Adaptive Signal Processing

Final Project Report

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Adaptive Beamforming for Directional Signal Enhancement

賴柏睿

Lai, Po - Jui

授課教師：劉俊麟 博士

中華民國 113 年 12 月 (14日)

March 3, 2025

Contents

List of Figures

List of Algorithms

Chapter 1

Details of the Beamformers

Based on the problem formulation, consider a uniform linear array with N elements and inter-element spacing $d = \lambda/2$. The output $\mathbf{x}(t)$ of this array is modeled as

$$\mathbf{x}(t) = \mathbf{a}(\theta_s(t))s(t) + \mathbf{a}(\theta_i(t))i(t) + \mathbf{n}(t) \tag{1.1}$$

where the time index $t = 1, 2, ..., L$. The steering vector is defined as

$$\begin{bmatrix} 1 \\ e^{j\pi \sin\theta} \\ e^{j2\pi \sin\theta} \\ \vdots \\ e^{j(N-1)\pi \sin\theta} \end{bmatrix} \tag{1.2}$$

The source signal $s(t) \in \mathbb{C}$, the interference signal $i(t) \in \mathbb{C}$, and the noise vector $\mathbf{n}(t) \in \mathbb{C}^N$.

The beamformer output can be written in the following form:

$$y(t) = \mathbf{w}^H(\mathbf{a}(\theta)s(t) + \mathbf{n}(t)) = \mathbf{B}_\theta(\theta) + \mathbf{w}^H\mathbf{n}(t) \tag{1.3}$$

where $\mathbf{B}_\theta(\theta) = \mathbf{w}^H\mathbf{a}(\theta)$ is the beampattern of the beamformer.

1.1 The Beamformer with Uniform Weights

The so-called beamformer with uniform weights refers to the weight vector:

$$\mathbf{w} = \frac{1}{N}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{1.4}$$

Therefore, the beampattern can be expressed as follows:

$$\mathbf{B}_\theta(\theta) = \frac{1}{N}\mathbf{1}^H\mathbf{a}(\theta) = \frac{1}{N}\sum_{n=0}^{N-1} e^{j\pi \sin\theta n} \tag{1.5}$$

$$= \begin{cases} \frac{1}{N}\frac{1-e^{jN\pi\sin\theta}}{1-e^{j\pi\sin\theta}}, \text{ if } e^{j\pi\sin\theta} \neq 1 \\ 1, \text{ if } e^{j\pi\sin\theta} = 1 \end{cases} \tag{1.6}$$

$$= e^{j\frac{N-1}{2}\pi\sin\theta} \times \frac{1}{N} \times \frac{\sin(\frac{N}{2}\pi\sin\theta)}{\sin(\frac{1}{2}\pi\sin\theta)} \tag{1.7}$$

From the above equation, it can be derived that

$$|\mathbf{B}_\theta(\theta)| = \frac{1}{N}\left|\frac{\sin(\frac{N}{2}\pi\sin\theta)}{\sin(\frac{1}{2}\pi\sin\theta)}\right| \tag{1.8}$$

From this, it can be seen that the beampattern reaches its maximum value of 1 when $\theta = 0$.

1.2 The Beamformer with Array Steering

The concept of the beamformer with array steering builds upon the uniform weights approach. In the uniform weights beamformer, the maximum value occurs at $\theta = 0$, meaning the mainlobe is enhanced for $\theta = 0$. However, not all signals originate from the 0-degree direction. Therefore, given a known direction of arrival

(DOA), assume the DOA is θ_s. The weight vector is defined as:

$$\mathbf{w} = \frac{1}{N}\mathbf{a}(\theta_s) \tag{1.9}$$

The beampattern can then be written as:

$$\mathbf{B}_\theta(\theta) = \mathbf{w}^H \mathbf{a}(\theta) \tag{1.10}$$

$$= e^{j\frac{N-1}{2}(\pi \sin\theta - \pi \sin\theta_s)} \times \frac{1}{N} \times \frac{\sin(\frac{N}{2}(\pi \sin\theta - \pi \sin\theta_s))}{\sin(\frac{1}{2}(\pi \sin\theta - \pi \sin\theta_s))} \tag{1.11}$$

From this, it can be derived that:

$$|\mathbf{B}_\theta(\theta)| = \frac{1}{N}\left|\frac{\sin(\frac{N}{2}(\pi \sin\theta - \pi \sin\theta_s))}{\sin(\frac{1}{2}(\pi \sin\theta - \pi \sin\theta_s))}\right| \tag{1.12}$$

Thus, the beampattern reaches its maximum value of 1 when $\theta = \theta_s$.

1.3 The MVDR Beamformer

MVDR stands for Minimum Variance Distortionless Response. The mathematical model of the MVDR beamformer is given by the following optimization problem:

$$\text{minimize} \quad \mathbb{E}[|y(t)|^2] \tag{1.13}$$

$$\text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta_s) = 1 \tag{1.14}$$

where $y(t) = \mathbf{w}^H \mathbf{x}(t)$. Solving this optimization problem minimizes the noise term $\mathbb{E}[\mathbf{w}^H(t)\mathbf{n}(t)]$ in $y(t)$. The above optimization problem is equivalent to:

$$\text{minimize} \quad \mathbf{w}^H \mathbf{R} \mathbf{w} \tag{1.15}$$

$$\text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta_s) = 1 \tag{1.16}$$

where $\mathbf{R} = \mathbb{E}[\mathbf{x}(t)\mathbf{x}^H(t)]$. The above problem can be solved using the Lagrange multiplier, and the mathematical expression is given as:

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^H \mathbf{R}\mathbf{w} - Re\{\lambda(\mathbf{w}^H \mathbf{a}(\theta_s) - 1)\} \tag{1.17}$$

By letting $\frac{\partial \mathcal{L}(\mathbf{w},\lambda)}{\partial \mathbf{w}^*} = 0$, we get

$$\mathbf{R}\mathbf{w} - \frac{1}{2}\lambda \mathbf{a}(\theta_s) = 0 \implies \mathbf{w} = \frac{1}{2}\lambda \mathbf{R}^{-1}\mathbf{a}(\theta_s) \tag{1.18}$$

Since we have the constraint $\mathbf{w}^H \mathbf{a}(\theta_s) = 1$, so

$$\lambda = \frac{2}{\mathbf{a}^H(\theta_s)\mathbf{R}^{-1}\mathbf{a}(\theta_\mathbf{s})} \tag{1.19}$$

Taking equation (1.19) into equation (1.18), we can get the optimal MVDR weight vector has the following form:

$$\mathbf{w}_{MVDR} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta_s)}{\mathbf{a}^H(\theta_s)\mathbf{R}^{-1}\mathbf{a}(\theta_s)} \tag{1.20}$$

1.4 The LCMV beamformer

The LCMV (Linearly Constrained Minimum Variance) beamformer is similar to the MVDR beamformer. Although MVDR can suppress the noise variance, it may suffer a bad performance when there is an interference signal. The LCMV beamformer not only consider the noise $\mathbf{n}(t)$ but the interference signal $i(t)$. Thus, the optimizatino problem can be formulated as the following form:

$$\text{minimize} \quad \mathbf{w}^H \mathbf{R}\mathbf{w} \tag{1.21}$$

$$\text{subject to} \quad \mathbf{C}^H \mathbf{w} = g \tag{1.22}$$

where $\mathbf{C} = \begin{bmatrix} \mathbf{a}(\theta_s) & \mathbf{a}(\theta_i) \end{bmatrix}$, and $\mathbf{g} = \begin{bmatrix} 1 \\ g_i* \end{bmatrix}$. Here θ_i denotes the DOA of the interference signal. Similarly, we use Lagrange multiplier to solve this optimization problem.

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^H \mathbf{R} \mathbf{w} - Re\{\lambda(\mathbf{C}^H \mathbf{w} - g)\} \tag{1.23}$$

Letting $\frac{\partial \mathcal{L}(\mathbf{w}, \lambda}{\partial \mathbf{w}*} = 0$, we get

$$\mathbf{R}\mathbf{w} - \frac{1}{2}\mathbf{C}\lambda = 0 \implies \mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}\mathbf{C}\lambda \tag{1.24}$$

Taking equation (1.24) back to constraint, we get

$$\mathbf{C}^H \mathbf{w} = g \implies \lambda = 2(\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{g} \tag{1.25}$$

According to equation (1.24) and equation (1.25), the optimal LCMV weight vector has the following form:

$$\mathbf{w}_{LCMV} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{g} \tag{1.26}$$

Chapter 2

De-noising $\tilde{\theta}_s(t)$ and $\tilde{\theta}_i(t)$

2.1 Motivation

The algorithm I chose to de-noise the DOA is the Kalman filter. Initially, I tried several types of de-noising algorithms such as wavelet transform and EMD algorithm. However, their performance was unsatisfactory. Consequently, I decided to try the Kalman filter, which was introduced in the lecture. Surprisingly, it outperformed the other algorithms I had tested. In the following sections, I will briefly review the mathematical concepts and the parameters I used.

2.2 Kalman Filter

Below is the block diagram of the Kalman filter, taken from the lecture slides.

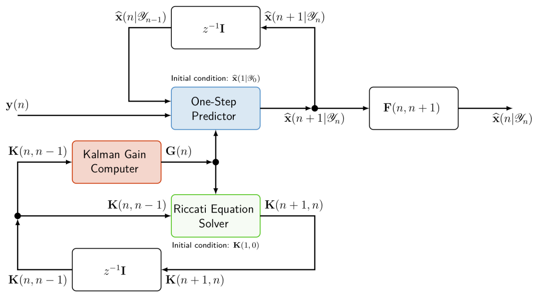

Figure 2.1: Block diagram of Kalman filter

We can see from Figure 2.1 that the Kalman filter consists of three main components: a one-step predictor, a Riccati equation solver, and a Kalman gain calculator. First, the Kalman filter receives an observation signal $\mathbf{y}(t) \in \mathbb{C}^N$, which is represented as:

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n), \tag{2.1}$$

where $\mathbf{x}(n) \in \mathbb{C}^M$ is the state vector at time n, $\mathbf{C}(n) \in \mathbb{C}^{N\times M}$ is the measurement matrix, and $\mathbf{v}_2(n) \in \mathbb{C}^N$ is the measurement noise. The objective is to estimate $\mathbf{x}(n)$ based on the observations $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)$. Let $\mathcal{Y}_n = \text{span}\{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)\}$. The output of the Kalman filter is then $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$.

For the one-step predictor, the innovation process of $\mathbf{y}(n)$, denoted as $\boldsymbol{\alpha}(n) \in \mathbb{C}^N$, is calculated as:

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}). \tag{2.2}$$

The mathematical derivation was covered in class and is omitted here. The output of the one-step predictor is:

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}(n)\boldsymbol{\alpha}(n), \tag{2.3}$$

where $\mathbf{F}(n+1,n) \in \mathbb{C}^{M\times M}$ is the transition matrix from time n to $n+1$, and $\mathbf{G}(n) \in \mathbb{C}^{M\times N}$ is the Kalman gain.

To compute the Kalman gain, the correlation matrix of $\boldsymbol{\alpha}(n)$, denoted as $\mathbf{R}(n) = \mathbb{E}[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(n)]$, is calculated as:

$$\mathbf{R}(n) = \mathbf{C}(n)\mathbf{K}(n,n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n), \tag{2.4}$$

where $\mathbf{K}(n,n-1) \in \mathbb{C}^{M\times M}$ is the predicted state-error correlation matrix, and $\mathbf{Q}_2(n) = \mathbb{E}[\mathbf{v}_2(n)\mathbf{v}_2^H(n)] \in \mathbb{C}^{N\times N}$. The Kalman gain is then computed as:

$$\mathbf{G}(n) = \mathbf{K}(n,n-1)\mathbf{C}^H(n)[\mathbf{R}(n)]^{-1}. \tag{2.5}$$

The correlation matrices $\mathbf{K}(n)$ and $\mathbf{K}(n+1, n)$ are needed to calculate the Kalman gain. Using the Riccati equation, these matrices are updated as follows:

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{R}(n)]^{-1}\mathbf{C}(n)\mathbf{K}(n, n-1), \tag{2.6}$$

$$\mathbf{K}(n+1, n) = \mathbf{F}(n+1, n)\mathbf{K}(n)\mathbf{F}^H(n+1, n) + \mathbf{Q}_1(n), \tag{2.7}$$

where $\mathbf{Q}_1(n) = \mathbb{E}[\mathbf{v}_1(n)\mathbf{v}_1^H(n)]$ is the process noise covariance matrix.

Using these equations, the Kalman filter integrates its three main modules to efficiently estimate the input signals based on prior observations and noise statistics.

2.3 Algorithm and Parameters

The Kalman filter is applicable to general cases. For Problem 2 in the final project, the observation signals are the noisy source DOA $\tilde{\theta}_s(t)$ and noisy interference DOA $\tilde{\theta}_i(t)$. The objective is to estimate the true values of the source DOA $\theta_s(t)$ and the interference DOA $\theta_i(t)$. Assuming discrete time t, the observation equation is:

$$\hat{\theta}_s(t) = \theta_s(t) + e_s(t), \tag{2.8}$$

where the measurement matrix $\mathbf{C}(t) = 1$, and the measurement noise $\mathbf{v}_2(t) = e_s(t)$. Assuming a static state transition for the source signal, we set $\mathbf{F}(t+1, t) = 1$ and $\mathbf{K}(t+1, t) = 1$. The initial estimate is $\hat{\theta}(1) = \tilde{\theta}(1)$. The algorithm is summarized as follows:

Algorithm 1 Kalman filter for Problem 2

$\hat{\theta}(1) \leftarrow \tilde{\theta}(1)$

$\mathbf{K}(t+1, t) \leftarrow 1$

$T \leftarrow$ Time length

$\mathbf{Q}_2(t) \leftarrow \mathbb{E}[e(t)e^H(t)]$

for $t = 1, 2, ..., T$ **do**

  $\mathbf{G}(t) \leftarrow \mathbf{K}(t, t-1)[\mathbf{K}(t, t-1) + \mathbf{Q}_2(t)]^{-1}$

  $\alpha(t) \leftarrow \tilde{\theta}(t) - \hat{\theta}(t|\mathbf{\Theta}_{t-1})$

  $\hat{\theta}(t+1|\mathbf{\Theta}_t) \leftarrow \hat{\theta}(t|\mathbf{\Theta}_{t-1}) + \mathbf{G}(t)\alpha(t)$

  $\mathbf{K}(t) \leftarrow \mathbf{K}(t, t-1) - \mathbf{G}(t)\mathbf{K}(t, t-1)$

  $\mathbf{K}(t+1, t) \leftarrow \mathbf{K}(t) + \mathbf{Q}_1(t)$

  $\hat{\theta}(t) \leftarrow \hat{\theta}(t+1|\mathbf{\Theta}_t)$

end for

2.4 Advantages and Results

The main advantage of using the Kalman filter for de-noising the DOA is its robust mathematical foundation, which guarantees reliable performance and facilitates result analysis. Furthermore, the Kalman filter performs exceptionally well when dealing with Gaussian noise, outperforming other de-noising algorithms.

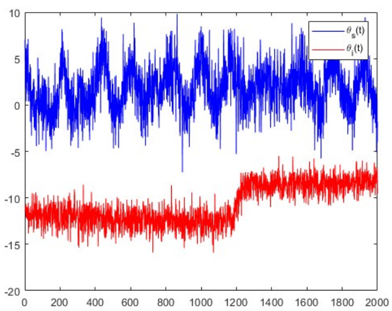

Figure 2.2: Noisy DOA

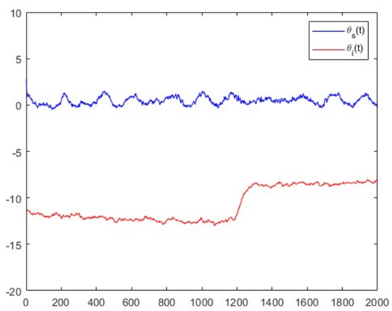

Figure 2.3: DOA after de-noising

Figures 3.4 and 3.5 depict the noisy DOA before and after de-noising, respectively. Clearly, the noisy DOA has been effectively de-noised, demonstrating the effectiveness of the Kalman filter.

Chapter 3

Subspace-Based Adaptive Beamforming

3.1 System Model

The subspace-based adaptive beamforming is essentially an improved version of the MVDR beamformer. Consider a uniform linear array (ULA) composed of Q array elements, one desired signal, and $J-1$ interference signals. The output of the array can be determined by the following formula:

$$\mathbf{x}(t) = \mathbf{d}(t) + \mathbf{i}(t) + \mathbf{n}(t) \tag{3.1}$$

$$= \mathbf{a}(\theta_1)s_1(t) + \sum_{i=2}^{J} \mathbf{a}(\theta_i)s_i(t) + \mathbf{n}(t), \tag{3.2}$$

where $\mathbf{d}(t) = \mathbf{a}(\theta_1)s_1(t) \in \mathbb{C}^Q$ is the desired signal vector, $\mathbf{i}(t) = \sum_{i=2}^{J} \mathbf{a}(\theta_i)s_i(t) \in \mathbb{C}^Q$ is the sum of interference signals, and $\mathbf{n}(t) \in \mathbb{C}^Q$ represents the noise vector. The output of the beamformer is expressed as:

$$y(t) = \mathbf{w}^H\mathbf{x}(t), \tag{3.3}$$

where $\mathbf{w} \in \mathbb{C}^Q$ is the weight vector.

We denote $\mathbf{R}_{i+n}$ as the covariance matrix of interference plus noise, defined as:

$$\mathbf{R}_{i+n} = \mathbb{E}[(\mathbf{i}(t) + \mathbf{n}(t))(\mathbf{i}(t) + \mathbf{n}(t))^H]. \tag{3.4}$$

Using the MVDR beamforming algorithm, the beamformer is constructed by solving the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w} & (3.5) \\ \text{subject to} \quad & \mathbf{w}^H \mathbf{a}(\theta_1) = 1. & (3.6) \end{aligned}$$

The detailed mathematical derivation is provided in Section 1.3. Here, we simply present the result for the MVDR beamformer:

$$\mathbf{w}_{\text{opt}} = \frac{\mathbf{R}_{i+n}^{-1} \mathbf{a}(\theta_1)}{\mathbf{a}^H(\theta_1) \mathbf{R}_{i+n}^{-1} \mathbf{a}(\theta_1)}. \tag{3.7}$$

3.2 The Proposed Algorithm

Now, we introduce the mathematical derivation and concept of subspace-based adaptive beamforming. Unlike MVDR beamforming, we consider the effect of interference. In an ideal scenario without noise or interference, $\mathbf{R}_x = \mathbb{E}[\mathbf{x}(t)\mathbf{x}^H(t)]$ would be a rank-1 matrix. However, in practice, $\text{rank}(\mathbf{R}_x) > 1$ due to interference and noise.

The core idea of subspace-based beamforming is that, assuming the power of the desired signal is greater than the interference and noise, the largest eigenvalue of $\mathbf{R}_x$ corresponds to the signal power. The next $J - 1$ largest eigenvalues correspond to the interference, while the smallest $Q - J$ eigenvalues represent the noise power.

To perform subspace-based adaptive beamforming, we perform eigenvalue decomposition of $\mathbf{R}_x$:

$$\mathbf{R}_x = \mathbf{U}_S \mathbf{\Sigma}_S \mathbf{U}_S^H + \mathbf{U}_N \mathbf{\Sigma}_N \mathbf{U}_N^H, \tag{3.8}$$

where $\mathbf{U}_S$ is the eigenvector matrix for the J signal components (desired signal + interference) and $\mathbf{\Sigma}_S$ is a diagonal matrix containing the eigenvalues of the signals, with the remaining diagonal entries set to 0. Similarly, $\mathbf{U}_N$ is the eigenvector matrix for the noise, and $\mathbf{\Sigma}_N$ is the diagonal matrix for the noise eigenvalues.

From linear algebra, we can write:

$$\mathbf{R}_x = \sum_{i=1}^{Q} \lambda_i \mathbf{e}_i \mathbf{e}_i^H, \tag{3.9}$$

where λ_i are the eigenvalues of $\mathbf{R}_x$, and $\mathbf{e}_i$ are the corresponding eigenvectors. Without loss of generality, we sort λ_i in descending order. Then,

$$\mathbf{U}_S = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_J \end{bmatrix}. \tag{3.10}$$

Since $\mathbf{R}_x$ is a positive definite matrix, the eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_Q$ are orthogonal. Moreover, the direction vectors of different signals are orthogonal to each other. Thus,

$$\mathbf{e}_j^H \mathbf{a}(\theta_i) = \begin{cases} \sqrt{Q}, & i = j, \\ 0, & \text{otherwise}. \end{cases} \tag{3.11}$$

We rewrite $\mathbf{R}_x$ as:

$$\begin{aligned} \mathbf{R}_x &= \sum_{i=1}^{Q} \lambda_i \mathbf{e}_i \mathbf{e}_i^H & (3.12) \\ &= \sum_{i=1}^{J} \left(\frac{\lambda_i - \sigma_n^2}{Q} \right) \mathbf{a}(\theta_i)\mathbf{a}^H(\theta_i) + \sigma_n^2 \mathbf{I}_Q & (3.13) \\ &= \sum_{i=1}^{J} \sigma_i^2 \mathbf{a}(\theta_i)\mathbf{a}^H(\theta_i) + \sigma_n^2 \mathbf{I}_Q, & (3.14) \end{aligned}$$

where σ_n^2 represents noise variance. By assumption, the smallest $Q - J$ eigenvalues correspond to noise power. The noise variance can thus be estimated as:

$$\tilde{\sigma}_n^2 = \frac{1}{Q-J} \sum_{i=J+1}^{Q} \lambda_i. \tag{3.15}$$

The signal power can be estimated as:

$$\tilde{\sigma}_i^2 = \frac{\lambda_i - \tilde{\sigma}_n^2}{Q}, \quad i = 1, 2, \ldots, J. \tag{3.16}$$

The estimated covariance matrix of interference plus noise is then:

$$\tilde{\mathbf{R}}_{i+n} = \sum_{i=2}^{J} \tilde{\sigma}_i^2 \mathbf{a}(\theta_i) \mathbf{a}^H(\theta_i) + \tilde{\sigma}_n^2 \mathbf{I}_Q. \tag{3.17}$$

Finally, substituting this back into the MVDR weight equation, the optimal solution for subspace-based adaptive beamforming is:

$$\tilde{\mathbf{w}}_{\text{opt}} = \frac{\tilde{\mathbf{R}}_{i+n}^{-1} \mathbf{a}(\theta_1)}{\mathbf{a}^H(\theta_1) \tilde{\mathbf{R}}_{i+n}^{-1} \mathbf{a}(\theta_1)}. \tag{3.18}$$

The algorithm is summarized as follows:

Algorithm 2 Subspace-Based Adaptive Beamforming

$L \leftarrow$ Time length
$\mathbf{R}_x \leftarrow \frac{1}{L} \sum_{t=1}^{L} \mathbf{x}(t) \mathbf{x}^H(t)$
Eigenvalue decomposition of $\mathbf{R}_x$
Sort eigenvalues in descending order
$\tilde{\sigma}_n^2 \leftarrow \frac{1}{Q-J} \sum_{i=J+1}^{Q} \lambda_i$
for $j = 1, 2, \ldots, J$ **do**
    $\tilde{\sigma}_j^2 = \frac{\lambda_j - \tilde{\sigma}_n^2}{Q}$
end for
for $t = 1, 2, \ldots, L$ **do**
    $\tilde{\mathbf{R}}_{i+n} \leftarrow \sum_{j=2}^{J} \tilde{\sigma}_j^2 \mathbf{a}(\theta_j(t)) \mathbf{a}^H(\theta_j(t)) + \tilde{\sigma}_n^2 \mathbf{I}_Q$
    $\tilde{\mathbf{w}}_{\text{opt}}(t) \leftarrow \frac{\tilde{\mathbf{R}}_{i+n}^{-1} \mathbf{a}(\theta_1)}{\mathbf{a}^H(\theta_1) \tilde{\mathbf{R}}_{i+n}^{-1} \mathbf{a}(\theta_1)}$
    $y(t) = \tilde{\mathbf{w}}_{\text{opt}}^H \mathbf{x}(t)$
end for

3.3 Beamforming Results

Figures below show the beampatterns of MVDR, LCMV, and the proposed algorithm.

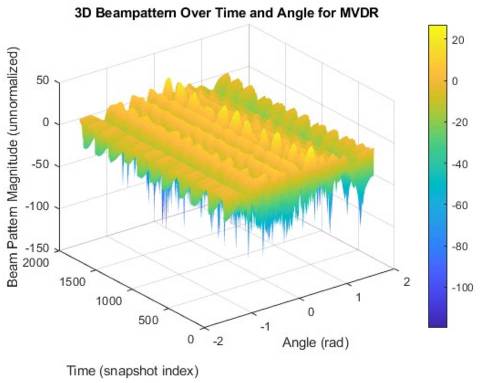

Figure 3.1: Beampattern of MVDR

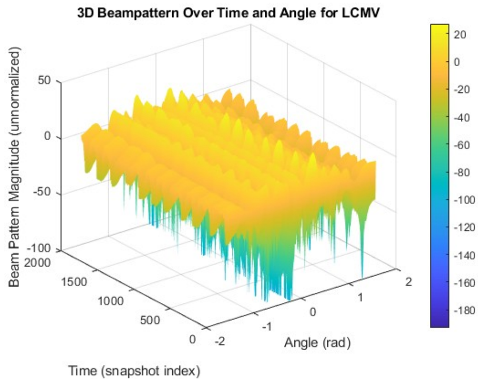

Figure 3.2: Beampattern of LCMV

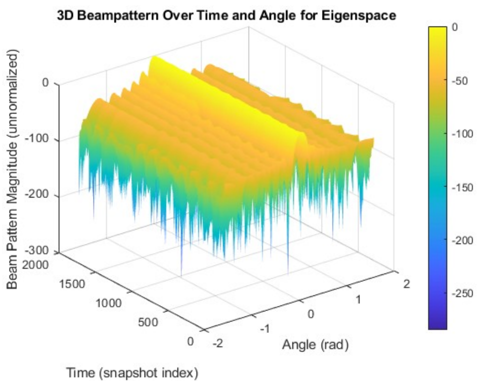

Figure 3.3: Beampattern of proposed algorithm

It is evident that the subspace-based beamforming outperforms MVDR and LCMV. The following figures show the estimated $\hat{s}(t)$ using MVDR, LCMV, and the proposed algorithm. The real part is in the upper half, and the imaginary part is in the lower half.

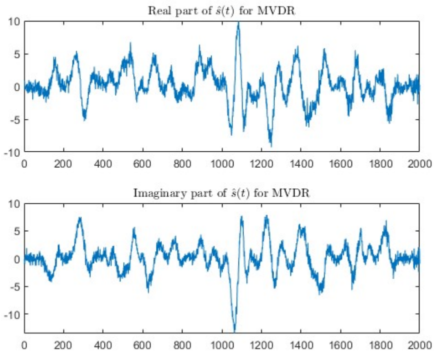

Figure 3.4: $\hat{s}(t)$ by MVDR

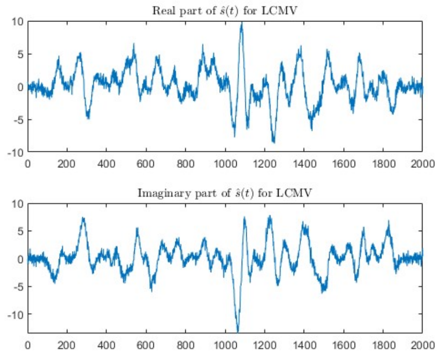

Figure 3.5: $\hat{s}(t)$ by LCMV

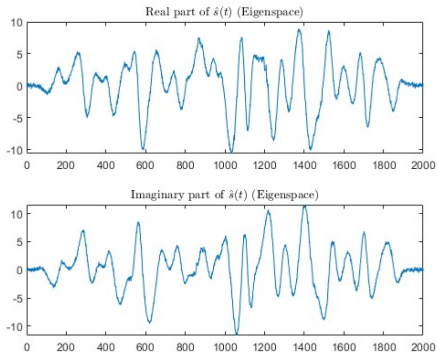

Figure 3.6: $\hat{s}(t)$ by proposed algorithm

The results clearly demonstrate that the subspace-based adaptive beamforming outperforms MVDR and LCMV, validating the effectiveness of the proposed algorithm.

To verify the distinction between the proposed algorithm and other beamformers, the weight vectors $\mathbf{w}_{\text{opt}}$ for MVDR, LCMV, and the proposed algorithm are displayed in the figure below. The uniform weights and array steering beamformers are excluded as their weight vectors are trivial: $\frac{1}{N}\mathbf{1}$ and $\frac{1}{N}\mathbf{a}(\theta)$, respectively.

```
>> w_MVDR(:, end)

ans =

   0.2813 + 0.4753i
  -0.0259 + 0.0468i
  -0.0988 + 0.0059i
   0.0103 - 0.1089i
   0.1027 + 0.0603i
   0.2510 + 0.0213i
   0.2109 - 0.0588i
   0.0321 - 0.0451i
   0.0673 + 0.0114i
  -0.0643 + 0.0986i
  -0.0090 - 0.0629i
   0.1528 - 0.5226i
```

Figure 3.7: The last weight vector of MVDR

```
>> w_LCMV(:, end)

ans =

   0.6105 + 0.3420i
  -0.0108 - 0.0692i
  -0.2419 + 0.0424i
  -0.0799 - 0.0552i
   0.0267 + 0.1463i
   0.2191 + 0.0669i
   0.1780 - 0.1250i
  -0.0007 - 0.0993i
  -0.0737 - 0.0371i
  -0.2121 + 0.1170i
   0.0172 + 0.0352i
   0.5045 - 0.4443i
```

Figure 3.8: The last weight vector of MVDR

```
>> w_eig(:, end)

ans =

   0.0990 - 0.0122i
   0.0926 - 0.0175i
   0.0851 - 0.0197i
   0.0780 - 0.0186i
   0.0726 - 0.0148i
   0.0699 - 0.0091i
   0.0704 - 0.0030i
   0.0740 + 0.0021i
   0.0800 + 0.0049i
   0.0872 + 0.0048i
   0.0942 + 0.0013i
   0.0996 - 0.0050i
```

Figure 3.9: The last weight vector of the proposed algorithm

The three figures, captured from MATLAB, display the weight vectors for MVDR, LCMV, and the proposed algorithm in order. It is evident that the proposed algorithm is distinct from the other beamformers.

Bibliography

[1] Liu, Linxian and Li, Yang, "Robust adaptive beamforming with enhancing the interference suppression capability," *SIP*, vol. 8, p. e15, Cambridge University Press, 2019. doi: 10.1017/ATSIP.2019.9.

[2] Du, Yuxi and Cui, Weijia and Wang, Yinsheng and Ba, Bin and Mei, Fengtong, "Subspace based adaptive beamforming algorithm with interference plus noise covariance matrix reconstruction," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6063500, Hindawi, 2021. doi: 10.1155/2021/6063500.

[3] Kalman, Rudolph Emil, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. doi: 10.1115/1.3662552.

[4] Donoho, David L, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995. doi: 10.1109/18.382009.