**FEDERAL UNIVERSITY OF CEARÁ**

**TECHNOLOGY CENTER**

**DEPARTMENT OF TELEINFORMATICS ENGINEERING**

**SEMESTER 2024.2**

# MUltiple SIgnal Classification (MUSIC), Root MUSIC and Minimum Variance Distortionless Response (MVDR) Algorithms

**STUDENT:** João Vitor de O. Fraga

**COURSE:** Telecommunications Engineering

**PROFESSOR:** Fazal-E-Asim

**SUBJECT:** Integrated Actions in Science and Technology I

# Table of Contents

# 1 Introduction

Direction of Arrival (DoA) estimation is a fundamental problem in array signal processing, widely used in radar, sonar, and wireless communications. Various algorithms have been developed to estimate the arrival angles of signals, including:

- MUSIC (Multiple Signal Classification), which uses the noise subspace for spectral estimation.
- Root MUSIC, an improved version of MUSIC that finds DoA estimates using polynomial roots.
- MVDR (Minimum Variance Distortionless Response), a beamforming technique that minimizes interference while preserving the desired signal.

This report details these algorithms, their mathematical formulations, and numerical results.

## 2 System Model

### 2.1 Mathematical Representation

We will use an architecture called Uniform Linear Array (ULA), which is represented in Fig. 1.

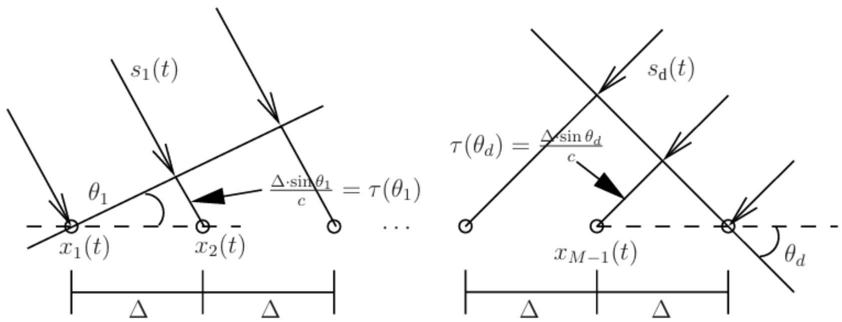


Figure 1: ULA representation.

Where:

- $\Delta$ is the spacing between one antenna and another;
- $M$ is the number of antennas;
- $\theta_d$ refers to the angle direction of arrival of the signal.

The time delay for a signal arriving at angle $\theta_i$ is:

$$\tau_M(\theta_i) = (M-1)\frac{\Delta \sin\theta_i}{c}, \tag{1}$$

where $c$ is the speed of propagation. The spatial frequency associated with the angle $\theta_i$ is given by:

$$\mu_i = -\frac{2\pi f_c}{c}\Delta \sin(\theta_i), \tag{2}$$

which, assuming $\Delta = \frac{\lambda}{2}$, simplifies to:

$$\mu_i = -\pi \sin(\theta_i). \tag{3}$$

## 2.2 Steering Matrix

The steering matrix, which represents the response of the array to incoming signals, is defined as:

$$\mathbf{A_{ULA}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j\mu_1} & e^{j\mu_2} & \cdots & e^{j\mu_d} \\ e^{j2\mu_1} & e^{j2\mu_2} & \cdots & e^{j2\mu_d} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(M-1)\mu_1} & e^{j(M-1)\mu_2} & \cdots & e^{j(M-1)\mu_d} \end{bmatrix} \in \mathbb{C}^{M\times d}. \tag{4}$$

Using this matrix, the received signal model can be expressed as:

$$\mathbf{X}(t) = \mathbf{A_{ULA}} \cdot \mathbf{S}(t) + \mathbf{N}(t), \tag{5}$$

where:

- $\mathbf{X} \in \mathbb{C}^{M\times t}$: Received signal at the antenna array.
- $\mathbf{A_{ULA}} \in \mathbb{C}^{M\times d}$: Steering matrix.
- $\mathbf{S} \in \mathbb{C}^{d\times t}$: Source signals.
- $\mathbf{N} \in \mathbb{C}^{M\times t}$: Additive noise.

Futhermore, the change in SNR is related to the standard deviation of the noise, as the transmission power remains constant. Thus, to change the SNR, we have:

$$P_{\text{noise}} = \frac{P_t}{\text{SNR}}, \tag{6}$$

Considering that we will transform the SNR from dB to linear and knowing that we model the noise as $P_{\text{noise}} \sim \mathcal{N}(0, \sigma^2)$, we have:

$$\sigma = \sqrt{\frac{P_t}{\text{SNR}}},\ P_t = 1 \therefore \sigma = \sqrt{\text{SNR}^{-1}} = \frac{1}{\sqrt{\text{SNR}}}. \tag{7}$$

Now, making the adjustment for the different SNR values.

## 2.3 Subspace Estimation

To work with all three algorithms, we need to estimate the noise subspace. This process consists of computing the covariance matrix, applying EVD or Singular Value Decomposition (SVD), and separating the signal and noise subspaces.

### 2.3.1 Covariance Matrix Computation

Given an $N$-element antenna array receiving $M$ signals, the covariance matrix of the received signals is estimated as:

$$\hat{\mathbf{R}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{X}(t)\mathbf{X}^H(t), \tag{8}$$

where $T$ is the number of snapshots.

### 2.3.2 EVD

The covariance matrix $\mathbf{R}$ is decomposed using EVD:

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H. \tag{9}$$

From this decomposition:

- The eigenvectors associated with the largest eigenvalues form the signal subspace $\mathbf{U_s}$.
- The eigenvectors associated with the smallest eigenvalues form the noise subspace $\mathbf{U_o}$.

## 2.4 MUSIC Algorithm

The MUSIC algorithm is a high-resolution subspace-based method for DoA estimation. It utilizes EVD on the covariance matrix, as described in Eq. (9), to separate the noise and signal subspaces.

The core principle of MUSIC is that the noise subspace $\mathbf{U_o}$ is orthogonal to the array steering vectors corresponding to the sources. Thus, the MUSIC spectrum is computed as:

$$S_{\text{MUSIC}}(\theta) = \frac{\mathbf{a}^H(\theta)\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{P_o}\mathbf{a}(\theta)}, \tag{10}$$

where:

- $\mathbf{P_o} = \mathbf{U_o}\mathbf{U_o}^H$ is the projector onto the noise subspace.
- $\mathbf{a}(\theta)$ is the steering vector, defined in Eq. (4).

## 2.5 Root MUSIC Algorithm

The Root MUSIC method formulates the problem as a polynomial, we start from the spectrum equation which is defined as:

$$P_{\text{RootMUSIC}}(\theta) = \frac{1}{|\mathbf{a}(\theta)^H \mathbf{C}\mathbf{a}(\theta)|} \tag{11}$$

Where, if we have a ULA, the n-th element of the steering vector array is:

$$a_m(\theta) = e^{jkd(m-1)\sin(\theta)},\ m = 1, 2, ..., M. \tag{12}$$

Now that we know this, we define the denominator of Eq. 11, as:

$$\begin{aligned} \mathbf{a}(\theta)^H \mathbf{C}\mathbf{a}(\theta) &= \sum_{m=1}^{M}\sum_{n=1}^{M} e^{-jkd(m-1)\sin\theta} C_{mn} e^{-jkd(n-1)\sin\theta} \\ &= \sum_{l=-M+1}^{M-1} c_l e^{jkdl\sin(\theta)} \end{aligned} \tag{13}$$

Where where $c_l$ is the sum of the diagonal elements of $\mathbf{C}$

From this we then define $z = e^{-jkd\sin\theta}$, and we are left with:

$$D(z) = \sum_{l=-M+1}^{M+1} c_l z^l \tag{14}$$

With this, we find the zeros of this equation and those that are closest to the circle of unit radius will be chosen to obtain the correct angle. This is because the Z transform has the property that a system is stable only if the pole is inside the unit circle, so we need this property.

Having defined the $z_i$ we are going to use, to find the angle of arrival we use:

$$\theta_i = -\arcsin\left(\frac{1}{kd}\arg(z_i)\right) \tag{15}$$

## 2.6 MVDR Algorithm

The MVDR beamformer minimizes the variance of noise while keeping the desired signal direction undistorted. The weights are computed as:

$$\mathbf{w}_{\mathrm{MVDR}} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta_0)}{\mathbf{a}^H(\theta_0)\mathbf{R}^{-1}\mathbf{a}(\theta_0)} \tag{16}$$

The beampattern is given by:

$$B(\theta) = \mathbf{w}_{\mathrm{MVDR}}^H \mathbf{a}(\theta). \tag{17}$$

# 3 Numerical Results

| **Parameter** | **Value** |
|---|---|
| Number of iterations | 5000 |
| Snapshot | 500 |
| Variation of $\theta$ | $\mathcal{U}[-60, 60]$ |
| Variation of Signal-to-Noise Ratio (SNR) | $\{-5, 0, 5, 10, 15, 20\}$ |

Table 1: Simulation parameters

The Root Mean Square Error (RMSE) is calculated as:

$$RMSE(\hat{\phi_r}) = \sqrt{\mathbb{E}[|\phi_r - \hat{\phi_r}|^2]}. \tag{18}$$

## 3.1 MUSIC

In Fig. 2 you can see the variation in the MUSIC spectrum as the SNR increases, and the higher our SNR, the more visible and precise our spectrum becomes.

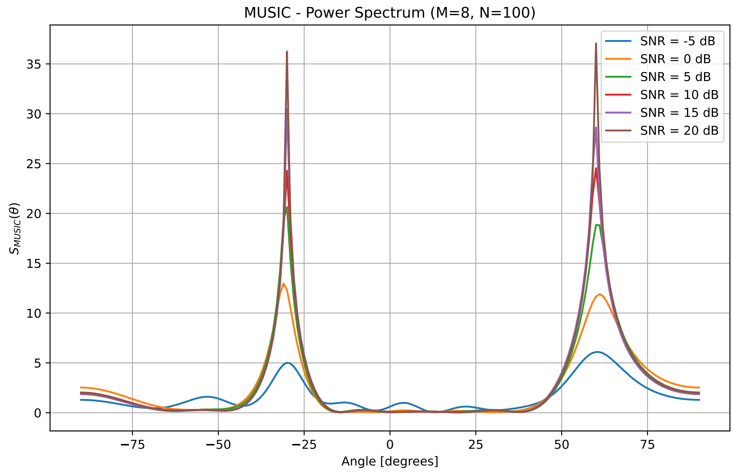


Figure 2: MUSIC with SNR variation

In Fig. 3, I varied the number of snapshots, which, like SNR, the higher the number, the more accurate the spectrum I obtained.

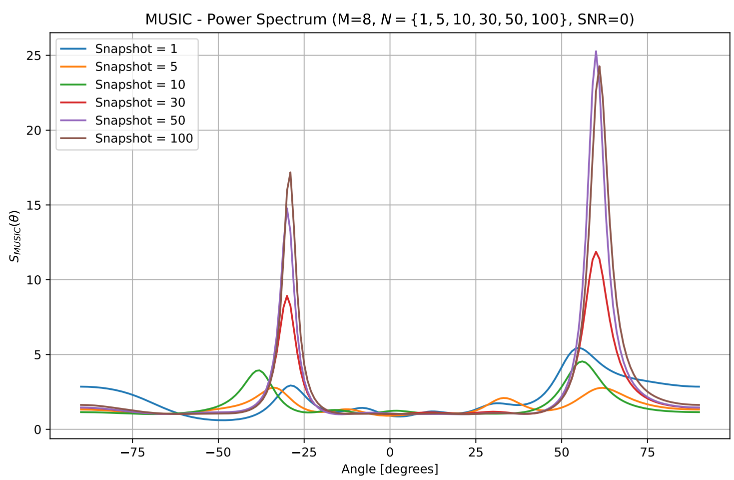


Figure 3: MUSIC with $t$ snapshot variation

To visualize the RMSE, we run a Monte Carlo simulation as mentioned in Table 1, where for each iteration a new random angle is generated.

That said, we can infer from Fig. 4 the expected behavior, since the higher the SNR, the better my algorithm will behave, due to the reduction in noise.

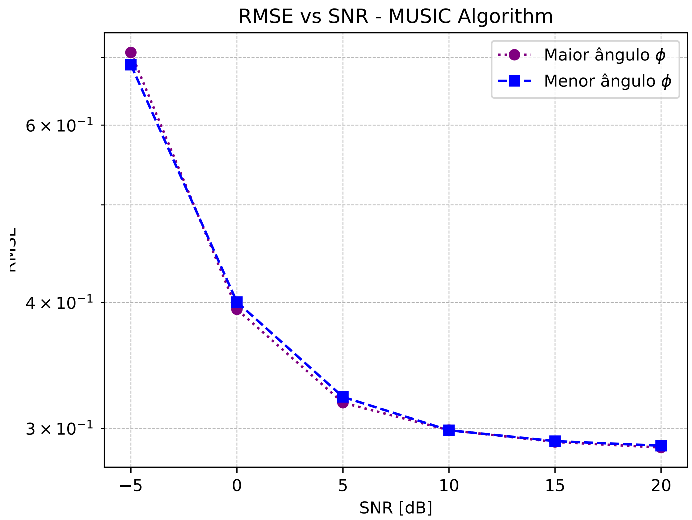


Figure 4: RMSE MUSIC Algorithm

However, it is possible to see that the values, even on a logarithmic scale, are still quite high. This is probably due to the “find peaks” function, which must not be set up very well, but which can and should be corrected in future work.

## 3.2 Root MUSIC

As for the root MUSIC, you can see the values of $z_i$ used to find the value of $\theta$ in Fig. 5, the map of poles and zeros.

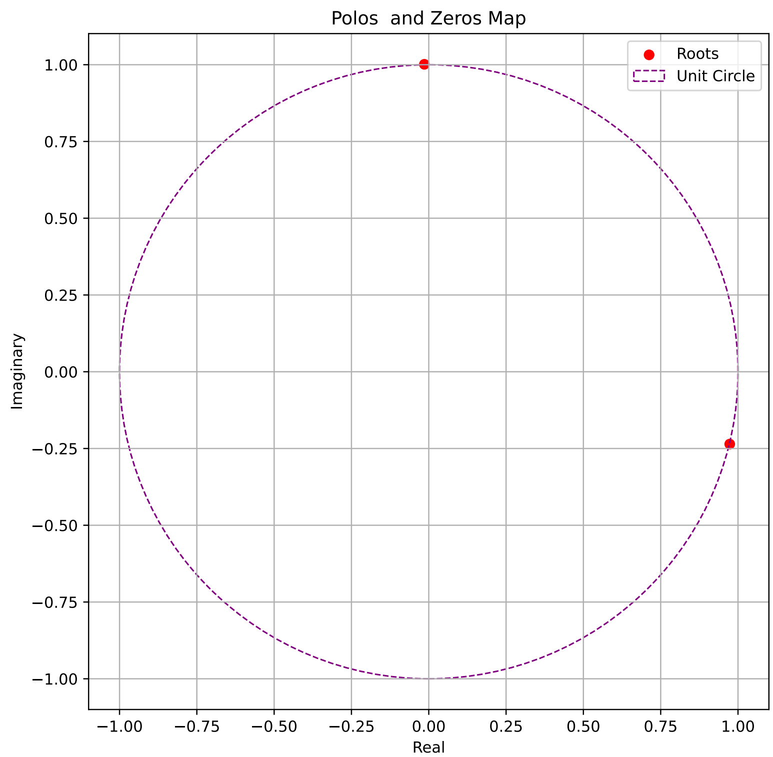


Figure 5: PZ Map - Root MUSIC

Now for the calculation of RMSE, two values were set: $[\theta_0 = -40,\ \theta_1 = 30]$, and from these values a monte carlo simulation was performed, where we can see the very low values of RMSE that the Root MUSIC algorithm gave us.

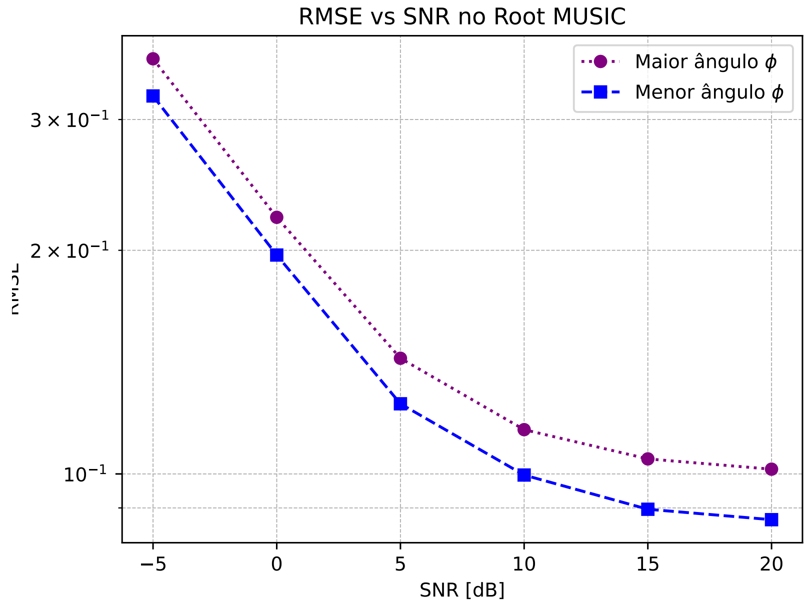


Figure 6: RMSE - Root MUSIC

### 3.3 MVDR

In Fig. 7, we present the MVDR beampattern, which shows a highly directional response, allowing suppression of undesired interferences.

The angle seted was defined by $\theta = 12$.

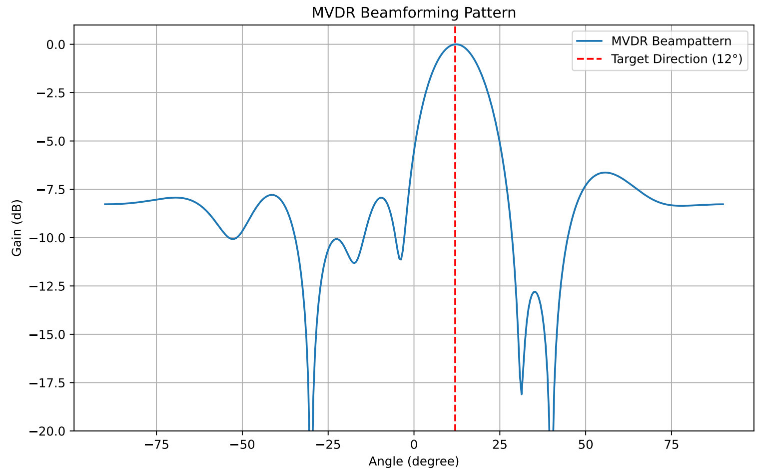


Figure 7: MVDR Beampattern

To better visualize the main lobe and sidelobes, a zoomed-in version is presented in Fig. 8.

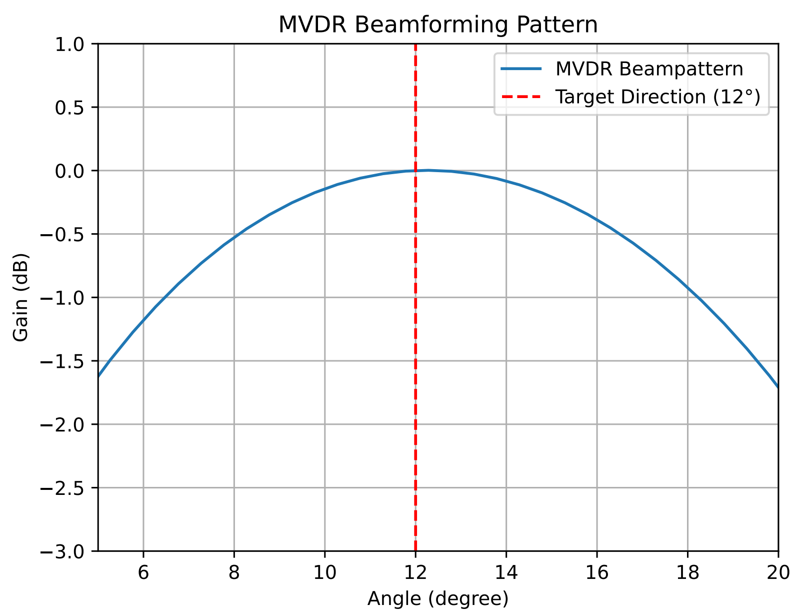


Figure 8: MVDR Beampattern with zoom

# 4 Conclusion

This report presented and analyzed three DoA estimation algorithms: MUSIC, Root MUSIC, and MVDR beamforming. Through numerical simulations, we observed the following:

- The MUSIC algorithm provides high-resolution DoA estimates, but its performance is affected by noise and the number of snapshots.
- Root MUSIC achieves similar accuracy to MUSIC, but is computationally more efficient since it avoids the spectral search.
- The MVDR beamformer effectively suppresses interference while maintaining a distortionless response in the desired direction.

The codes used during the course are available at this GitHub link.