IPSA Toulouse

Signal processing for Radar array

Project: Study and implementation of beamforming and DoA estimation techniques

Authors: Alyssa ARESSY, Lisa BLASCO, 5TS1

Supervisor: Samy LABSIR

2025 - 2026

Introduction

In this report, we consider a network of N ULA antennas that receives P sources with directions $\boldsymbol{\theta}_S = [\theta_1, ..., \theta_P]^\top \in \mathbb{R}^P$. First, we will implement the beamforming to identify the source of interest, and in the second part, estimate the directions of arrival of every source by using the DoA estimation methods.

1 Beamforming techniques

In this first section, the objective is to implement beamforming techniques by considering a specific scenario. For that, we consider that only one source, at each instant $k \in [\![1,\, K]\!]$, $s(k) \in \mathbb{R}$ with power P_S and direction, $\mathbf{a}(\theta_S)$ is received by the ULA and corrupted by an interference $i(k) \in \mathbb{R}$, with power P_I and direction $\mathbf{a}_I(\theta_I)$. We consider a ULA, reception is done over $\theta \in [0,\, 2\pi]$ and the expression of a source direction for such a network is for $n \in [\![1,\, N]\!]$:

$$\mathbf{a}(\theta) = \begin{pmatrix} 1 \\ \vdots \\ \exp^{-2\pi j (n-1) \frac{d}{\lambda} \sin(\theta)} \\ \vdots \\ \exp^{-2\pi j (N-1) \frac{d}{\lambda} \sin(\theta)} \end{pmatrix} \tag{1}$$

The signal received by the ULA $\mathbf{y}(k) \in \mathbb{C}^N$ can be expressed through the expression

$$\mathbf{y}(k) = \mathbf{a}(\theta_S)\; s(k) + \mathbf{y}_I(k) + \mathbf{n}(k) \qquad \mathbf{n}(k) \sim \mathcal{CN}(0, \sigma^2\, \mathbf{I}) \tag{2}$$

$$\text{with} \qquad s(k) \sim \mathcal{N}(0, P_S) \qquad \mathbf{y}_I(k) = \mathbf{a}_I(\theta_I)\, i(k) \qquad i(k) \sim \mathcal{N}(0, P_I) \tag{3}$$

The study is carried out by considering two cases depending on the knowledge of the interference-noise covariance matrix $\mathbf{C}$ and of the signal-interference-noise covariance matrix $\mathbf{R}$.

1.1 Known covariance matrices

In the first case, the covariance matrices involved in the problem are assumed to be completely known. Consequently, the matrices $\mathbf{C}$ and $\mathbf{R}$ can be computed theoretically and implemented without any approximation.

1.1.1 Covariance matrices R and C expression

By definition, we have the interference-noise covariance matrix $\mathbf{C}$ and the signal-interference-noise covariance matrix $\mathbf{R}$:

$$\begin{cases} \mathbf{C} = \sigma^2\, \mathbf{I} + P_I\; \mathbf{a}_I(\theta_I)\; \mathbf{a}_I(\theta_I)^\dagger = \mathbb{E}\left((\mathbf{y}_I(k) + \mathbf{n}(k))\, (\mathbf{y}_I(k) + \mathbf{n}(k))^\dagger \right) \\ \mathbf{R} = \mathbf{C} + P_S\; \mathbf{a}(\theta_S)\; \mathbf{a}(\theta_S)^\dagger = \mathbb{E}\left((\mathbf{a}(\theta_S)\; s(k) + \mathbf{y}_I(k) + \mathbf{n}(k))\, (\mathbf{a}(\theta_S)\; s(k) + \mathbf{y}_I(k) + \mathbf{n}(k))^\dagger \right) \end{cases}$$

First for $\mathbf{C}$:

$$\begin{aligned} \mathbf{C} &= \mathbb{E}\left((\mathbf{y}_I(k) + \mathbf{n}(k))\, (\mathbf{y}_I(k) + \mathbf{n}(k))^\dagger \right) \\ &= \mathbb{E}\left(\mathbf{y}_I(k)\mathbf{y}_I(k)^\dagger + \mathbf{y}_I(k)\mathbf{n}(k)^\dagger + \mathbf{n}(k)\mathbf{y}_I(k)^\dagger + \mathbf{n}(k)\mathbf{n}(k)^\dagger \right) \\ &= \mathbb{E}\left(\mathbf{y}_I(k)\mathbf{y}_I(k)^\dagger \right) + \mathbb{E}\left(\mathbf{y}_I(k)\mathbf{n}(k)^\dagger \right) + \mathbb{E}\left(\mathbf{n}(k)\mathbf{y}_I(k)^\dagger \right) + \mathbb{E}\left(\mathbf{n}(k)\mathbf{n}(k)^\dagger \right) \end{aligned} \tag{4}$$

By assuming independence between the noise and the interference, the crossed terms are 0 :

$$\mathbb{E}\left(\mathbf{y}_I(k)\mathbf{n}(k)^\dagger \right) = \mathbb{E}\left(\mathbf{n}(k)\mathbf{y}_I(k)^\dagger \right) = 0$$

We also know from Equation 2 that :

$$\mathbb{E}\left(\mathbf{y}_I(k)\mathbf{y}_I(k)^\dagger\right) = \mathbb{E}\Big(\Big(\mathbf{a}_I(\theta_I)i(k)\Big)\Big(\mathbf{a}_I(\theta_I)i(k)\Big)^\dagger\Big) \qquad i(k) \sim \mathcal{N}(0, P_I)$$

As $\mathbf{a}_I(\theta_I)$ is not random and $\mathbb{E}\big(i(k)i(k)^\top\big) = P_I \in \mathbb{R}$:

$$\begin{aligned}\mathbb{E}\left(\mathbf{y}_I(k)\mathbf{y}_I(k)^\dagger\right) &= \mathbf{a}_I(\theta_I)\mathbb{E}\Big(i(k)i(k)^\top\Big)\mathbf{a}_I(\theta_I)^\dagger && i(k) \sim \mathcal{N}(0, P_I)\\ &= \mathbb{E}\Big(i(k)i(k)^\top\Big)\ \mathbf{a}_I(\theta_I)\mathbf{a}_I(\theta_I)^\dagger && i(k) \sim \mathcal{N}(0, P_I)\\ &= P_I\ \mathbf{a}_I(\theta_I)\mathbf{a}_I(\theta_I)^\dagger\end{aligned}$$

From Equation 2 directly with $\mathbf{n}(k) \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$: $\mathbb{E}\left(\mathbf{n}(k)\ \mathbf{n}(k)^\dagger\right) = \sigma^2\mathbf{I}$. Replacing terms in Equation 4 :

$$\mathbf{C} = \sigma^2\mathbf{I} + P_I\ \mathbf{a}_I(\theta_I)\mathbf{a}_I(\theta_I)^\dagger$$

Likewise for $\mathbf{R}$, we identify the same terms from $\mathbf{C}$ plus the additional ones :

$$\mathbf{R} = \mathbb{E}\left((\mathbf{a}(\theta_S)\ s(k) + \mathbf{y}_I(k) + \mathbf{n}(k))\,(\mathbf{a}(\theta_S)\ s(k) + \mathbf{y}_I(k) + \mathbf{n}(k))^\dagger\right)$$

$$\mathbf{R} = \mathbf{C} + \mathbb{E}\Big(\big(\mathbf{a}(\theta_S)\ s(k)\big)\ \big(\mathbf{a}(\theta_S)\ s(k)\big)^\dagger\Big) + \mathbb{E}\Big(\big(\mathbf{a}(\theta_S)\ s(k)\big)\ \big(\mathbf{y}_I(k) + \mathbf{n}(k)\big)^\dagger\Big) + \\ \mathbb{E}\Big(\big(\mathbf{y}_I(k) + \mathbf{n}(k)\big)\ \big(\mathbf{a}(\theta_S)\ s(k)\big)^\dagger\Big)$$

By assuming independence between the noise and the signal and independence between the interference and the signal, crossed terms are 0 and with $\mathbf{a}(\theta_S)$ not random and $\mathbb{E}\big(s(k)s(k)^\top\big) = P_S \in \mathbb{R}$:

$$\begin{aligned}\mathbf{R} &= \mathbf{C} + \mathbb{E}\big(s(k)s(k)^\top\big)\,\mathbf{a}(\theta_S)\,\mathbf{a}(\theta_S)^\dagger \qquad s(k) \sim \mathcal{N}(0, P_S)\\ &= \mathbf{C} + P_S\,\mathbf{a}(\theta_S)\,\mathbf{a}(\theta_S)^\dagger\end{aligned}$$

With these equations, we can implement the beamforming techniques. We can note that beamforming only depends on the antenna network geometry, including the positions of antennas, wavelength, and DoA, so we do not need to implement the received signal $\mathbf{y}(t)$ for these techniques.

The goal of this part is to compare the different methods. We start by implementing the conventional beamforming $\mathbf{w}_{CBF}$ and verifying the relation between the Signal-Interference-Noise ratios ($SINR_{out} = N\,SINR_{in}$). Once it is done, we implement the optimal adaptive beamforming $\mathbf{w}_{opt}$ which uses the true covariance matrix $\mathbf{R}$ and its $SINR$. We obtain :

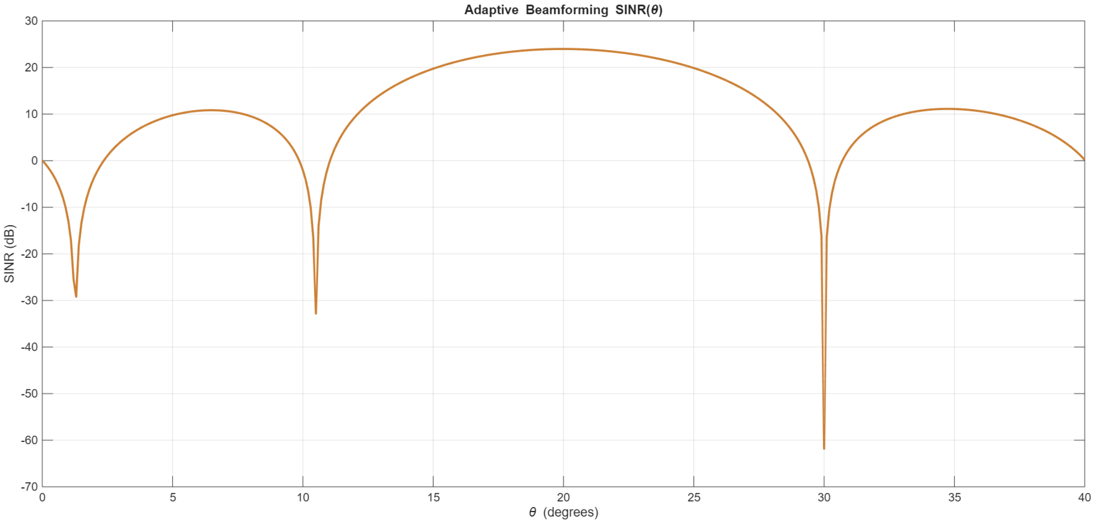


Figure 1: Adaptive Beamforming

Now, we assume that the angle θ_S is not very well-known and is provided by an estimator $\widehat{\theta_S}$. To handle this, we need to implement both the Minimum Power Distortionless Response ($MVDR$) filter $\mathbf{w}_{MVDR}$ and the Minimum Variance Distortionless Response ($MPDR$) filter $\mathbf{w}_{MPDR}$, considering that only the matrix $\mathbf{R}$ is known.

1.1.2 Signal-to-Interference and Noise Ratio (SINR) expression

We filter the signal at reception with a filter $\mathbf{w}$. Meaning the received signal is:

$$\mathbf{y}_R(k) = \mathbf{w}\,\mathbf{y}(k) = \underbrace{\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)}_{signal\ of\ interest} + \underbrace{\mathbf{w}^\dagger\ (\mathbf{y}_I(k) + \mathbf{n}(k))}_{interf.\ and\ noise}$$

The Signal to Interference and Noise Ratio (SINR) is, by definition, a ratio of the *useful* power part (associated with the signal) over the *lost* part (associated with the noise and the interference). At reception, it can be expressed as :

$$\begin{aligned} SINR_{out}(\mathbf{w}) &= \frac{\mathbb{E}\big(|\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)|^2\big)}{\mathbb{E}\big(|\mathbf{w}^\dagger\ (\mathbf{y}_I(k)+\mathbf{n}(k))|^2\big)} \\ &= \frac{\mathbb{E}\Big(\big(\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)\big)\,\big(\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)\big)^\dagger\Big)}{\mathbb{E}\Big(\big(\mathbf{w}^\dagger\ (\mathbf{y}_I(k)+\mathbf{n}(k))\big)\big(\mathbf{w}^\dagger\ (\mathbf{y}_I(k)+\mathbf{n}(k))\big)^\dagger\Big)} \\ &= \frac{\mathbb{E}\Big(\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)s(k)^\top\,\mathbf{a}(\theta)^\dagger\,\mathbf{w}\Big)}{\mathbb{E}\Big(\mathbf{w}^\dagger\,(\mathbf{y}_I(k)+\mathbf{n}(k))\,(\mathbf{y}_I(k)+\mathbf{n}(k))^\dagger\ \mathbf{w}\Big)} \end{aligned}$$

As $\mathbf{w}$ and $\mathbf{a}(\theta_S)$ not random, the expectation of $s(k)s(k)^\top$ is $\in \mathbb{R}$. Then from Equation 2 and Equation 3

$$\begin{aligned} SINR_{out}(\mathbf{w}) &= \frac{\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,\mathbb{E}\Big(s(k)s(k)^\top\Big)\,\mathbf{a}(\theta)^\dagger\,\mathbf{w}}{\mathbf{w}^\dagger\,\mathbb{E}\Big((\mathbf{y}_I(k)+\mathbf{n}(k))\,(\mathbf{y}_I(k)+\mathbf{n}(k))^\dagger\Big)\,\mathbf{w}} \\ &= \frac{P_S\,\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,\mathbf{a}(\theta)^\dagger\,\mathbf{w}}{\mathbf{w}^\dagger\,\mathbf{C}\,\mathbf{w}} = \frac{P_S\,|\mathbf{w}^\dagger\,\mathbf{a}(\theta)|^2}{\mathbf{w}^\dagger\,\mathbf{C}\,\mathbf{w}} \end{aligned}$$

From this expression, considering moreover that we want to verify the unit constraint $|\mathbf{w}^\dagger\mathbf{a}(\theta_S)| = 1$, the maximization of the $SINR(\mathbf{w})$ is equivalent to the minimization of $\mathbf{w}^\dagger\,\mathbf{C}\,\mathbf{w}$.

$$\underset{\mathbf{w}}{\arg\max}\quad SINR(\mathbf{w}) = \underset{\mathbf{w}}{\arg\min}\quad \mathbf{w}^\dagger\,\mathbf{C}\,\mathbf{w}$$

1.1.3 Minimum Power Distortionless Response (MPDR) filter expression

In practice, the matrix $\mathbf{C}$ is not available and cannot even be reliably estimated. Therefore, we must work with $\mathbf{R}$, and our objective becomes to maximize the $SSINR$:

$$\underset{\mathbf{w}}{\arg\max}\quad SSINR_{out}(\mathbf{w}) = \frac{\mathbb{E}\big(|\mathbf{w}^\dagger\,\mathbf{a}(\theta)\,s(k)|^2\big)}{\mathbb{E}\big(|\mathbf{w}^\dagger\ (\mathbf{a}(\theta_S)\ s(k)+\mathbf{y}_I(k)+\mathbf{n}(k))|^2\big)} = \frac{P_S\,|\mathbf{w}^\dagger\,\mathbf{a}(\theta)|^2}{\mathbf{w}^\dagger\,\mathbf{R}\,\mathbf{w}}$$

The Minimum Power Distortionless Response filter $MPDR$ satisfies the following expression:

$$\mathbf{w}_{MPDR} = \underset{\mathbf{w}}{\arg\min}\quad \mathbf{w}^\dagger\,\mathbf{R}\,\mathbf{w} \tag{5}$$

We must solve a constrained optimization problem, which can be done by using the Lagrangian of the problem. Let us define the Lagrangian $\mathcal{L}(\lambda,\,\mathbf{w})$ with respect to $\mathbf{w} \in \mathbb{C}^N$ and $\lambda \in \mathbb{C}$ and its associated problem. We ar now searching for $\tilde{\mathbf{w}}$ and $\tilde{\lambda}$ such as:

$$\Big\{\tilde{\mathbf{w}},\,\tilde{\lambda}\Big\} = \underset{\mathbf{w},\,\lambda}{\arg\min}\quad \mathcal{L}(\lambda,\,\mathbf{w}) = \underset{\mathbf{w},\,\lambda}{\arg\min}\quad \mathbf{w}^\dagger\mathbf{R}\,\mathbf{w} + \lambda\,\Big(\mathbf{w}^\dagger\mathbf{a}(\theta_S) - 1\Big) + \lambda^*\,\Big(\mathbf{a}(\theta_S)^\dagger\mathbf{w} - 1\Big)$$

To solve the optimization problem, we compute the derivatives of the Lagrangian:

$$\mathcal{L}(\lambda,\mathbf{w}) = \mathbf{w}^\dagger\mathbf{R}\,\mathbf{w} + \lambda\,\Big(\mathbf{w}^\dagger\mathbf{a}(\theta_S) - 1\Big) + \lambda^*\,\Big(\mathbf{a}(\theta_S)^\dagger\mathbf{w} - 1\Big)$$

Step 1: Directional derivative
For any perturbation $\mathbf{h}$ in $\mathbf{w}$, the first-order (Taylor) expansion gives us:

$$f(\mathbf{w}+\epsilon\mathbf{h}) = f(\mathbf{w}) + \epsilon \left\langle \mathbf{h}, \frac{\partial f}{\partial \mathbf{w}} \right\rangle + o(\epsilon),$$

where $\langle \mathbf{h}, \mathbf{v} \rangle = \mathbf{h}^\dagger \mathbf{v}$ denotes the Hermitian scalar product. We apply this to the two relevant terms:

1. Quadratic term:

$$f_1(\mathbf{w}) = \mathbf{w}^\dagger \mathbf{R} \mathbf{w}$$

Expanding in the direction $\mathbf{h}$:

$$f_1(\mathbf{w}+\epsilon\mathbf{h}) = (\mathbf{w}+\epsilon\mathbf{h})^\dagger \mathbf{R}(\mathbf{w}+\epsilon\mathbf{h}) = \mathbf{w}^\dagger\mathbf{R}\mathbf{w} + \epsilon\mathbf{h}^\dagger\mathbf{R}\mathbf{w} + \epsilon\mathbf{w}^\dagger\mathbf{R}\mathbf{h} + \mathcal{O}(\epsilon^2)$$

The derivative with respect to $\mathbf{w}$ is:

$$\frac{\partial f_1}{\partial \mathbf{w}} = \mathbf{R}\mathbf{w}$$

2. Linear constraint term:

$$f_2(\mathbf{w}) = \lambda^*(\mathbf{a}(\theta_S)^\dagger \mathbf{w} - 1)$$

Directional derivative:

$$f_2(\mathbf{w}+\epsilon\mathbf{h}) = \lambda^*(\mathbf{a}(\theta_S)^\dagger(\mathbf{w}+\epsilon\mathbf{h}) - 1) = f_2(\mathbf{w}) + \epsilon\lambda^*\mathbf{a}(\theta_S)^\dagger\mathbf{h} + \mathcal{O}(\epsilon^2)$$

so that,

$$\frac{\partial f_2}{\partial \mathbf{w}} = \lambda^*\mathbf{a}(\theta_S)$$

Step 2: Derivative of the Lagrangian
Combining the terms, we obtain:

$$\frac{\partial \mathcal{L}(\lambda, \mathbf{w})}{\partial \mathbf{w}} = \mathbf{R}\mathbf{w} + \lambda^*\mathbf{a}(\theta_S).$$

The derivative with respect to the Lagrange multiplier λ is straightforward:

$$\frac{\partial \mathcal{L}(\lambda, \mathbf{w})}{\partial \lambda} = \mathbf{w}^\dagger\mathbf{a}(\theta_S) - 1.$$

Lagrange conditions give us that the problem solutions are $\left\{\tilde{\mathbf{w}},\, \tilde{\lambda}\right\}$ such that :

$$\begin{cases} \left.\dfrac{\partial \mathcal{L}(\lambda,\, \mathbf{w})}{\partial \lambda}\right|_{\lambda=\tilde{\lambda},\, \mathbf{w}=\tilde{\mathbf{w}}} = 0 \\ \left.\dfrac{\partial \mathcal{L}(\lambda,\, \mathbf{w})}{\partial \mathbf{w}}\right|_{\lambda=\tilde{\lambda},\, \mathbf{w}=\tilde{\mathbf{w}}} = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{R}\,\tilde{\mathbf{w}} \,+\, \tilde{\lambda}^*\mathbf{a}(\theta_S) = 0 \\ \tilde{\mathbf{w}}^\dagger\mathbf{a}(\theta_S) - 1 = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \tilde{\mathbf{w}} = -\tilde{\lambda}^*\,\mathbf{R}^{-1}\,\mathbf{a}(\theta_S) \\ -\,\tilde{\lambda}^* = \dfrac{1}{\mathbf{a}(\theta_S)^\dagger\,\mathbf{R}^{-1}\,\mathbf{a}(\theta_S)} \end{cases}$$

By substituting the second equation in the first one, we obtain the $MPDR$ filter solution of Equation 5

$$\mathbf{w}_{MPDR} = \tilde{\mathbf{w}} = \frac{\mathbf{R}^{-1}\,\mathbf{a}(\theta_S)}{\mathbf{a}(\theta_S)^\dagger\,\mathbf{R}^{-1}\,\mathbf{a}(\theta_S)}$$

Now that the expressions of the filters are computed, we can implement the two approaches by assuming the error between the true angle and the estimated angle is equal to 2° so $\widehat{\theta_s} = 22°$, and compare them to the Conventional and Adaptive beamforming :

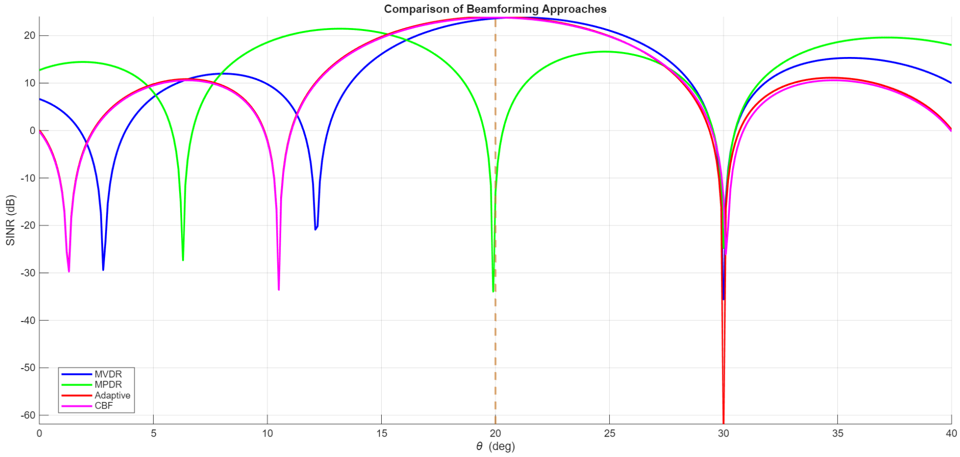


Figure 2: Comparison of Beamforming Approaches

We notice that around $\theta_s = 20°$ the four methods exhibit a strong peak, which corresponds to the direction of the desired signal. This peak represents the main beam of the array response and indicates that each beamforming successfully steers its gain toward the target source. A sharp drop appears around $\theta_i = 30$ which corresponds to the direction of the interfering source. The deeper the drop, the more effective the beamforming is in canceling the interference. Adaptive, MVDR, and MPDR achieve strong interference suppression, whereas the Conventional Beamformer (CBF) produces a much shallower null.

1.2 Unknown covariance matrices

In the second case, we explore a more realistic scenario where the matrices C and R are unknown and must be estimated. To accomplish this, we need to collect multiple snapshots of the received signal, denoted as $\mathbf{Y} = \{\mathbf{y}(k)\}_{k\in[\![1,\,K]\!]}$, at different time instances. Typically, the covariance matrices are estimated using:

$$\widehat{\mathbf{C}} = \frac{1}{K}\sum_{k=1}^{K}\big((\mathbf{y}_I(k)+\mathbf{n}(k))(\mathbf{y}_I(k)+\mathbf{n}(k))^\dagger\big)$$

$$\widehat{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}(k)\,\mathbf{y}(k)^\dagger$$

We are now interested in implementing and analyzing beamforming approaches based on these estimations.

1.2.1 Noise probability density reformulation

We have $\mathbf{n}(k) \sim \mathcal{CN}(0,\,\sigma^2\,\mathbf{I})$ with $\mathcal{CN}$ the complex normal distribution:

$$\frac{1}{\pi^N\,|\boldsymbol{\Sigma}|}\ \exp^{-(\mathbf{n}(k)-\mathbf{m})^\dagger\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{n}(k)-\mathbf{m})} \tag{6}$$

From there, we can apply it specifically for the studied case with $\mathbf{m} = 0$ and $\boldsymbol{\Sigma} = \sigma^2\,\mathbf{I}$

$$p(\mathbf{n}(k)) = \frac{1}{(\pi\,\sigma^2)^N}\ \exp^{-\frac{1}{\sigma^2}\mathbf{n}(k)^\dagger\mathbf{n}(k)}$$

We want to separate the real and imaginary parts of this signal to simplify the problem with $\mathbf{n}(k) = \mathbf{n}_{re}(k) + j\,\mathbf{n}_{im}(k)$ where $\mathbf{n}_{re}(k) \in \mathbb{R}^N$ and $\mathbf{n}_{im}(k) \in \mathbb{R}^N$. Under these assumptions, $\mathbf{n}(k)$ hermitian norm becomes:

$$\mathbf{n}(k)^\dagger\,\mathbf{n}(k) = (\mathbf{n}_{re}(k) - j\,\mathbf{n}_{im}(k))\,(\mathbf{n}_{re}(k) + j\,\mathbf{n}_{im}(k)) = \|\mathbf{n_{re}}(k)\|^2 + \|\mathbf{n}_{im}(k)\|^2$$

We replace :

$$\begin{aligned} p(\mathbf{n}(k)) &= \frac{1}{(\pi\,\sigma^2)^{N/2}}\,\frac{1}{(\pi\,\sigma^2)^{N/2}}\ \exp^{-\frac{1}{\sigma^2}(\|\mathbf{n_{re}}(k)\|^2+\|\mathbf{n}_{im}(k)\|^2)} \\ &= \frac{1}{(2\,\pi\,\frac{\sigma^2}{2})^{N/2}}\exp^{-\frac{1}{2\,\frac{\sigma^2}{2}}\|\mathbf{n_{re}}(k)\|^2} \times \frac{1}{(2\,\pi\,\frac{\sigma^2}{2})^{N/2}}\ \exp^{-\frac{1}{2\,\frac{\sigma^2}{2}}\|\mathbf{n}_{im}(k)\|^2} \end{aligned}$$

The real vectorial Gaussian for $\mathcal{N}(\mathbf{m}_0,\ \sigma_0^2\,\mathbf{I})$ probability density has the following expression:

$$p(\mathbf{x}) = \frac{1}{(2\,\pi\,\sigma_0^2)^{N/2}}\,\exp^{-\frac{1}{2\,\sigma_0^2}\|\mathbf{x}-\mathbf{m}_0\|^2}$$

We arrive at the separation in a product of two real gaussians:

$$p(\mathbf{n}(k)) = \mathbf{n_{re}}(k)\ \mathbf{n_{im}}(k) = \mathcal{N}(0,\ \frac{\sigma^2}{2}\mathbf{I})\ \mathcal{N}(0,\ \frac{\sigma^2}{2}\mathbf{I})$$

We implement the received signal $y(k) \quad \forall k \in [\![1,\ K]\!]$:

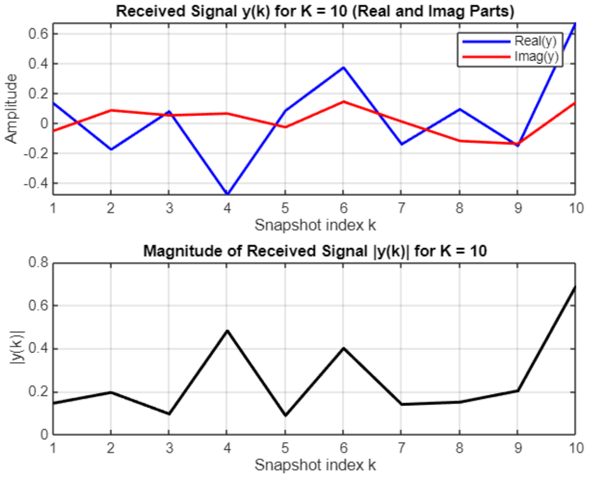


Figure 3: Received signal

This allows us to implement the new $MVDR$ and $MPDR$ filters with the matrices R and C unknown. By computing the errors, we obtain:

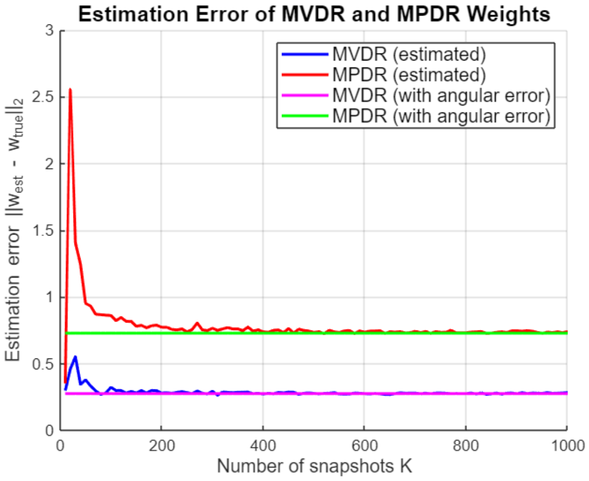


Figure 4: Comparison error of $MVDR$ and $MPDR$

We see that the errors of the new filters converged to the filter's values when R and C were known. Overall, the $MPDR$ filter has higher error and converges more slowly than the $MVDR$ filter. To compare them further, we calculate the $SINR$ of the two methods over 50 Monte-Carlo realizations. For this, we consider R and C known. We obtain:

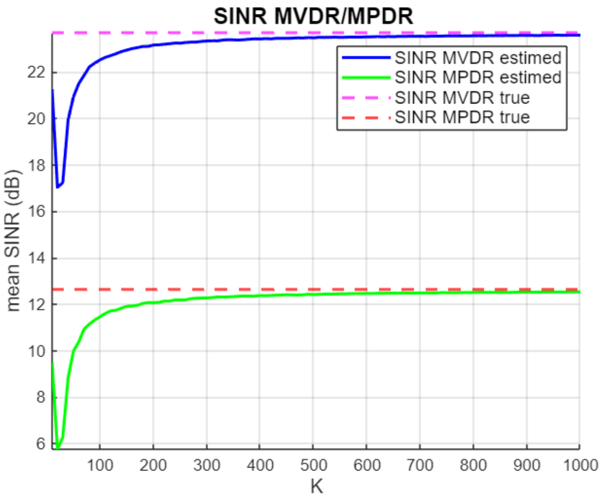


Figure 5: Comparison SINR for MVDR and MPDR

The mean $SINR$ over the 50 realizations for the two methods converge to the real values of $SINR$: as K increases, noise in the sample covariance estimate decreases, so the estimated beamformer weights approach the optimal ones. $MVDR$ reaches a much higher $SINR$ than $MPDR$ and converges faster, confirming its superior interference suppression capability.

This result is expected given the intrinsic structure of the covariance matrices $\mathbf{R}$ and $\mathbf{C}$. While $\mathbf{R}$ contains full signal, interference, and noise information, $\mathbf{C}$ only accounts for interference and noise, which naturally limits the performance of the corresponding beamformer. Although this leads to slower convergence and reduced performance, the $MPDR$ formulation is more realistic in practice, since the signal covariance is generally unknown and difficult to estimate.

1.2.2 Robust MPDR

To address these limitations, we now introduce a robust $MPDR$ formulation, designed to mitigate the impact of model uncertainties and estimation errors while preserving the distortionless response constraint. This approach aims to improve stability and performance in realistic operating conditions.

We implement the Robust MPDR filter, a variant of the MPDR where a diagonal loading factor μ is added to the estimated covariance matrix $\widehat{\mathbf{R}}$, with values $\mu = [0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1]$.

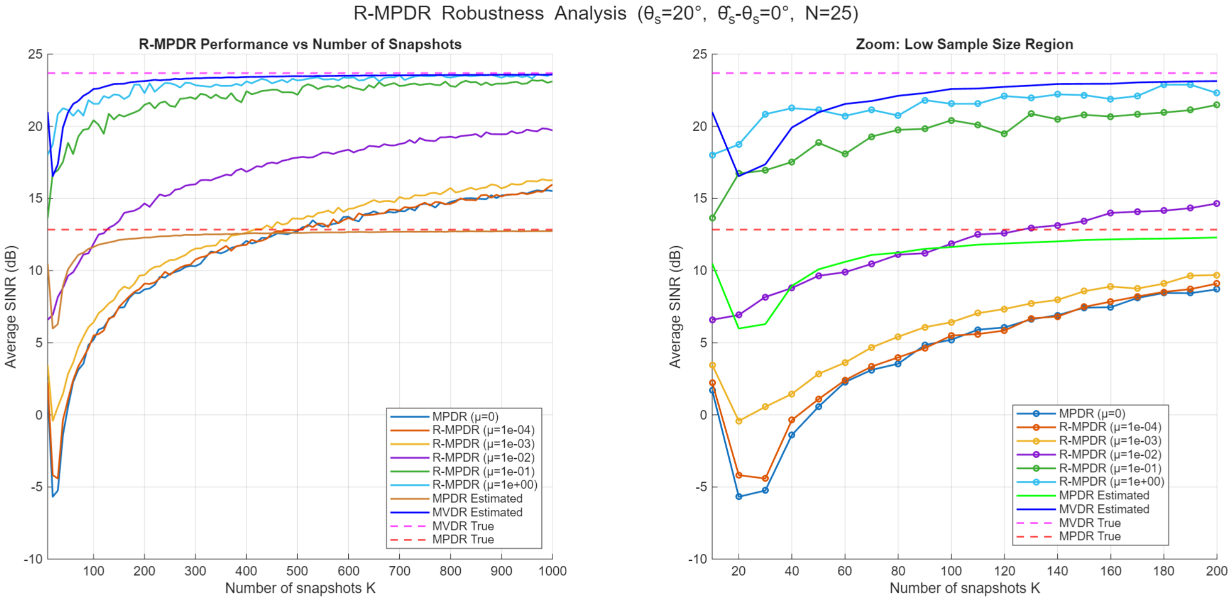


Figure 6: Comparison MVDR, MPDR, and Robust MPDR with θ_S known

When the source direction θ_S is accurately known, this method is particularly effective: as the parameter μ increases, the performance approaches the $MPDR$ SINR, demonstrating near-optimal behavior.

We also consider the more realistic scenario where the direction is estimated with some error ($\widehat{\theta_S}$ assumed to have a 2° deviation) to assess the method's robustness under imperfect knowledge.

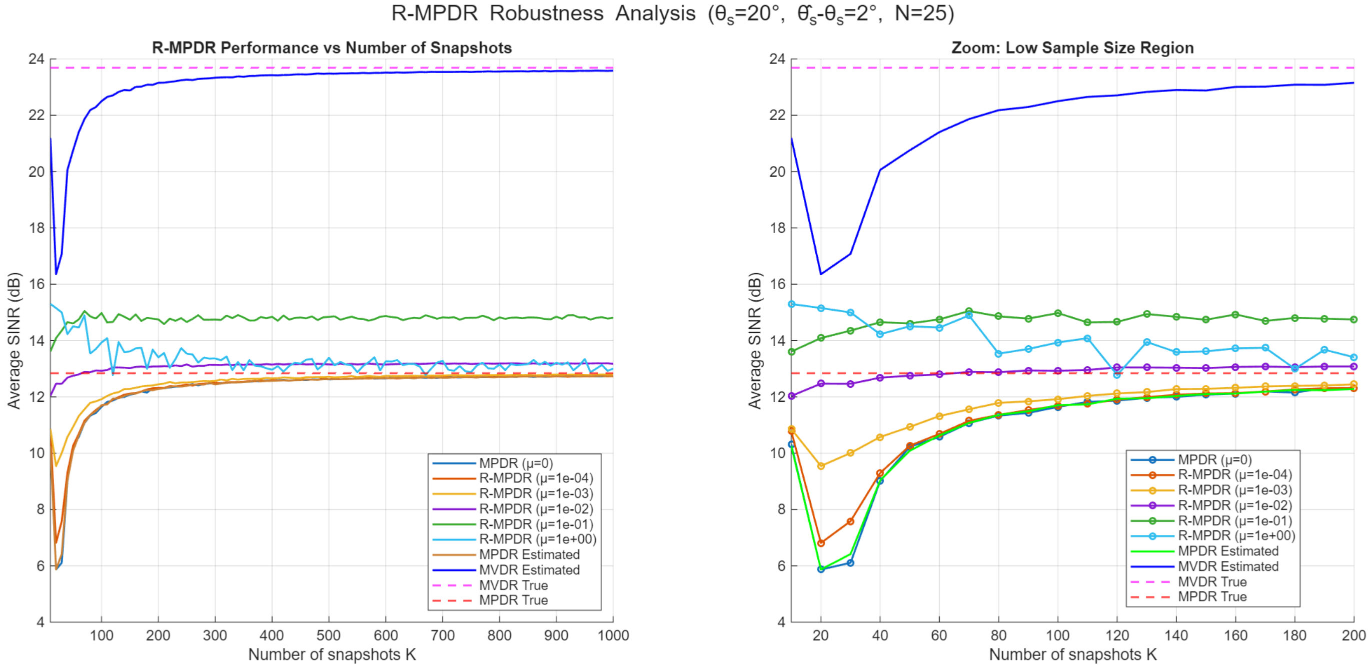


Figure 7: Comparison MVDR, MPDR, and Robust MPDR with θ_S estimated

Here we observe that for $\mu = 0$, the performance corresponds to the classical $MPDR$ filter, as expected. As μ increases, the performance improves: the filter achieves better SINR even with a small number of snapshots K and converges faster. However, when μ is set too high, the SINR can temporarily exceed the ideal $MPDR$ value, but this comes at the cost of stability, leading to increased fluctuations and degraded performance for larger K.

To assess the relevance of different μ values, we compared the absolute μ to the average power of the covariance matrix:

$$\text{Relative percentage} = \frac{\mu}{\text{tr}(\widehat{\mathbf{R}})/N} \times 100.$$

This quantifies the strength of the regularization relative to the mean received power across the array. If μ is set too large ($1e{-}1$ or 1), its contribution dominates with a relative percentage higher than 30%, making all diagonal terms nearly identical. This degrades performance since the algorithm can no longer distinguish differences in the covariance values.

Conversely, if μ is too small ($1e^{-3}$, $1e^{-4}$ and 0), its effect is negligible with a relative percentage lower than 1%, and low (or no) improvement is observed in the beamformer's performance.

The goal is to choose a moderate μ that balances these effects, contributing between a few % and 20% relative to the mean power. This increases the overall diagonal, improves matrix conditioning and stability during inversion, while still preserving enough variation to distinguish between entries. Therefore, to improve MPDR performance, μ values between $1e^{-3}$ and $1e{-}2$ are recommended. Overall, this method helps accelerate convergence and improve the accuracy of the MPDR algorithm.

2 Implementation of DoA estimation techniques

Having studied beamforming techniques and their performance in interference suppression, we now turn to direction-of-arrival (DoA) estimation methods. These approaches are complementary: while beamformers rely on knowledge of the source directions to optimize signal reception, DoA estimation allows us to accurately determine those directions in practice. This is particularly important because, as we have seen, the performance of some beamforming filters is highly sensitive to angle errors, highlighting the need for precise DoA estimation to achieve reliable beamformer operation.In the previous section, the focus was on improving SINR by filtering the received signal at a fixed θ. Our goal now shifts to accurately estimating the source direction θ_S.

In this section, we will implement the various DoA estimation techniques. The primary modification is that we now consider $P = 3$ sources and $\mathbf{P}_S = 10$. To achieve this, we assume a new scenario where the ULA network receives three unknown sources with directions $\boldsymbol{\theta}_S = [\theta_1 = -25°,\, \theta_2 = 20°,\, \theta_3 = 25°]^\top$. The other simulation parameters remain unchanged. As previously, the matrices $\mathbf{C}$ and $\mathbf{R}$ are estimated and K is fixed to 500. Additionally, we assume the presence of interference. Consequently, the signal is modeled by:

$$\mathbf{y}(k) = \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k) + \mathbf{y}_I(k) + \mathbf{n}(k) \qquad \mathbf{n}(k) \sim \mathcal{CN}(0, \sigma^2\,\mathbf{I}) \tag{7}$$

with $\mathbf{y}(k) \in \mathbb{C}^N$, the steering matrix $\mathbf{A}(\boldsymbol{\theta}_S) = [\mathbf{a}(\theta_1),\, \mathbf{a}(\theta_2),\, \mathbf{a}(\theta_3)] \in \mathcal{M}_{N,P}(\mathbb{C})$ and $\forall k \in [\![1,\, K]\!]$ $\mathbf{s}(k) \in \mathbb{C}^3$.

DoA estimation methods can be classified into parametric approaches and non-parametric approaches.

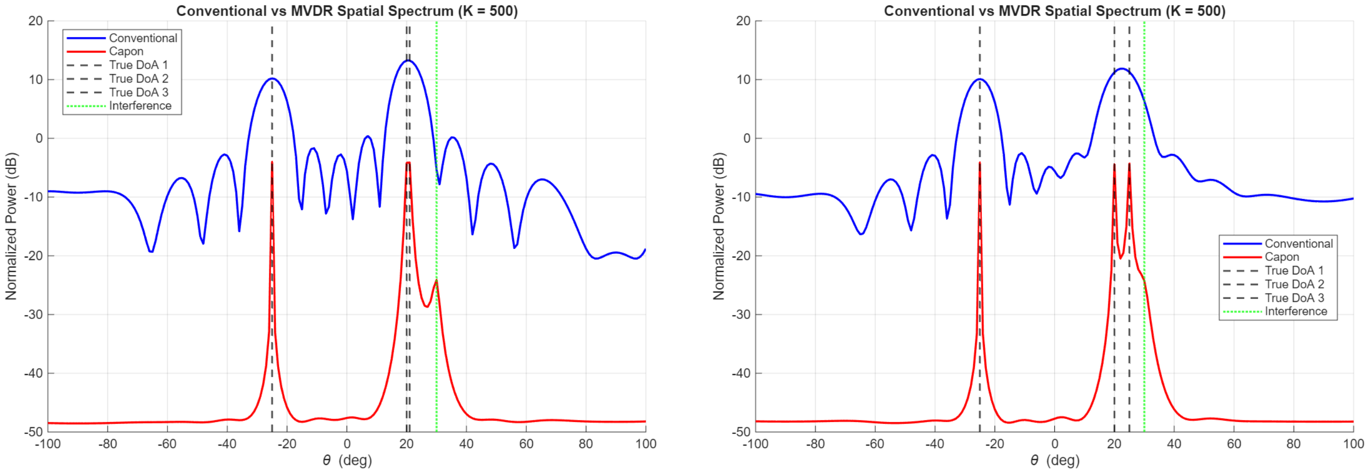
2.1 Implementation of non-parametric approaches

Non-parametric methods do not assume a specific model for the source signals: they work directly with the output power received after beamforming. We start by implementing the two methods based on the conventional $\mathbf{w}_{CBF}$ and adaptive beamforming with Capon's method $\widehat{\mathbf{w}}_{MPDR}$, with interferences.

Although the DoA estimation methods are not performing the same task as beamforming, they are based on the same underlying equations demonstrated in the previous beamforming section. In both the conventional (Bartlett) and Capon (MVDR) approaches, the steering vectors and covariance matrices introduced for beamforming are used to evaluate the spatial power distribution as a function of angle. To estimate the directions of arrival, we compute the power spectrum for all candidate angles $\theta \in [0,\, 2\pi]$:

$$P_{Conv.}(\theta) = \mathbf{a}(\theta)^\dagger \widehat{\mathbf{R}} \mathbf{a}(\theta) \qquad \text{and} \qquad P_{Capon}(\theta) = \frac{1}{\mathbf{a}(\theta)^\dagger \widehat{\mathbf{R}}^{-1} \mathbf{a}(\theta)}$$

and then identify the peaks of $P(\theta)$. These peaks indicate the estimated source directions $\hat{\theta}_S$, achieving the goal of accurate DoA estimation using the same structures introduced for beamforming. We obtain:



(a) For $\theta_3 = 21$

(b) For $\theta_3 = 25$

Figure 8: Conventional and Capon power spectrum comparison

We can see that both methods show spectral peaks at the true angles of arrival, indicated by the dashed vertical lines. However, the Capon spectrum exhibits much sharper and narrower peaks than the conventional method. This means it is more effective at separating closely spaced sources.

The conventional beamformer has large side lobes while the $MVDR$ suppresses them, dropping more than 40 dB below the main peaks. The $MVDR$ spectrum includes deep nulls at the positions of interferers, unlike the conventional method. This is a signature of adaptive beamforming: $MVDR$ actively cancels interference while preserving the desired signal.When the third source is $\theta_3 = 21°$, we see that the second and third DoA are not accurately separated. The methods are not precise enough.

2.2 Implementation of parametric approaches

Parametric approaches model the signal as a sum of a finite number of sources and exploit this structure.

2.2.1 MUSIC method

In the MUSIC method, we assume a structured form for the signal covariance matrix,

$$\mathbf{R} = \mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}(\theta)^\dagger + \sigma^2\mathbf{I}.$$

Although this does not model the signal samples directly, it incorporates prior knowledge about the signal subspace, which is why MUSIC is considered a parametric DoA estimation method.

Power expression with the MUSIC approach

Both the conventional and Capon methods compute the spatial power directly from the beamformer output. We now aim to improve upon this by exploiting the classical decomposition of the received signal into signal and interference-plus-noise components. This approach provides a more accurate estimate of the source power and is widely applicable across different communication scenarios.

We decompose $\mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}(\theta)^\dagger$ into an orthonormal basis of eigenvectors $\{\mathbf{u}_i\}_{i\in[\![1;N]\!]}$. $\mathbf{U}$ composed of all $\mathbf{u}_i$ to take advantages of the orthonormal properties by keeping the information contained in the covariance matrix, then gives the following equation :

$$\mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}(\theta)^\dagger = \mathbf{U}\,\Lambda\,\mathbf{U}^\dagger$$

where Λ is the diagonal matrix of the eigenvalues λ_i associated with the eigenvectors $\mathbf{u}_i$. The matrix $\mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}(\theta)^\dagger$ is of rank P, which means that (in descending order) the P first eigenvalues are non-zero, and the $N-P$ others are null. By separating $\mathbf{U}$ into two separate parts $\mathbf{U}_S = [\mathbf{u}_1, \cdots, \mathbf{u}_P]$ and $\mathbf{U}_N = [\mathbf{u}_{P+1}, \cdots, \mathbf{u}_N]$, we have :

$$\mathbf{A}(\theta)\mathbf{R}_S\mathbf{A}(\theta)^\dagger = \sum_{p=1}^{P} \lambda_p\,\mathbf{u}_p\mathbf{u}_p^\dagger = \mathbf{U}_S\,\Lambda_S\,\mathbf{U}_S^\dagger$$

And then, the orthogonality of the matrix U_N (from the SVD definition) yields to :

$$\mathbf{R} = \mathbf{U}_S\,\Lambda_S\,\mathbf{U}_S^\dagger + \sigma^2\mathbf{U}_N\,\mathbf{U}_N^\dagger \tag{8}$$

To compute the estimated power, subsubsection 2.2.1 gives us that both $\mathbf{a}(\theta_S)$ and $\mathbf{U}_S$ generate the same space, meaning they are different bases of the same space. Then, as we know $\mathbf{U}_S \perp \mathbf{U}_N$, we obtain $\mathbf{a}(\theta_S) \perp \mathbf{U}_N$. We also have $\mathbf{a}(\theta_S)^\dagger \cdot \mathbf{U}_N = 0$. The power of a signal is the sum of the energies of the studied signal. Here, in the considered subspace, it can be written as $\mathbf{a}(\theta_S)^\dagger\,\mathbf{U}_N\,\mathbf{U}_N\,\mathbf{a}(\theta_S)^\dagger$. We want :

$$P_{MUSIC}(\theta) = \frac{1}{\mathbf{a}(\theta_S)^\dagger\;\mathbf{U}_S\mathbf{U}_S^\dagger\;\mathbf{a}(\theta_S)}$$

From our assumptions, we determine $\widehat{\mathbf{U}}_S$ from $\mathbf{R}$. In reality, we do not have access to $\mathbf{R}$ so we have to estimate it from a distribution. Meaning $\mathbf{R}$ becomes $\widehat{\mathbf{R}}$ and $\mathbf{U}_S$ becomes $\widehat{\mathbf{U}}_S$:

$$P_{MUSIC}(\theta) = \frac{1}{\mathbf{a}(\theta_S)^\dagger\;\widehat{\mathbf{U}}_S\widehat{\mathbf{U}}_S^\dagger\;\mathbf{a}(\theta_S)}$$

We implement this method with interferences. We set θ between $[-100°, +100°]$.

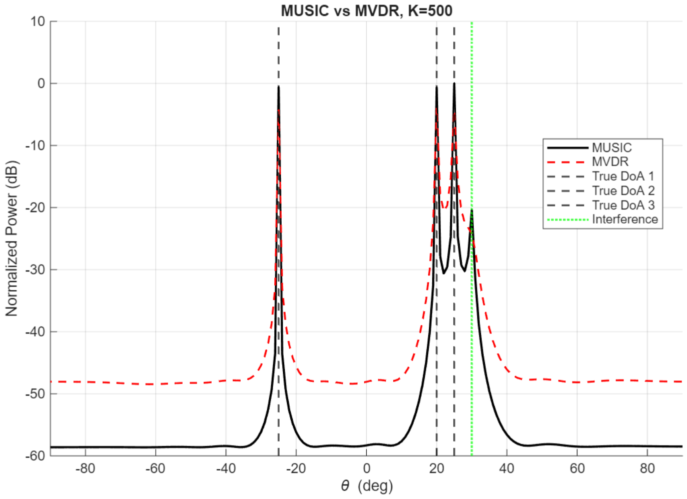


Figure 9: Comparison Music and Capon

$MUSIC$ shows very sharp, narrow peaks aligned with the true DoAs. This is expected because $MUSIC$ exploits the orthogonality between the noise subspace and the array steering vectors. As a result, $MUSIC$ can separate closely spaced sources even when conventional or $MVDR$ beamforming struggles. On the other hand, $MVDR$ also shows peaks at the correct DoAs, but the peaks are wider, less pronounced, and sometimes slightly shifted depending on estimation noise. Overall, $MUSIC$ exhibits stronger discrimination capability.

When interferences are added, $MVDR$ suppresses the response near the interference DoA, effectively placing a null while maintaining gain toward the desired direction. $MUSIC$, however, does not actively null interferences, highlighting that it is primarily a detection or estimation technique, whereas $MVDR$ also functions as a beamforming or filtering method.

In practical scenarios, the number of sources is not always known in advance. To address this, we implemented model-order selection criteria such as the Akaike Information Criterion (AIC) and the Minimum Description Length (MDL), which estimate the number of sources based on the eigenstructure of the covariance matrix. A peak-finding function was also used to rank the estimated DoAs from highest to lowest power. Table 1 summarizes the estimated DoAs for the different methods under both known and MDL-estimated numbers of sources. The last column indicates the interference detected when the number of sources is estimated using MDL. Since the true signal sources generally have higher power than the interference, the peak-finding algorithm ranks the interference last when ordering the detected directions in descending power.

Table 1: Estimated DoAs for different methods and cases

	$\theta_3 = 25$				$\theta_3 = 21$			
Method	θ_1	θ_2	θ_3	θ_i	θ_1	θ_2	θ_3	θ_i
True DoA	-25	20	25	30	-25	20	21	30
Conventional DoA	-25	8	22	-11	-25	7	21	35
Capon DoA	-25	20	25	50	-25	-10	21	30
MUSIC DoA	-25	20	25	30	-25	2	21	30

From the table, we observe that when the sources are sufficiently separated, Capon's and MUSIC DoA methods are able to detect the main true sources ($[-25, 20, 25]$) reasonably well. Only the conventional method has already too large lobes so that it cannot distinguish 20 and 25. However, only $MUSIC$ detects the interference (peak at 30°) when the number of sources is estimated using MDL, while conventional and Capon methods miss it. When some sources are closer together ($[-25, 21, 25]$), the three performs similarly and we notice that θ_2 and θ_3 cannot be separated. Overall, $MUSIC$ exhibits the strongest ability to detect both main and weaker or additional signals.

2.2.2 Maximum Likelihood method

Now, we propose to implement the parametric approach based on the maximization of the likelihood so that:

$$\boldsymbol{\theta}_S = \arg\max_{\boldsymbol{\theta}_S} \quad p(\mathbf{Y}|\boldsymbol{\theta}_S) \qquad \mathbf{Y} = \{\mathbf{y}(1), \cdots, \mathbf{y}(K)\} \tag{9}$$

Particularly, we propose to study the deterministic likelihood approach, which assumes that $s(k)$ is not random.

1. When s and σ^2 are known

Theorical estimator expression

From Equation 7, when $\mathbf{s}(k)$ modeling is known :

$$\mathbf{y}(k) = \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k) + \mathbf{n}(k) \qquad \mathbf{n}(k) \sim \mathcal{CN}(0,\ \sigma^2\,\mathbf{I}) \qquad \Leftrightarrow \qquad \mathbf{y}(k) \sim \mathcal{CN}(\mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k),\ \sigma^2\,\mathbf{I})$$

The equation Equation 9 becomes :

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{S,\,ML1} &= \arg\max_{\boldsymbol{\theta}_S} \quad p(\mathbf{Y}|\boldsymbol{\theta}_S) = \arg\min_{\boldsymbol{\theta}_S} \quad -\ln p(\mathbf{Y}|\boldsymbol{\theta}_S)\\ &= \arg\min_{\boldsymbol{\theta}_S} \quad -\ln\Big(\prod_{k=1}^{K} p(\mathbf{y}(k)|\boldsymbol{\theta}_S)\Big)\end{aligned}$$

Supposing independence and replacing with the complex Gaussian from Equation 6 :

$$\widehat{\boldsymbol{\theta}}_{S,\,ML1} = \arg\min_{\boldsymbol{\theta}_S} \quad -\ln\prod_{k=1}^{K}\Big(\frac{1}{\pi^N\,|\boldsymbol{\Sigma}|}\ \exp^{-\frac{1}{\sigma^2}(\mathbf{y}(k)-\mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))^\dagger\,(\mathbf{y}(k)-\mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))}\Big)$$

All terms independent in $\boldsymbol{\theta}_S$ do not influence the arg min, they are only a constant with respect to this problem. Let's develop by simplifying them :

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{S,\,ML1} &= \arg\min_{\boldsymbol{\theta}_S} \quad -\sum_{k=1}^{K}\Big(-\frac{1}{\sigma^2}(\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))^\dagger\,(\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))\Big)\\ &= \arg\min_{\boldsymbol{\theta}_S} \quad \sum_{k=1}^{K}\Big((\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))^\dagger\,(\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k))\Big)\\ &= \arg\min_{\boldsymbol{\theta}_S} \quad \sum_{k=1}^{K}\|\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)\|^2\end{aligned}$$

When s is known, this expression can be computed directly because $\mathbf{y}(k)$ is the received signal, $A(\theta)$ is by design. A Gauss-Newton algorithm can solve for our only unknown θ_S.

Cramér-Rao Bound expression

To find the expression of the *CRB*, we use the Slepian Bangs formula. The general expression of the Fisher information matrix terms $\forall l \in [\![1,\,P]\!], \forall l' \in [\![1,\,P]\!]$ considering $\mathbf{y}(k) \sim \mathcal{CN}\Big(\boldsymbol{\mu}_k(x), \boldsymbol{\Sigma}(x)\Big)$ is :

$$[\mathbf{I}(x)]_{l,l'} = 2\sum_{k=1}^{K}\mathrm{Re}\left(\frac{\partial\boldsymbol{\mu}_k(x)}{\partial(x)_l}^\dagger\,\boldsymbol{\Sigma}(x)^{-1}\,\frac{\partial\boldsymbol{\mu}_k(x)}{\partial(x)_{l'}}\right) + K\,\mathrm{tr}\left(\boldsymbol{\Sigma}(x)^{-1}\frac{\partial\boldsymbol{\Sigma}(x)}{\partial(x)_l}\boldsymbol{\Sigma}(x)^{-1}\frac{\partial\boldsymbol{\Sigma}(x)}{\partial(x)_{l'}}\right)$$

In our case, we have $\boldsymbol{\mu}_k(\boldsymbol{\theta}) = \mathbf{A}(\boldsymbol{\theta}_S)\,\mathbf{s}(k)$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \sigma^2\,\mathbf{I}$ independent from $\boldsymbol{\theta}$ (derivatives are 0), so that it simplifies to the first (mean) term only:

$$[\mathbf{I}(\boldsymbol{\theta}_S)]_{l,l'} = \frac{2}{\sigma^2}\sum_{k=1}^{K}\mathrm{Re}\left(\frac{\partial\boldsymbol{\mu}_k(\boldsymbol{\theta}_S)}{\partial(\boldsymbol{\theta}_S)_l}^\dagger\,\frac{\partial\boldsymbol{\mu}_k(\boldsymbol{\theta}_S)}{\partial(\boldsymbol{\theta}_S)_{l'}}\right)$$

Using $\boldsymbol{\mu}_k(\boldsymbol{\theta}_S) = \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)$,we obtain

$$\frac{\partial \boldsymbol{\mu}_k(\boldsymbol{\theta}_S)}{\partial (\boldsymbol{\theta}_S)_l} = \frac{\partial \mathbf{A}(\boldsymbol{\theta}_S)}{\partial (\boldsymbol{\theta}_S)_l}\mathbf{s}(k) = \mathbf{DA}_l\,\mathbf{s}(k)$$

so that the Fisher Information Matrix entries become:

$$[\mathbf{I}(\boldsymbol{\theta}_S)]_{l,l'} = \frac{2}{\sigma^2}\sum_{k=1}^{K}\mathrm{Re}\Big(\mathbf{s}(k)^\dagger\,\mathbf{DA}_l^\dagger\,\mathbf{DA}_{l'}\,\mathbf{s}(k)\Big)\,.$$

Finally, the Cramér–Rao bound for this case is given by:

$$\mathrm{CRB}(\boldsymbol{\theta}_S) = \mathbf{I}(\boldsymbol{\theta}_S)^{-1}.$$

2. When s and σ^2 are unknown

Theorical estimator expression

When neither s nor σ^2 are known, the terms with σ^2 that we integrated in the constant for the development above now have to be considered. By developing from Equation 2.2.2 like above, we end up with:

$$\widehat{\boldsymbol{\theta}}_{S,\,ML2} = \underset{\boldsymbol{\theta}_S,\,\sigma^2,\,\mathbf{s}(k)}{\arg\min}\quad -K\ln\Big(\frac{1}{\pi^N\,|\boldsymbol{\Sigma}|}\Big) + \frac{1}{\sigma^2}\sum_{k=1}^{K}\|\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)\|^2$$

With $\boldsymbol{\Sigma} = \sigma^2\,\mathbf{I}$, we have $|\boldsymbol{\Sigma}| = \sigma^{2\,N}$, and the term in π is still a constant for the minimization:

$$\widehat{\boldsymbol{\theta}}_{S,\,ML2} = \underset{\boldsymbol{\theta}_S,\,\sigma^2,\,\mathbf{s}(k)}{\arg\min}\quad KN\ln\big(\sigma^2\big) + \frac{1}{\sigma^2}\sum_{k=1}^{K}\|\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)\|^2 \tag{10}$$

First, we minimize with respect to $\mathbf{s}(k)$ is a classical linear least–squares problem, so it can be expressed easily:

$$\widehat{\mathbf{s}}(k) = \underset{\mathbf{s}(k)}{\arg\min}\quad \|\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)\|^2 = \underset{\mathbf{s}(k)}{\arg\min}\quad (\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k))^\dagger\,(\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k))$$

The normal equation directly gives us:

$$\widehat{\mathbf{s}}(k) = \big(\mathbf{A}(\boldsymbol{\theta}_S)^\dagger\mathbf{A}(\boldsymbol{\theta}_S)\big)^{-1}\mathbf{A}(\boldsymbol{\theta}_S)^\dagger\,\mathbf{y}(k)$$

This estimator is the unique vector whose image through $\mathbf{A}(\boldsymbol{\theta}_S)$ is the orthogonal projection of $\mathbf{y}(k)$ onto the column space of $\mathbf{A}(\boldsymbol{\theta}_S)$. This defines the orthogonal projector and complement:

$$\mathbf{P}_{\mathbf{A}}(\boldsymbol{\theta}_S) = \mathbf{A}(\boldsymbol{\theta}_S)\,\big(\mathbf{A}(\boldsymbol{\theta}_S)^\dagger\mathbf{A}(\boldsymbol{\theta}_S)\big)^{-1}\,\mathbf{A}(\boldsymbol{\theta}_S)^\dagger \qquad \text{and} \qquad \mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\;= \mathbf{I} - \mathbf{P}_{\mathbf{A}}(\boldsymbol{\theta}_S)$$

With these defined, the vector fitted to this particular subspace becomes $\widehat{\mathbf{y}}(k) = \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k) = \mathbf{P}_{\mathbf{A}}(\boldsymbol{\theta}_S)\mathbf{y}(k)$. Thus, the solution becomes:

$$\underset{\mathbf{s}(k)}{\arg\min}\quad \|\mathbf{y}(k) - \mathbf{A}(\boldsymbol{\theta}_S)\mathbf{s}(k)\|^2 = \mathbf{y}(k)^\dagger\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)$$

After projecting out the signal subspace, the log-likelihood from Equation 10 becomes independent from $\mathbf{s}(k)$:

$$\widehat{\boldsymbol{\theta}}_{S,\,ML2} = \underset{\boldsymbol{\theta}_S,\,\sigma^2}{\arg\min}\quad KN\ln\big(\sigma^2\big) + \frac{1}{\sigma^2}\sum_{k=1}^{K}\mathbf{y}(k)^\dagger\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)$$

With respect to σ^2 it takes the form:

$$f(\sigma^2) = A\ln(\sigma^2) + \frac{B}{\sigma^2} \quad \text{with} \quad A = KN, \quad B = \sum_{k=1}^{K}\mathbf{y}(k)^\dagger\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)$$

Differentiating with respect to σ^2 and setting the derivative to zero gives

$$\frac{\partial f}{\partial \sigma^2} = \frac{A}{\sigma^2} - \frac{B}{(\sigma^2)^2} = 0 \quad \Longrightarrow \quad \widehat{\sigma}^2 = \frac{B}{A}$$

Substituting this expression back into the likelihood removes all terms independent of $\boldsymbol{\theta}_S$. Therefore, the maximum likelihood estimator of $\boldsymbol{\theta}_S$ is obtained by minimizing:

$$\widehat{\boldsymbol{\theta}}_{S,\,ML2} = \underset{\boldsymbol{\theta}_S}{\arg\min} \quad \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}(k)^\dagger \mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)$$

We use that $\mathbf{y}(k)^\dagger\,\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k) \in \mathbb{C}$ to replace with $\mathbf{y}(k)^\dagger\,\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k) = \mathrm{tr}(\mathbf{y}(k)^\dagger \mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k))$, and as tr is a circular operator, the equality extends to $= \mathrm{tr}(\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)\,\mathbf{y}(k)^\dagger)$. We then have by linearity of both sum and tr :

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}_{S,\,ML2} &= \underset{\boldsymbol{\theta}_S}{\arg\min} \quad \frac{1}{K}\sum_{k=1}^{K} \mathrm{tr}\Big(\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\mathbf{y}(k)\,\mathbf{y}(k)^\dagger\Big) \\
&= \underset{\boldsymbol{\theta}_S}{\arg\min} \quad \mathrm{tr}\Big(\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\underbrace{\frac{1}{K}\sum_{k=1}^{K}\mathbf{y}(k)\,\mathbf{y}(k)^\dagger}_{\widehat{\mathbf{R}}}\Big) \\
\widehat{\boldsymbol{\theta}}_{S,\,ML2} &= \underset{\boldsymbol{\theta}_S}{\arg\min} \quad \mathrm{tr}\Big(\mathbf{P}_A^\perp(\boldsymbol{\theta}_S)\,\widehat{\mathbf{R}}\Big)
\end{aligned}$$

with $\mathbf{P}_A^\perp(\boldsymbol{\theta}_S) = \mathbf{I} - \mathbf{A}(\boldsymbol{\theta}_S)\big(\mathbf{A}(\boldsymbol{\theta}_S)^\dagger\,\mathbf{A}(\boldsymbol{\theta}_S)\big)^{-1}\mathbf{A}(\boldsymbol{\theta}_S)^\dagger$

Cramér-Rao Bound expression

To find the expression of the *CRB*, we use the Slepian Bangs formula. The general expression of the Fisher information matrix terms $\forall l \in [\![1,\,P]\!], \forall l' \in [\![1,\,P]\!]$ considering $\mathbf{y}(k) \sim \mathcal{CN}\Big(\boldsymbol{\mu}_k(x), \boldsymbol{\Sigma}(x)\Big)$ is :

$$[\mathbf{I}(x)]_{l,l'} = 2\sum_{k=1}^{K}\mathrm{Re}\left(\frac{\partial \boldsymbol{\mu}_k(x)}{\partial (x)_l}^\dagger\,\boldsymbol{\Sigma}(x)^{-1}\,\frac{\partial \boldsymbol{\mu}_k(x)}{\partial (x)_{l'}}\right) + K\,\mathrm{tr}\left(\boldsymbol{\Sigma}(x)^{-1}\frac{\partial \boldsymbol{\Sigma}(x)}{\partial (x)_l}\boldsymbol{\Sigma}(x)^{-1}\frac{\partial \boldsymbol{\Sigma}(x)}{\partial (x)_{l'}}\right)$$

In our case, we have $\boldsymbol{\mu}_k(\boldsymbol{\theta}) = 0$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \mathbf{R}$, so that it simplifies to:

$$[\mathbf{I}(\boldsymbol{\theta})]_{l,l'} = K\,\mathrm{tr}\left(\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial (\boldsymbol{\theta})_l}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial (\boldsymbol{\theta})_{l'}}\right)$$

We need to compute $\mathbf{R}$ derivative for each $l \in [\![1,\,P]\!]$ and $l' \in [\![1,\,P]\!]$. Let l be in $[\![1,\,P]\!]$:

$$\frac{\partial \mathbf{R}}{\partial (\boldsymbol{\theta})_l} = \frac{\partial \mathbf{A}(\boldsymbol{\theta})\,\mathbf{R}_S\,\mathbf{A}(\boldsymbol{\theta})^\dagger + \sigma^2\,\mathbf{I}}{\partial (\boldsymbol{\theta})_l} = \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial (\boldsymbol{\theta})_l}\,\mathbf{R}_S\,\mathbf{A}(\boldsymbol{\theta})^\dagger + \mathbf{A}(\boldsymbol{\theta})\,\mathbf{R}_S\,\frac{\partial \mathbf{A}(\boldsymbol{\theta})^\dagger}{\partial (\boldsymbol{\theta})_l} \tag{11}$$

Now, let's decompose the matrix derivative of $\mathbf{A}(\boldsymbol{\theta}) \in \mathcal{M}_{N,P}(\mathbb{C})$ for our l index :

$$\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial (\boldsymbol{\theta})_l} = \left[\frac{\partial \mathbf{a}(\theta_1)}{\partial (\boldsymbol{\theta})_l}, \frac{\partial \mathbf{a}(\theta_2)}{\partial (\boldsymbol{\theta})_l}, \ldots, \frac{\partial \mathbf{a}(\theta_P)}{\partial (\boldsymbol{\theta})_l}\right] \tag{12}$$

Let k be in $[\![1;P]\!]$. If $k \neq l$, $\frac{\partial \mathbf{a}(\theta_k)}{\partial (\boldsymbol{\theta})_l} = 0$ and if $k = l$, $\frac{\partial \mathbf{a}(\theta_k)}{\partial (\boldsymbol{\theta})_l} = \frac{\partial \mathbf{a}(\theta_l)}{\partial (\boldsymbol{\theta})_l}$. Now, let's decompose the derivative of $\mathbf{a}(\theta_l) \in \mathbb{C}^N$, with respect to $(\boldsymbol{\theta})_l$. From Equation 1 as it's a ULA network with N antennas:

$$\mathbf{a}(\theta_l) = \{\exp^{-2\pi j(n-1)\frac{d}{\lambda}\sin(\theta_l)}\}_{n \in [\![1;N]\!]} = \begin{pmatrix} 1 \\ \vdots \\ \exp^{-2\pi j(n-1)\frac{d}{\lambda}\sin(\theta_l)} \\ \vdots \\ \exp^{-2\pi j(N-1)\frac{d}{\lambda}\sin(\theta_l)} \end{pmatrix}$$

Meaning that the vectorial derivative of $\mathbf{a}(\theta_l)$ is:

$$\frac{\partial \mathbf{a}(\theta_l)}{\partial(\boldsymbol{\theta})_l} = \frac{\partial}{\partial(\boldsymbol{\theta})_l}\begin{pmatrix} 1 \\ \vdots \\ \exp^{-2\pi j(n-1)\frac{d}{\lambda}\sin(\theta_l)} \\ \vdots \\ \exp^{-2\pi j(N-1)\frac{d}{\lambda}\sin(\theta_l)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 2\pi j(n-1)\frac{d}{\lambda}\cos(\theta_l)\exp^{-2\pi j(n-1)\frac{d}{\lambda}\sin(\theta_l)} \\ \vdots \\ 2\pi j(N-1)\frac{d}{\lambda}\cos(\theta_l)\exp^{-2\pi j(N-1)\frac{d}{\lambda}\sin(\theta_l)} \end{pmatrix}$$

Put that back in Equation 12. We define $[\mathbf{DA}_l] \in \mathcal{M}_{N,P}(\mathbb{C})$ such as:

$$[\mathbf{DA}_l] := \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial(\boldsymbol{\theta})_l} = \begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 2\pi j(n-1)\frac{d}{\lambda}\cos(\theta_l)\exp^{-2\pi j(n-1)\frac{d}{\lambda}\sin(\theta_l)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \underbrace{2\pi j(N-1)\frac{d}{\lambda}\cos(\theta_l)\exp^{-2\pi j(N-1)\frac{d}{\lambda}\sin(\theta_l)}}_{\textit{column } l} & \cdots & 0 \end{pmatrix}$$

Then once this is computed, we can express $[\mathbf{DR}_l] \in \mathcal{M}_{N,P}(\mathbb{C})$ when substituting in Equation 11:

$$[\mathbf{DR}_l] := \mathbf{DA}_l\,\mathbf{R}_S\,\mathbf{A}(\boldsymbol{\theta})^\dagger + \mathbf{A}(\boldsymbol{\theta})\,\mathbf{R}_S\,\mathbf{DA}_l^\dagger$$

With the same expression for l' the derivative. We finally have $\forall l \in [\![1,\, P]\!], \forall l' \in [\![1,\, P]\!]$:

$$[\mathbf{I}(\boldsymbol{\theta})]_{l,l'} = K\,\mathrm{Tr}\big(\mathbf{R}^{-1}\,[\mathbf{DR}_l]\,\mathbf{R}^{-1}\,[\mathbf{DR}_{l'}]\big)$$

Let $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_P]^T$ be the parameter vector. The Fisher Information Matrix is computed element-wise as follows.

For all $(l, l') \in [\![1, P]\!]^2$:

- Computation of the Jacobian matrices for A :

$$[\mathbf{DA}_l] \;=\; \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_l}, \qquad [\mathbf{DA}_{l'}] \;=\; \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_{l'}}.$$

- Computation of the Jacobian matrices for R :

$$[\mathbf{DR}_l] \;=\; \mathbf{DA}_l\,\mathbf{R}_S\,\mathbf{A}(\boldsymbol{\theta})^\dagger + \mathbf{A}(\boldsymbol{\theta})\,\mathbf{R}_S\,\mathbf{DA}_l^\dagger$$

$$[\mathbf{DR}_{l'}] \;=\; [\mathbf{DA}_{l'}]\,\mathbf{R}_S\,\mathbf{A}(\boldsymbol{\theta})^\dagger + \mathbf{A}(\boldsymbol{\theta})\,\mathbf{R}_S\,[\mathbf{DA}_{l'}]^\dagger$$

- From the (l, l') entry of the Fisher matrix:

$$[\mathbf{I}(\boldsymbol{\theta})]_{l,l'} = K\,\mathrm{Tr}\big(\mathbf{R}^{-1}\,[\mathbf{DR}_l]\,\mathbf{R}^{-1}\,[\mathbf{DR}_{l'}]\big)$$

Once the Fisher Information Matrix $\mathbf{I}(\boldsymbol{\theta})$ is completed, the Cramér–Rao Bound is obtained from its inverse:

$$CRB(\boldsymbol{\theta}) = \mathrm{tr}\big(\mathbf{I}(\boldsymbol{\theta})^{-1}\big)$$

Implementation and Comparison

We implemented the Maximum Likelihood (ML) DoA estimator for two scenarios: when the source signals are *deterministic* (known) and when they are *stochastic* (unknown). For deterministic signals, the Cramér-Rao Bound (CRB) is identical across sources, as the only limiting factor is additive noise. For stochastic signals, the CRB is higher, especially for closely spaced sources, reflecting the increased difficulty in separating correlated steering vectors.

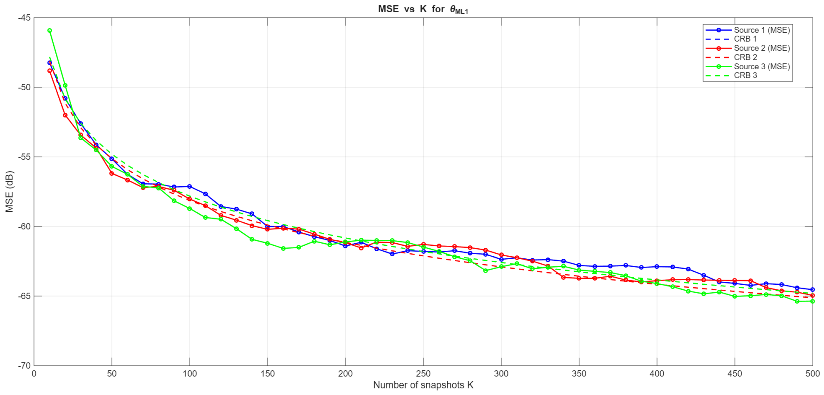


Figure 10: MSE for ML estimation of three sources in the deterministic case

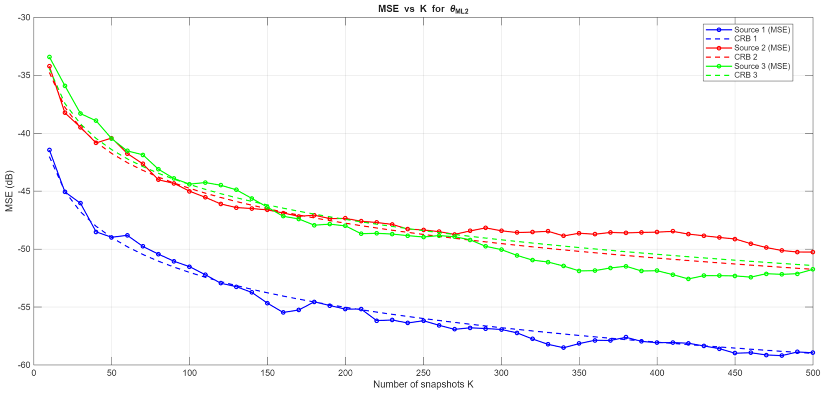


Figure 11: MSE for ML estimation of three sources in the stochastic case

Monte Carlo simulations confirm that ML is highly sensitive to initialization and noise. For deterministic signals, convergence is generally achieved even with moderate deviations from the true angles, and estimation errors closely follow the CRB. For stochastic signals, convergence is more challenging: small errors in the initial guess can prevent the algorithm from reaching the global minimum, and only accurate initializations or low-noise conditions yield correct estimates.

To improve convergence, we use MUSIC to provide initial DoA estimates and compare ML behavior with three initialization strategies: far from the true DoAs, close to the true DoAs, and MUSIC-based estimates.

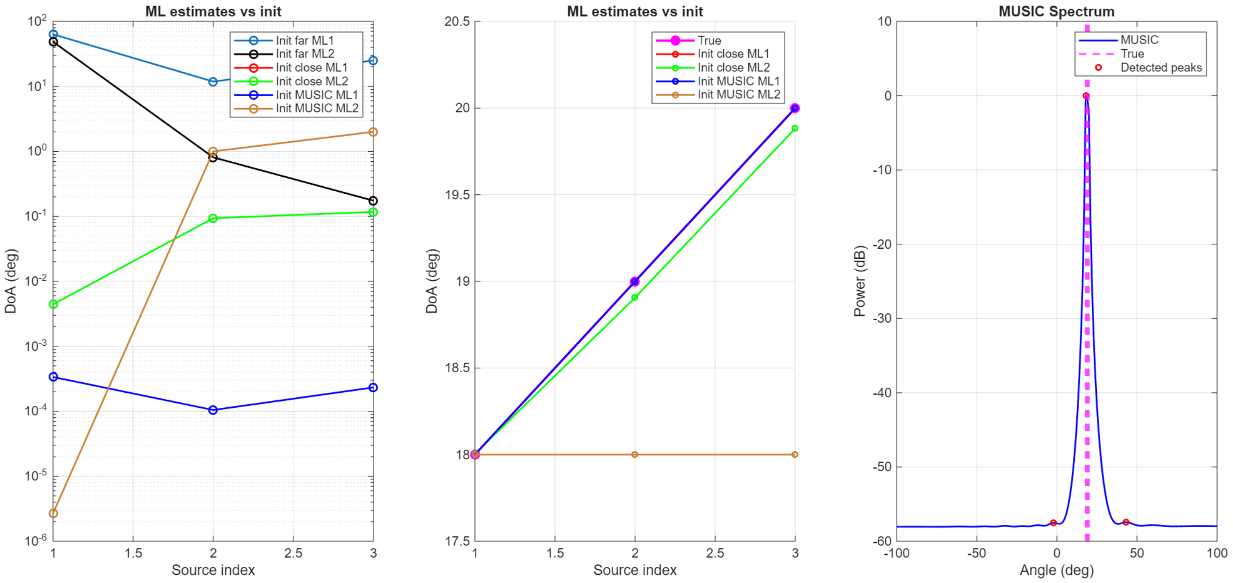


Figure 12: ML estimates with different initializations compared to true source directions

The results show that:

- For closely spaced sources, ML often fails to converge when initialized far from the true angles. Small initialization errors lead to large estimation errors.

- MUSIC-based initialization consistently yields accurate ML estimates, closely matching the true DoAs.

- For well-separated sources, ML is more robust, though MUSIC still provides the most reliable starting point.

This demonstrates the practical value of combining subspace methods like MUSIC with ML estimation: MUSIC provides good initial guesses that improve ML robustness and accuracy, especially in challenging scenarios with closely spaced sources. While MUSIC alone is effective for most cases, ML can refine estimates when high angular precision is required. However, when the signals are stochastic, ML without good initialization or prior knowledge cannot reliably separate closely spaced sources.

Conclusion