathbf{\Theta}$ is a diagonal matrix, $\mathbf{T}^{-1}\mathbf{\Theta T}$ is recognized as an eigenvalue equation. $\mathbf{T}^{-1}$ contains the eigenvectors and the diagonal entries of $\mathbf{\Theta}$ are the eigenvalues.

$$\hat{\mathbf{U}}_x^{\dagger}\hat{\mathbf{U}}_y = \left(\left(\hat{\mathbf{U}}_x^{\mathrm{H}}\hat{\mathbf{U}}_x\right)^{-1}\hat{\mathbf{U}}_x^{\mathrm{H}}\right)\hat{\mathbf{U}}_y = \left(\left(\mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{AT}\right)^{-1}\mathbf{T}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\right)\mathbf{A\Theta T} = \mathbf{T}^{-1}\mathbf{A}^{\dagger}\mathbf{A\Theta T} = \mathbf{T}^{-1}\mathbf{\Theta T} \tag{14}$$

Step 5: Since $\mathbf{\Theta} = \begin{bmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_d \end{bmatrix}$, DOAs α_i can be computed from each of them by this relation $\theta_i = e^{j2\pi\Delta\sin(\alpha_i)}$. This is ESPRIT algorithm.

1.3 Estimations of frequencies

1.3.1 Frequency estimation using ESPRIT

Since the structure of $\mathbf{S}$ also follows the shift-invariance property, we can simply transpose $\mathbf{X}$ and $\mathbf{Y}$. The next few steps of ESPRIT algorithm are same with direction estimation from step 2 to step 5.

1.4 Joint estimations of directions and frequencies

1.4.1 Construct Khatri-Rao product structure

For narrowband signals with bandwidth less than $1/T$, take samples with a normalized period $T = 1$ to satisfy the Nyquist rate. Meanwhile the sampling rate is P times larger than the data rate and is also set as 1 in this project. Thus the modulation frequencies f_i of the $i-$th source are bounded in $-\frac{1}{2} \leq f_i < \frac{1}{2}$. Without multipath, the data model at the receiver is

$$\mathbf{x}(t) = \sum_{1}^{d} \mathbf{a}\left(\theta_i\right)\beta_i e^{j\frac{2\pi}{P}f_i t}s_i(t) \tag{15}$$

which can be rewrite as a matrix form

$$\mathbf{x}(t) = \mathbf{A}_{\theta}\mathbf{B}\Phi^t\mathbf{s}(t) \tag{16}$$

where

$$\Phi = \begin{bmatrix} \phi_1 & & 0 \\ & \ddots & \\ 0 & & \phi_d \end{bmatrix}, \quad \phi_i = e^{j\frac{2\pi}{P}f_i}. \tag{17}$$

With a factor m to smooth the data in time series and N sampled series, the received data matrix $\mathbf{X}$ of size $mM \times (N-m)$ is

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(0) & \mathbf{x}(1) & \cdots & \mathbf{x}(N-m-1) \\ \mathbf{x}(1) & \mathbf{x}(2) & \cdots & \mathbf{x}(N-m) \\ \vdots & \vdots & & \vdots \\ \mathbf{x}(m-1) & \mathbf{x}(m) & \cdots & \mathbf{x}(N) \end{bmatrix} \tag{18}$$

Substitute Equation 16 into the matrix,

$$\mathbf{X} = \begin{bmatrix} \mathbf{A}_\theta \mathbf{B}\mathbf{s}(0) & \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}\mathbf{s}(1) & \cdots \\ \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}\mathbf{s}(1) & \mathbf{A}_\theta \mathbf{B}\Phi^2\mathbf{s}(2) & \cdots \\ \vdots & \vdots & \\ \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^{m-1}\mathbf{s}(m-1) & \mathbf{A}_\theta \mathbf{B}\mathbf{\Phi}^{m}\mathbf{s}(m) & \cdots . \end{bmatrix} \tag{19}$$

With assumption

$$\mathrm{s}(t) \approx \mathrm{s}\left(t + \frac{1}{P}\right) \approx \cdots \approx \mathrm{s}\left(t + \frac{m-1}{P}\right)$$

the data model $\mathbf{X}$ can be decomposed as

$$\begin{aligned} \mathbf{X} &\approx \begin{bmatrix} \mathbf{A}_\theta \\ \mathbf{A}_\theta \Phi \\ \vdots \\ \mathbf{A}_\theta \Phi^{m-1} \end{bmatrix} \mathbf{B} \begin{bmatrix} \mathbf{s}_0 & \Phi^P \mathbf{s}_1 & \cdots & \Phi^{(N-1)P} \mathbf{s}_{N-1} \end{bmatrix} \\ &= (\mathbf{F}_\phi \circ \mathbf{A}_\theta)\, \mathbf{B}\, (\mathbf{F'}_\phi \odot \mathbf{S}) \end{aligned} \tag{20}$$

where $\mathbf{F}_\phi = [\mathbf{f}(\phi_1), \cdots, \mathbf{f}(\phi_d)]$, and

$$\mathbf{f}(\phi) = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \end{bmatrix}, \quad \phi := e^{j\frac{2\pi}{P} f} \tag{21}$$

1.4.2 Joint diagonalization problem

As same as classical ESPRIT algorithm, compute the SVD of $\mathbf{X}$, $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$. As the rank of $\mathbf{X}$ equals to d, $\mathbf{U}$ has d columns. Thus there exists $d \times d$ matrix $\mathbf{T}$ resulting to $\mathbf{U} = \mathbf{A}\mathbf{T}^{-1}$. Then, to estimate $\mathbf{\Phi}$ and $\mathbf{\Theta}$, two types of selection matrices are defined as[1]

$$\begin{cases} \boldsymbol{J}^x(\phi) := \begin{bmatrix} \boldsymbol{I}_{m-1} & \mathbf{0}_1 \end{bmatrix} \otimes \boldsymbol{I}_M \\ \boldsymbol{J}^y(\phi) := \begin{bmatrix} \mathbf{0}_1 & I_{m-1} \end{bmatrix} \otimes \boldsymbol{I}_M \\ \boldsymbol{J}^x(\theta) := \boldsymbol{I}_m \otimes \begin{bmatrix} \boldsymbol{I}_{M-1} & \mathbf{0}_1 \end{bmatrix} \\ \boldsymbol{J}^y(\theta) := \boldsymbol{I}_m \otimes \begin{bmatrix} \mathbf{0}_1 & \boldsymbol{I}_{M-1} \end{bmatrix}. \end{cases} \tag{22}$$

For $\boldsymbol{\Phi}$, the submatrices contain the first and the last $M(m-1)$ rows of $\mathbf{U}$, respectively, while for $\boldsymbol{\Theta}$ with m blocks, the first and last $M-1$ rows of each block are selected:

$$\begin{cases} \boldsymbol{U}_{x,\phi} = \boldsymbol{J}^x(\phi)\boldsymbol{U} \\ \boldsymbol{U}_{y,\phi} = \boldsymbol{J}^y(\phi)\boldsymbol{U} \end{cases} \quad \begin{cases} \boldsymbol{U}_{x,\theta} = \boldsymbol{J}^x(\theta)\boldsymbol{U} \\ \boldsymbol{U}_{y,\theta} = \boldsymbol{J}^y(\theta)\boldsymbol{U}. \end{cases} \tag{23}$$

Since these matrices have structures

$$\begin{cases} U_{x,\phi} = \boldsymbol{A}'\boldsymbol{T}^{-1} \\ \boldsymbol{U}_{y,\phi} = \boldsymbol{A}'\boldsymbol{\Phi}\boldsymbol{T}^{-1} \end{cases} \quad \begin{cases} U_{x,\theta} = \boldsymbol{A}''\boldsymbol{T}^{-1} \\ U_{y,\theta} = \boldsymbol{A}''\Theta^{-1}, \end{cases} \tag{24}$$

the joint diagonalization problem is constructed:

$$\begin{aligned} \boldsymbol{U}_{x,\phi}^{\dagger}\boldsymbol{U}_{y,\phi} &= \boldsymbol{T}\boldsymbol{\Phi}\boldsymbol{T}^{-1} \\ \boldsymbol{U}_{x,\theta}^{\dagger}\boldsymbol{U}_{y,\theta} &= \boldsymbol{T}\boldsymbol{\Theta}\boldsymbol{T}^{-1}. \end{aligned} \tag{25}$$

$\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ can be obtained by solving this diagonalization problem. Thus f and θ can also be solved out.

1.5 Comparison

1.5.1 Evaluation of three algorithms

To evaluate three algorithms, mean values and standard deviations of the estimated angles and frequencies are calculated on over 1000 tests with independent complex Gaussian noise. the SNR of input signals are selected as [0,4,8,...,20dB]. The computation results are showed as follows.

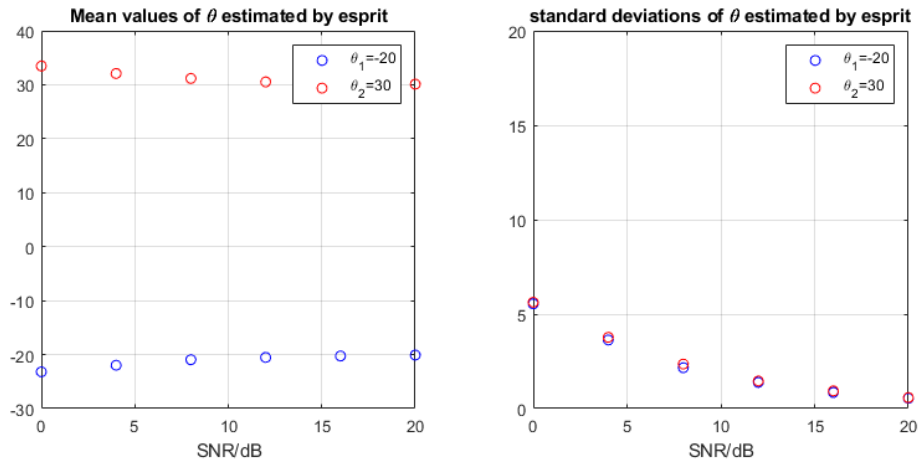

Figure 3: Mean values and standard deviations of the estimated angles $\boldsymbol{\theta}$ individually

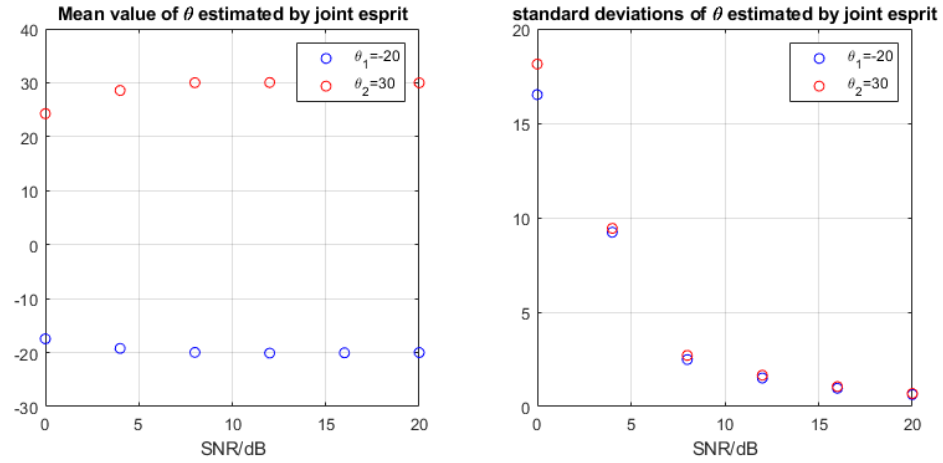

Figure 4: Mean values and standard deviations of the estimated angles $\boldsymbol{\theta}$ jointly

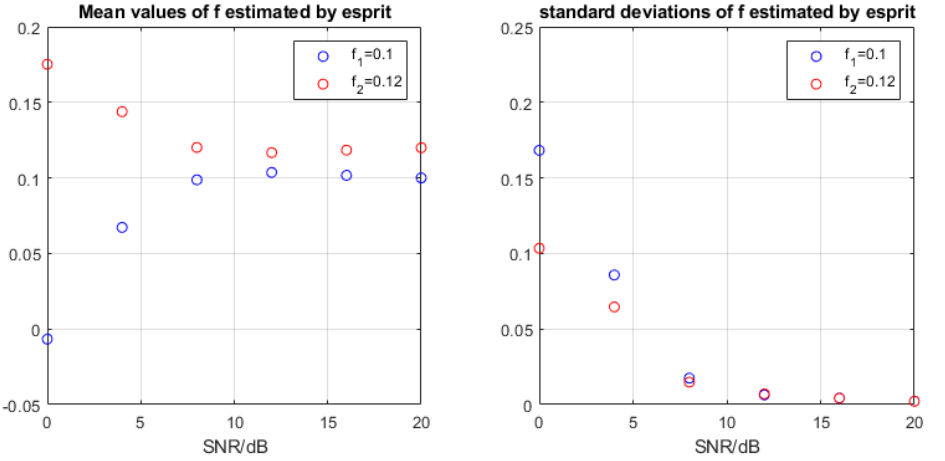

Figure 5: Mean values and standard deviations of the estimated frequencies $\boldsymbol{f}$ individually

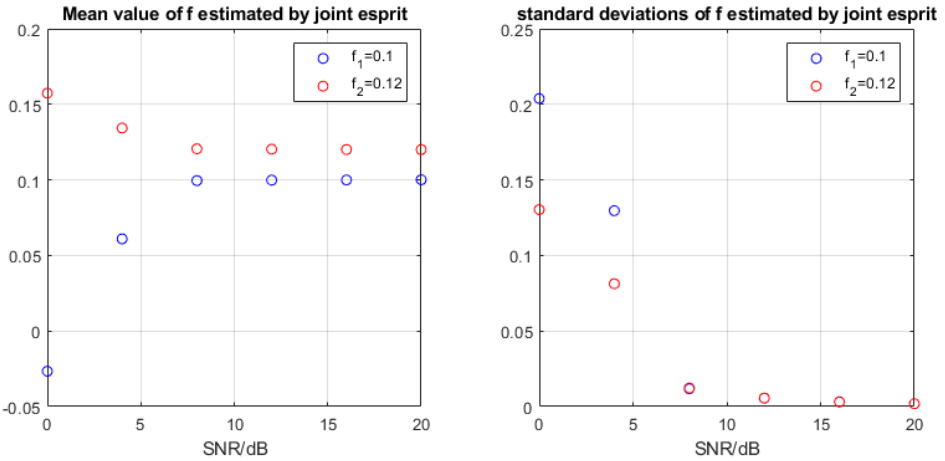

Figure 6: Mean values and standard deviations of the estimated frequencies $\boldsymbol{f}$ jointly

From Figure 3 to Figure 6, it can be seen that as SNR increases, the estimated values get closer to the true values and thus standard deviations decrease. Meanwhile, the individual estimations always perform better

than joint estimations at low-level SNR. Once the SNR is over 5dB, joint estimation is comparable to the performance of individual estimation on $\boldsymbol{\theta}$.

1.5.2 Beamformer calculation

Zero-Forcing solution based on noisy model aims to minimize the model fitting error:

$$\min \|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2.$$

With $\mathbf{A}$ known, there are

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2 \quad \Rightarrow \quad \hat{\mathbf{S}} = \mathbf{A}^{\dagger}\mathbf{X}$$

so that $\mathbf{W}^H = \mathbf{A}^{\dagger}$.
With $\mathbf{S}$ known, the problem can be written and solved as

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AS}\|_{\mathrm{F}}^2 \quad \Rightarrow \quad \hat{\mathbf{A}} = \mathbf{X}\mathbf{S}^{\dagger} = \mathbf{X}\mathbf{S}^{\mathrm{H}}\left(\mathbf{S}\mathbf{S}^{\mathrm{H}}\right)^{-1}$$

so that $\mathbf{W}^{\mathrm{H}} = \mathbf{X}\mathbf{S}^{\dagger}$.

1.5.3 Spatial-responses comparison

The zero-forcing beamformers are calculated in both conditions without noise and with noise. The spatial responses of those beamformers $y(\theta) = \left|\mathbf{w}^H\mathbf{a}(\theta)\right|$ for $-90° \leq \theta \leq 90°$ are visualized as follows.

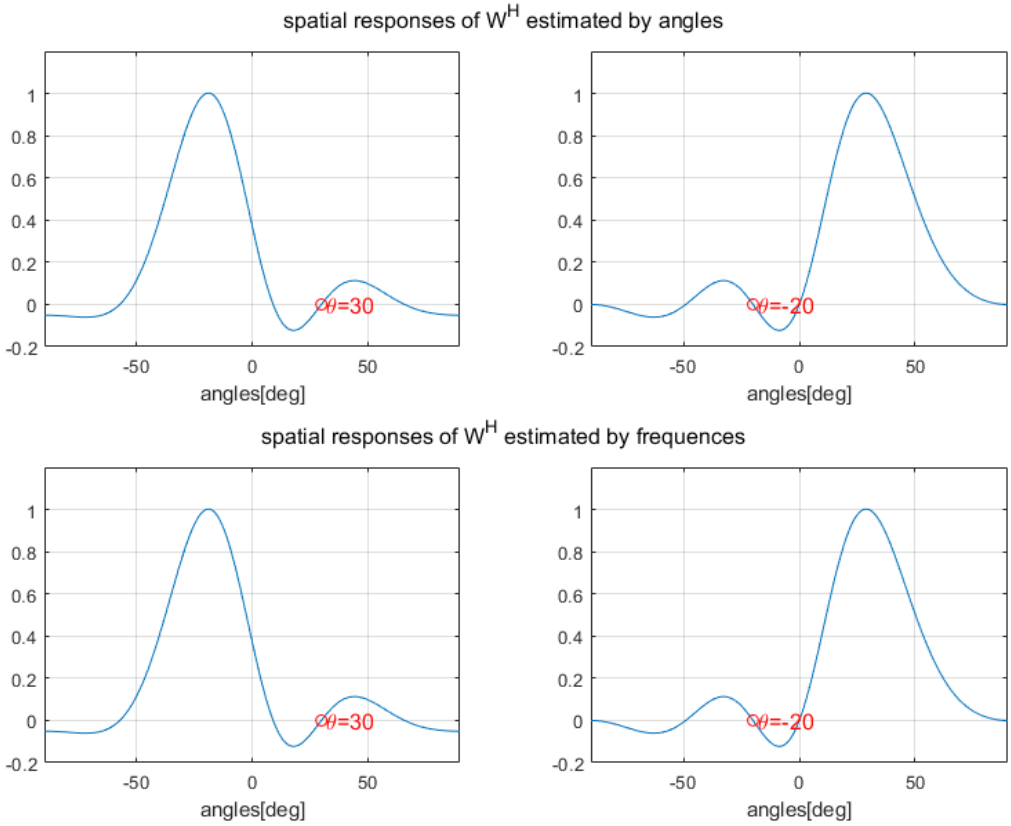

Figure 7: spatial responses of W^H estimated by angles and frequencies in noise-free condition

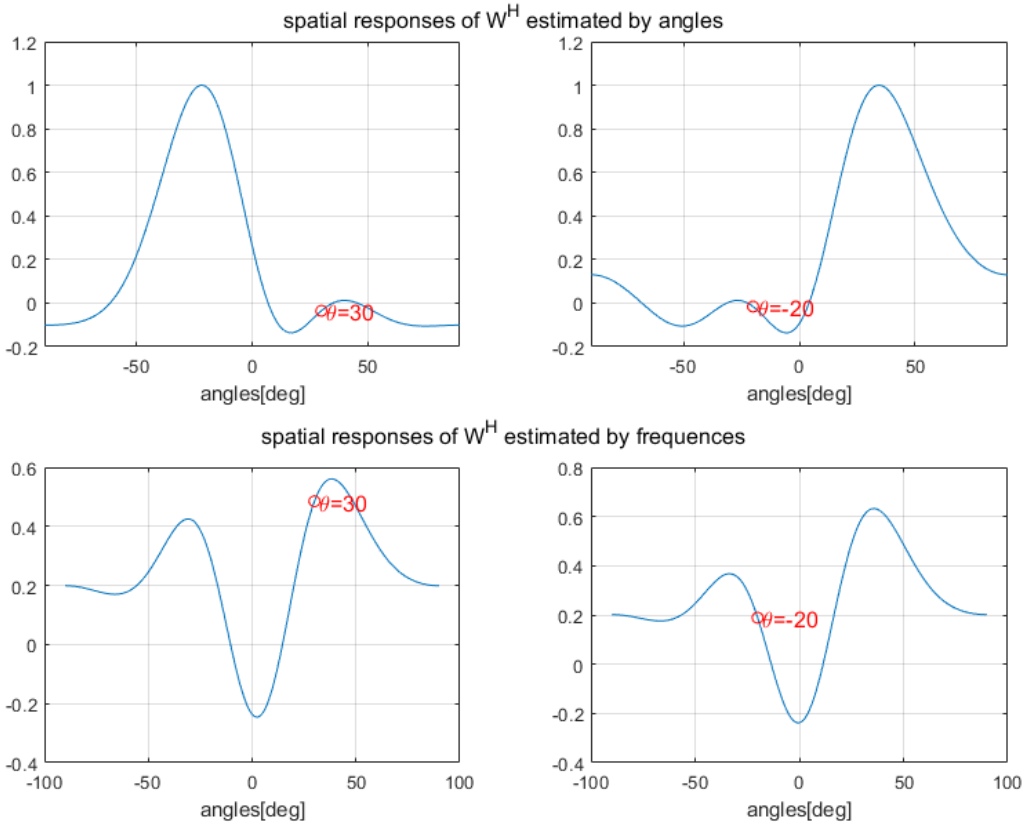

Figure 8: spatial responses of W^H estimated by angles and frequencies with SNR=10

Figure 7 shows that when there is no noise, the interference is cancelled at $\theta = -20$ and $\theta = 30$ for both beamformers. In noisy condition as Figure 8, the beamformer estimated by angles can still cancel the interference at desired angles, however, for the beamformer estimated by frequencies, it does not cancel all interference, therefore, the output source estimated by frequency is not unbiased.

2 Channel Equalization

In this part, a single source (a data symbol sequence) is transmitted towards a single receive antenna over a specific channel. The goal is to reconstruct the transmitted data symbols by linear equalization.

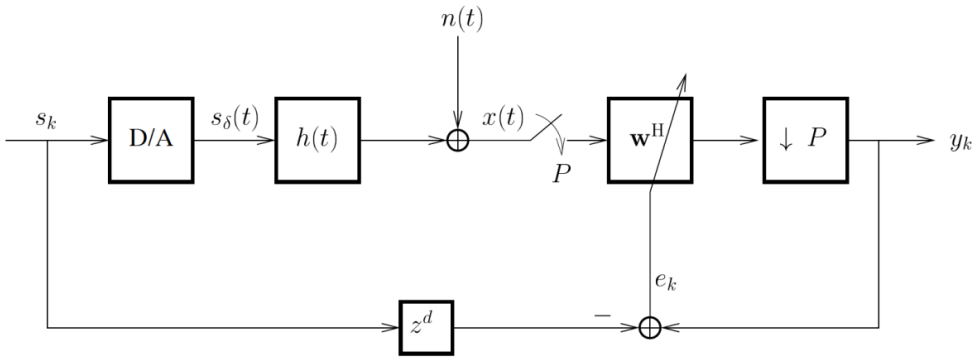

Figure 9: channel equalization

As Figure 9 shows, before a digital bit sequence can be transmitted over a channel, it has to be transformed

into an analog signal (D/A) in continuous time and then gets convoluted with $h(t)$. During the transmission, there is noise added into the data symbols. The linear equalizer can be written as a vector $\boldsymbol{w}$ which combines the rows of $\boldsymbol{X}$ to generate an output $\boldsymbol{y} = \boldsymbol{w}^{\boldsymbol{H}}\boldsymbol{X}$.

2.1 Signal model

2.1.1 Transmitted sequence x construction

All signals are assumed under wideband, which convolutions by pulse shape functions and channel propagation delays play an important role. Here local scattering is ignored and there are r rays bouncing off remote objects. As an extension of the narrowband model in Equation 5, the received parametric signal model can be written as Equation 26:

$$x(t) = \sum_k h(t-k)s_k + n(t), \quad \mathbf{h}(t) = \left[\sum_{i=1}^{r} \mathbf{a}(\theta_i)\,\beta_i g(t-\tau_i)\right], \tag{26}$$

where $\mathbf{x}(t)$ is a vector consisting of the M antenna outputs, $\mathbf{a}(\theta)$ is the array response vector, and the impulse response $g(t)$ collects all temporal aspects. s_k is a sequence chosen from quadrature phase shift keying (QPSK) constellation. We assume the values $s_k \in \{\pm 1/\sqrt{2} \pm j/\sqrt{2}\}$. The noise $n(t)$ is modeled as complex zero-mean Gaussian with standard deviation σ.

$$\mathbf{x} = \mathbf{H}\mathbf{s} \Leftrightarrow \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & & & \mathbf{0} \\ h[1] & h[0] & & \\ h[2] & h[1] & \ddots & \\ \vdots & h[2] & \ddots & h[0] \\ h[L-1] & \vdots & \ddots & h[1] \\ & h[L-1] & \ddots & h[2] \\ & & \ddots & \vdots \\ \mathbf{0} & & & h[L-1] \end{bmatrix} \begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s\left[N_s-1\right] \end{bmatrix} \tag{27}$$

Let $h[k]$ be a finite impulse response (FIR) filter. The matrix equation corresponding to a convolution $x[n] = h[n] * s[n] = \sum_{k=0}^{L-1} h[k]s[n-k]$ is Equation 27. As $\hat{\mathbf{s}} = \mathbf{H}^{\dagger}\mathbf{x} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{x}$, $\mathbf{H}$ needs to be a tall matrix to get inverted. If it is not tall, oversampling is considered. It means sampling faster than the symbol rate.

The received signal $x(t)$ is sampled at rate $1/P$, which means the sampling rate is P times larger than the data rate. We stack the data as Equation 28.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}[0] & \cdots & \mathbf{x}[N-1] \end{bmatrix} = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-1) \\ x\left(\frac{1}{P}\right) & x\left(1+\frac{1}{P}\right) & \cdot & \\ \vdots & & & \vdots \\ x\left(\frac{P-1}{P}\right) & & \cdots & x\left(N-1+\frac{P-1}{P}\right) \end{bmatrix} \tag{28}$$

where

$$\mathbf{x}[n] = \begin{bmatrix} x(n) \\ x\left(n+\frac{1}{P}\right) \\ \vdots \\ x\left(n+\frac{P-1}{P}\right) \end{bmatrix},$$

$\mathbf{X}$ has size $P \times N$; its nth column $\mathbf{x}[n]$ contains the P samples taken during the nth symbol period. Based on the FIR assumption, it follows that X has a factorization as Equation 29:

$$\mathbf{X} = \mathbf{HS} \tag{29}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}[0] & \mathbf{h}[1] & \cdots & \mathbf{h}[L-1] \end{bmatrix} = \begin{bmatrix} h(0) & h(1) & \cdots & h(L-1) \\ h\left(\frac{1}{P}\right) & \cdot & & \cdot \\ \vdots & & & \vdots \\ h\left(\frac{P-1}{P}\right) & \cdot & \cdots & h\left(L-\frac{1}{P}\right) \end{bmatrix} \tag{30}$$

$$\mathbf{S} = \left[\begin{array}{ccc|ccc|ccc} s_0 & s_1 & \cdots & s_{L-1} & \cdots & s_{N_s-2} & s_{N_s-1} & & \mathbf{0} \\ & s_0 & \cdots & \cdots & \cdots & \cdots & s_{N_s-2} & s_{N_s-1} & \\ & & \ddots & s_1 & \cdots & \cdots & \cdots & \ddots & \ddots \\ \mathbf{0} & & & s_0 & \cdots & \cdots & s_{N_s-L} & \cdots & s_{N_s-2} \quad s_{N_s-1} \end{array}\right] \tag{31}$$

where $\mathbf{H}$ has the size $P \times L$ and $\mathbf{S}$ has the size $L \times N$. $L = 1$ since $\mathbf{h}$ here is defined as

$$h(t) = \begin{cases} 1 \text{ for } t \in [0, 0.25) \\ -1 \text{ for } t \in [0.25, 0.5) \\ 1 \text{ for } t \in [0.5, 0.75) \\ -1 \text{ for } t \in [0.75, 1). \end{cases}$$

To construct the sequence $\mathbf{x} = [x(0), x(1/P), \cdots, x(N-1/P)]^T$ as required, we can re-arrange Equation 28 as

$$\mathbf{X} = \begin{bmatrix} x[0] \\ \vdots \\ x\left[N-1\right] \end{bmatrix} = \begin{bmatrix} x(0) \\ \vdots \\ x\left(N-1/P\right) \end{bmatrix},$$

and it has the size $PN \times 1$.

2.1.2 Rank of augmented data matrix

Now take $P = 4$, $N = 500$, $\sigma = 0$, and construct the augmented data matrix

$$\mathcal{X} = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-2) \\ x\left(\frac{1}{P}\right) & x\left(1+\frac{1}{P}\right) & \cdots & x\left(N-2+\frac{1}{P}\right) \\ \vdots & \vdots & & \vdots \\ x\left(1+\frac{P-1}{P}\right) & x\left(2+\frac{P-1}{P}\right) & \cdots & x\left(N-1+\frac{P-1}{P}\right) \end{bmatrix} : 2P \times (N-1). \tag{32}$$

With $\mathbf{x}[n] = [x(n), x(n+\frac{1}{P}), \cdots, x(n+\frac{P-1}{P})]$, the data matrix stacked 2 samples can be written as

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}(0) & \mathbf{x}(1) & \cdots & \mathbf{x}(N-2) \\ \mathbf{x}(1) & \mathbf{x}(2) & \cdots & \mathbf{x}(N-1). \end{bmatrix} \tag{33}$$

Since the channel responds $h(t)$ are all constant values, while they correspond to a same signal, the selection of P has no impact on the rank of data matrix. In other words, for one stack, the rows are all linear dependent. What decides the rank is the number of stacks. Therefore, in the present situation, the rank of data matrix is 2.

2.2 Zero-forcing and Wiener equalizer

Take $P = 4$, $N = 500$, $\sigma = 0.5$, Construct a data matrix with the same structure as 2.1.2 showed. Factorize it as $\mathcal{X} = \mathcal{H}\mathcal{S}$ where

$$\mathcal{H} = \begin{bmatrix} \mathbf{0} & \mathbf{H} \\ \mathbf{H} & \mathbf{0} \end{bmatrix}.$$

- **Zero-forcing receiver**
  When intersymbol interference (ISI) is not viewed as noise, the minimization is

  $$\min \|\mathcal{X} - \mathcal{H}\mathcal{S}\|_{\mathrm{F}}^2.$$

  As there is only a shift in the output sequences, we can select one as the output. Thus the ZF receiver is

  $$\mathbf{w} = \mathcal{H}^{\dagger}\mathbf{e} = (\mathcal{H}\mathcal{H}^{\mathrm{H}})^{\dagger}\mathbf{h}$$

- **Wiener receiver**
  According to output error minimization,

  $$\mathbf{W}^{\mathrm{H}} = \arg\min_{\mathbf{W}} \left\|\mathbf{W}^{\mathrm{H}}\mathcal{X} - \mathcal{S}\right\|_{\mathrm{F}}^2 = \mathcal{S}\mathcal{X}^{\dagger} = \mathbf{S}\mathcal{X}^{\mathrm{H}}\left(\mathcal{X}\mathcal{X}^{\mathrm{H}}\right)^{-1} = \hat{\mathbf{R}}_{xs}^{\mathrm{H}}\hat{\mathbf{R}}_{x}^{-1}$$

  thus $\mathbf{w} = \hat{\mathbf{R}}_{x}^{-1}\hat{\mathbf{R}}_{xs}\mathbf{e} = \left(\mathcal{H}\mathcal{H}^{\mathrm{H}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{h}$.

The estimated symbols of the two equalizers are showed in Figure 10.If take $P = 8$, other configurations are the same as before, the results are showed in Figure 11.
As Figure 10 and Figure 11 shows, there is small difference between two equalizers. From their expressions, it can be seen that the only difference is a term of noise power matrix. Moreover, since there are only two series of signals are stacked, there is only one different point between outputs of different delays. In usual, we would like to take the middle samples. What is more, the estimation by taking $P = 8$ is more concentrated than by taking $P = 4$. It clearly shows that as sampling rate increases, the performances of the beamformers get better.

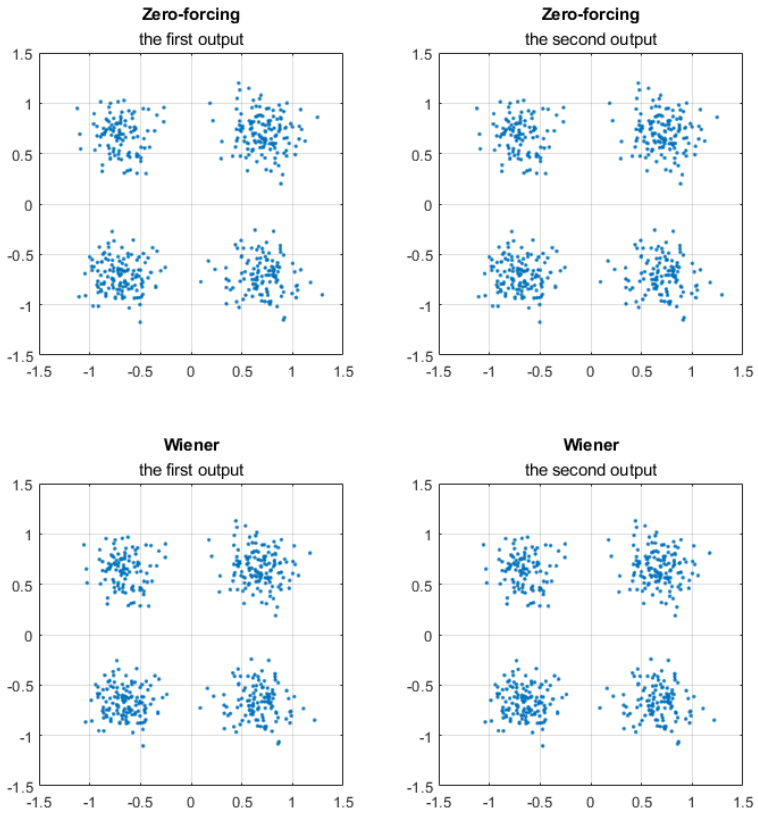

Figure 10: The estimated symbols of ZF and Wiener equalizers in the complex plane ($P = 4$)

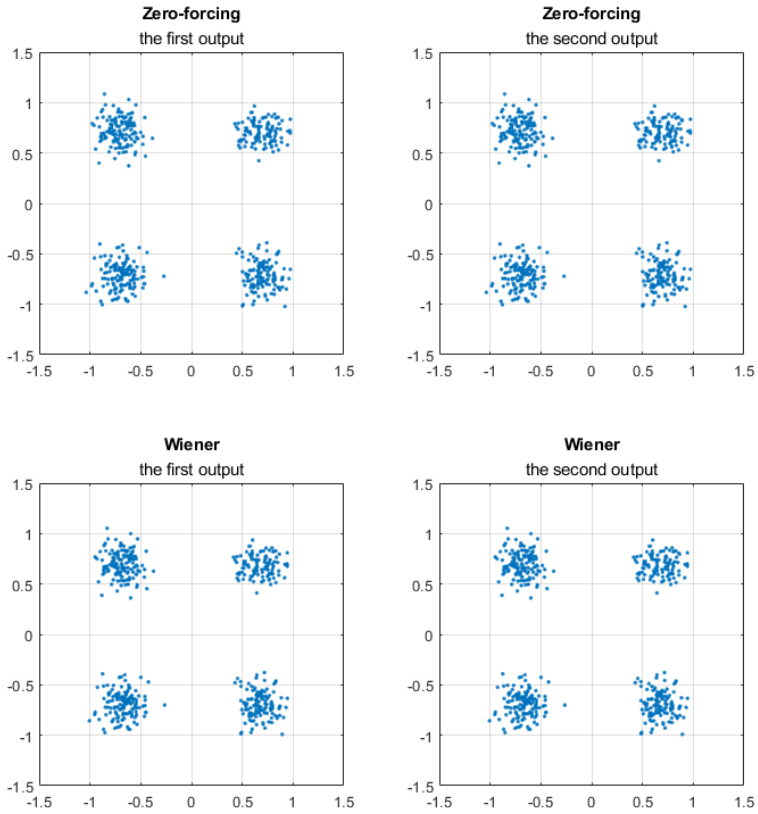

Figure 11: The estimated symbols of ZF and Wiener equalizers in the complex plane ($P = 8$)

A Estimation & Comparison of Directions and frequencies

```
clear all;
% generate signal model
% M=5;
% N=20;
% Delta=1/2;
% Theta=[-20,30];
% F=[0.1,0.12]';
SNR=20;

% [X,A,S]=gendata(M,N,Delta,Theta,F,SNR,0);
%
% [¬,Singular,¬] = svd(X,'econ');
% % figure(1)
% % stem(diag(Singular),'o')
%
% %% estimation of directions
% d=2;
% theta=esprit(X,d);
%
% %% estimation of frequency
% f=espritfreq(X,d);
%
% %% joint estimation
% [theta_j,f_j]=joint(X,2,3);

%% comparison
d=2;
M=3;
N=20;
Delta=1/2;
Theta=[-20,30];
F=[0.1,0.12]';
t=1000;

% plot mean values and deviations
[theta_mean,theta_dev]=theta_pform(M,N,Delta,Theta,F,d,t);
[f_mean,f_dev]=f_pform(M,N,Delta,Theta,F,d,t);
[theta_j_mean,theta_j_dev,f_j_mean,f_j_dev]=theta_j_pform(M,N,Delta,Theta,F,d,t);

% beamformer computing
snrr=10;
[X,A,S]=gendata(M,N,Delta,Theta,F,snrr,1);
MM=(0:M-1)';
theta_hat=esprit(X,d)';
theta_hat=deg2rad(theta_hat);
delay=2.*pi.*Delta.*sin(theta_hat).*MM;
A1_hat=exp(1j.*delay);
WH1=pinv(A1_hat);

NN=(1:N);
f_hat=espritfreq(X,d);
S2_hat=exp(1j.*2.*pi.*f_hat'.*NN);
A2_hat=X*(S2_hat')/(S2_hat*(S2_hat'));
WH2=S2_hat*pinv(X);

theta_var=(-90:1:90);
theta_var2=deg2rad(theta_var);
shangbiao=2.*pi.*Delta.*sin(theta_var2).*MM;
A_theta=exp(1j.*shangbiao);
y_theta1=zeros(1,length(theta_var));
y_theta2=zeros(1,length(theta_var));
y_theta3=zeros(1,length(theta_var));
y_theta4=zeros(1,length(theta_var));

for k=1:length(theta_var2)
    y_theta1(k)=real(det(WH1(1,:)*A_theta(:,k)));
    y_theta2(k)=real(det(WH1(2,:)*A_theta(:,k)));
    y_theta_A=y_theta1+y_theta2;
```

```
end
for k=1:length(theta_var2)
    y_theta3(k)=real(det(WH2(1,:)*A_theta(:,k)));
    y_theta4(k)=real(det(WH2(2,:)*A_theta(:,k)));
    y_theta_S=y_theta3+y_theta4;
end


fig5=figure(5);
sgtitle('spatial responses of W^H estimated by angles')
subplot(1,2,1)
plot(theta_var,y_theta1)
grid on
hold on
axis([-90 90 -0.2 1.2])
xlabel('angles[deg]')
p2=find(theta_var==30);
plot(theta_var(p2),y_theta1(p2),'o','color','r')
text(theta_var(p2)+2,y_theta1(p2),['\theta=',num2str(theta_var(p2))],'color','r','FontSize',12)
hold off
subplot(1,2,2)
plot(theta_var,y_theta2)
hold on
axis([-90 90 -0.2 1.2])
p1=find(theta_var==-20);
plot(theta_var(p1),y_theta2(p1),'o','color','r')
grid on
text(theta_var(p1)+2,y_theta2(p1),['\theta=',num2str(theta_var(p1))],'color','r','FontSize',12)
hold off
xlabel('angles[deg]')
set(fig5, 'Position', [100,100,900,300])

fig6=figure(6);
sgtitle('spatial responses of W^H estimated by frequences')
subplot(1,2,1)
plot(theta_var,y_theta3)
grid on
hold on
xlabel('angles[deg]')
p4=find(theta_var==30);
plot(theta_var(p4),y_theta3(p4),'o','color','r')
text(theta_var(p4)+2,y_theta3(p4),['\theta=',num2str(theta_var(p4))],'color','r','FontSize',12)
hold off
subplot(1,2,2)
plot(theta_var,y_theta4)
hold on
p3=find(theta_var==-20);
plot(theta_var(p3),y_theta4(p3),'o','color','r')
grid on
text(theta_var(p3)+2,y_theta4(p3),['\theta=',num2str(theta_var(p3))],'color','r','FontSize',12)
hold off
xlabel('angles[deg]')
set(fig6, 'Position', [100,100,900,300])

%% plotting Function part

function [theta_mean,theta_dev]=theta_pform(M,N,Delta,Theta,F,d,t)

theta_mean=zeros(6,d);
theta_dev=zeros(6,d);
n=1;
for snr=0:4:20
    theta=zeros(t,d);
    for k=1:t
        [X,¬,¬]=gendata(M,N,Delta,Theta,F,snr,1);
        theta(k,:)=esprit(X,d);
    end
    theta_mean(n,:)=mean(theta);
    theta_dev(n,:)=std(theta);

    fig1=figure(1);
    subplot(1,2,1)
```

```
141      plot(snr,theta_mean(n,1),'bo')
142      hold on
143      grid on
144      plot(snr,theta_mean(n,2),'ro')
145      legend('\theta_1=-20','\theta_2=30')
146      title('Mean values of \theta estimated by esprit')
147      xlabel('SNR/dB')
148      subplot(1,2,2)
149      plot(snr,theta_dev(n,1),'bo')
150      hold on
151      grid on
152      plot(snr,theta_dev(n,2),'ro')
153      legend('\theta_1=-20','\theta_2=30')
154      axis([0 20 0 20])
155      title('standard deviations of \theta estimated by esprit')
156      xlabel('SNR/dB')
157      set(fig1, 'Position', [100,100,900,375])
158      n=n+1;
159  end
160  hold off
161  end
162
163  function [f_mean,f_dev]=f_pform(M,N,Delta,Theta,F,d,t)
164
165  f_mean=zeros(6,d);
166  f_dev=zeros(6,d);
167  n=1;
168  for snr=0:4:20
169      f=zeros(t,d);
170      for k=1:t
171          [X,¬,¬]=gendata(M,N,Delta,Theta,F,snr,1);
172          f(k,:)=espritfreq(X,d);
173      end
174      f_mean(n,:)=mean(f);
175      f_dev(n,:)=std(f);
176
177      fig2=figure(2);
178      subplot(1,2,1)
179      plot(snr,f_mean(n,1),'bo')
180      hold on
181      grid on
182      plot(snr,f_mean(n,2),'ro')
183      legend('f_1=0.1','f_2=0.12')
184      title('Mean values of f estimated by esprit')
185      xlabel('SNR/dB')
186      subplot(1,2,2)
187      plot(snr,f_dev(n,1),'bo')
188      hold on
189      grid on
190      plot(snr,f_dev(n,2),'ro')
191      legend('f_1=0.1','f_2=0.12')
192      axis([0 20 0 0.25])
193      title('standard deviations of f estimated by esprit')
194      xlabel('SNR/dB')
195      set(fig2, 'Position', [100,100,900,375])
196      n=n+1;
197  end
198  hold off
199  end
200
201  function [theta_mean,theta_dev,f_mean,f_dev]=theta_j_pform(M,N,Delta,Theta,F,d,t)
202
203  theta_mean=zeros(6,d);
204  theta_dev=zeros(6,d);
205  f_mean=zeros(6,d);
206  f_dev=zeros(6,d);
207  n=1;
208  for snr=0:4:20
209      theta=zeros(t,d);
210      f=zeros(t,d);
211      for k=1:t
212          [X,¬,¬]=gendata(M,N,Delta,Theta,F,snr,1);
```

```
213             [theta(k,:),f(k,:)]=joint(X,d,3);
214         end
215         theta_mean(n,:)=mean(theta);
216         theta_dev(n,:)=std(theta);
217         f_mean(n,:)=mean(f);
218         f_dev(n,:)=std(f);
219 
220         fig3=figure(3);
221         subplot(1,2,1)
222         plot(snr,theta_mean(n,1),'bo')
223         hold on
224         grid on
225         plot(snr,theta_mean(n,2),'ro')
226         legend('\theta_1=-20','\theta_2=30')
227         title('Mean value of \theta estimated by joint esprit')
228         xlabel('SNR/dB')
229         subplot(1,2,2)
230         plot(snr,theta_dev(n,1),'bo')
231         hold on
232         grid on
233         plot(snr,theta_dev(n,2),'ro')
234         legend('\theta_1=-20','\theta_2=30')
235         title('standard deviations of \theta estimated by joint esprit')
236         xlabel('SNR/dB')
237         set(fig3, 'Position', [100,100,900,375])
238 
239         fig4=figure(4);
240         subplot(1,2,1)
241         plot(snr,f_mean(n,1),'bo')
242         hold on
243         grid on
244         plot(snr,f_mean(n,2),'ro')
245         legend('f_1=0.1','f_2=0.12')
246         title('Mean value of f estimated by joint esprit')
247         xlabel('SNR/dB')
248         subplot(1,2,2)
249         plot(snr,f_dev(n,1),'bo')
250         hold on
251         grid on
252         plot(snr,f_dev(n,2),'ro')
253         legend('f_1=0.1','f_2=0.12')
254         axis([0 20 0 0.25])
255         title('standard deviations of f estimated by joint esprit')
256         xlabel('SNR/dB')
257         set(fig4, 'Position', [100,100,900,375])
258         n=n+1;
259     end
260     hold off
261     end
```

B gendata function

```
1  function [X,A,S]=gendata(M,N,Delta,theta,f,SNR,n)
2  MM=(0:M-1)';
3  theta=deg2rad(theta);
4  delay=2.*pi.*Delta.*sin(theta).*MM;
5  k=(1:N);
6  
7  A=exp(1j.*delay);
8  S=exp(1j.*2.*pi.*f.*k);
9  if n==0
10     X=A*S;
11 else if n==1
12         X=awgn(A*S,SNR,'measured');
13     end
14 end
15 end
```

C esprit function

```
function theta=esprit(X,d)
M=size(X,1);
XX=X(1:M-1,:);
YY=X(2:M,:);
Z=[XX;YY];%2(M-1)*N
[Uz,¬,¬]=svd(Z);
Uz=Uz(:,1:d);
Ux=Uz(1:M-1,:);
Uy=Uz(M:2*M-2,:);
Theta=eig(pinv(Ux)*Uy);
theta=single(real(asind(log(Theta)./(1j*pi))));
theta=sort(theta);
end
```

D espritfreq function

```
function f=espritfreq(X,d)
M=size(X,2);
X=X';
XX=X(1:M-1,:);
YY=X(2:M,:);
Z=[XX;YY];%2(N-1)*M
[Uz,¬,¬]=svd(Z);
Uz=Uz(:,1:d);
Ux=Uz(1:M-1,:);
Uy=Uz(M:2*M-2,:);
Theta=eig(pinv(Ux)*Uy);
f=-(real(log(Theta)./1j)./(2*pi))';
f=sort(f);
end
```

E joint function

```
function [theta,f]=joint(X,d,m)

M=size(X,1);
N=size(X,2);
Z=zeros(m*M,N-m+1);
for k=0:(m-1)
    for p=1:N-m+1
        Z(k*M+1:(k+1)*M,p)=X(:,p+k);
    end
end
[Uz,¬,¬]=svd(Z);
Uz=Uz(:,1:d);


Im1=[eye(m-1),zeros(m-1,1)];%[Im-1 0]
Im2=[zeros(m-1,1),eye(m-1)];%[0 Im-1]
IM1=[eye(M-1),zeros(M-1,1)];%[IM-1 0]
IM2=[zeros(M-1,1),eye(M-1)];%[0 IM-1]

Jx_phi=kron(Im1,eye(M));
Jy_phi=kron(Im2,eye(M));
Jx_theta=kron(eye(m),IM1);
Jy_theta=kron(eye(m),IM2);

Ux_phi=Jx_phi*Uz;
Uy_phi=Jy_phi*Uz;
```

```
Ux_theta=Jx_theta*Uz;
Uy_theta=Jy_theta*Uz;

MM_len=d;
MM=zeros(MM_len,MM_len,2);
MM(:,:,1)=pinv(Ux_phi)*Uy_phi;
MM(:,:,2)=pinv(Ux_theta)*Uy_theta;

[¬,LL] = acdc(MM);

theta=diag(LL(:,:,2));
theta=int8(real(asind(log(theta)./(1j*pi))));
theta=sort(theta);
f=diag(LL(:,:,1));
f=(real(log(f)./1j)./(2*pi))';
f=sort(f);
```

F Channel Equalization

```
clear all
P=4;
N=500;
sigma=0;
L=1;
Ns=N-L+1;
indx=ceil(4.*rand(1,2000));
sk=[1/sqrt(2)+1j/sqrt(2) 1/sqrt(2)-1j/sqrt(2) -1/sqrt(2)+1j/sqrt(2) -1/sqrt(2)-1j/sqrt(2)];
s=zeros(1,Ns);
for k=1:Ns
    num=indx(k);
    s(k)=sk(num);
end
%%%%%%Ns>L
%%%%%%N=Ns+L-1
[x,¬]=gendata_conv(s,P,N,sigma);
X=zeros(2*P,N-1);
for m=1:2*P
    for n=1:N-1
        X(m,n)=x(P*(n-1)+m);
    end
end

%% ZF
P2=4;
N2=500;
sigma2=0.5;

[x2,H]=gendata_conv(s,P2,N2,sigma2);
X2=zeros(2*P2,N2-1);
for m=1:2*P2
    for n=1:N2-1
        X2(m,n)=x2(P2*(n-1)+m);
    end
end
ze=zeros(size(H,1),1);
Hm=[ze H;H,ze];
e1=zeros(size(Hm,2),1);
e1(1)=1;
e2=zeros(size(Hm,2),1);
e2(2)=1;
h1=Hm*[e1 e2];
W_zf=pinv(Hm*(Hm'))*h1;
W_wi=(Hm*(Hm')+(sigma2^2).*eye(size(Hm,1)))\h1;

fig1=figure(1);
title('P=4')
subplot(2,2,1)
plot(real(S_zf(1,:)),imag(S_zf(1,:)),'.')
```

```
title('Zero-forcing')
subtitle('the first output')
axis([-1.5 1.5 -1.5 1.5])
grid on
subplot(2,2,2)
plot(real(S_zf(2,:)),imag(S_zf(2,:)),'.')
title('Zero-forcing')
subtitle('the second output')
axis([-1.5 1.5 -1.5 1.5])
grid on
subplot(2,2,3)
plot(real(S_wi(1,:)),imag(S_wi(1,:)),'.')
title('Wiener')
subtitle('the first output')
axis([-1.5 1.5 -1.5 1.5])
grid on
subplot(2,2,4)
plot(real(S_wi(2,:)),imag(S_wi(2,:)),'.')
title('Wiener')
subtitle('the second output')
axis([-1.5 1.5 -1.5 1.5])
grid on
set(fig1, 'Position', [100,100,800,800])

figure(2)
plot(real(s),imag(s),'+')
```

G gendata_conv function

```
%%%%%%N=Ns+L-1
function [x,H]=gendata_conv(s,P,N,sigma)
Ns=length(s);
L=1;
%t=linspace(0,1,L*P);
h1=ones(1,L*P/4);
h2=-ones(1,L*P/4);
h=[h1 h2 h1 h2];
H=zeros(P,L);
for m=1:P
    for n=1:L
        H(m,n)=h(P*(n-1)+m);
    end
end
r=[s zeros(1,L-1)];
c=[s(1) zeros(1,L-1)];
S=toeplitz(c,r);
X=H*S;
Noi=sqrt(sigma^2/2)*(randn(P*N,1) +1j*randn(P*N,1));
x=reshape(X,P*N,1)+Noi;
end
```

References

[1] A. N. Lemma, A.-J. van der Veen, and E. F. Deprettere, "Analysis of esprit based joint angle-frequency estimation," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 5. IEEE, 2000, pp. 3053–3056.