Core Principle of MUSIC: Subspace Orthogonality

1. Signal Model

We consider a discrete-time signal $x[n]$ composed of a sum of M complex sinusoids and additive white noise $w[n]$. The sampling frequency is denoted by F_s.

$$x[n] = \sum_{k=1}^{M} A_k \exp\left\{ j \left(\frac{2\pi f_k n}{F_s} + \phi_k \right) \right\} + w[n]$$

Here, j is the imaginary unit, and f_k is the physical frequency in Hz. This signal model can be expressed more concisely using the normalized angular frequency, $\omega_k = 2\pi f_k / F_s$.

$$x[n] = \sum_{k=1}^{M} c_k \exp(j \omega_k n) + w[n]$$

The term ω_k, with units of radians per sample [rad/sample], represents the "phase change per sample" and is a key quantity that characterizes the signal's periodicity. The term $c_k = A_k \exp(j\phi_k)$ is the complex amplitude.

2. Covariance Matrix Estimation

Subspace methods begin by analyzing a covariance matrix, which captures the statistical properties of the signal. This section explains how this crucial matrix is constructed from an observed sequence of signal samples.

2.1. Why a Matrix? Capturing the Signal's Structure

The data we have is a one-dimensional sequence of samples, $x[0], x[1], x[2], \ldots$. However, a sinusoidal signal contains an inherent temporal correlation structure, meaning that future samples can be partially predicted from past samples. To reveal this hidden structure, we embed the one-dimensional signal sequence into a multi-dimensional vector space.

This is achieved by creating delay vectors. We first define a subspace dimension, L, and then progressively form L-dimensional vectors from the signal:

- $\mathbf{v}_0 = [x[0], x[1], ..., x[L-1]]^\top$
- $\mathbf{v}_1 = [x[1], x[2], ..., x[L]]^\top$
- $\mathbf{v}_2 = [x[2], x[3], ..., x[L+1]]^\top$

- ...

These vectors $\mathbf{v}_i$ can be viewed as “snapshots” of L consecutive samples of the signal. The subspace dimension L is a key parameter that affects the analysis.

2.2 Defining the Covariance Matrix

The covariance matrix $\mathbf{R}_x$ represents the average correlation between these snapshot vectors. Mathematically, it is defined as the expected value of the outer product of a snapshot vector $\mathbf{v}_i$ with its Hermitian conjugate $\mathbf{v}_i^H$ (also denoted as $\mathbf{v}_i^*$).

$$\mathbf{R}_x = \mathbb{E}[\mathbf{v}_i \mathbf{v}_i^H]$$

In practice, we only have a finite number of samples. Therefore, we approximate this expectation by computing the sample covariance matrix, $\hat{\mathbf{R}}_x$, which is the sample mean over N available snapshots:

$$\mathbf{R}_x \approx \hat{\mathbf{R}}_x \triangleq \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{v}_i \mathbf{v}_i^H$$

2.3. Efficient Calculation using a Hankel Matrix

To compute the sample mean efficiently, we can leverage a Hankel matrix. A data matrix $\mathbf{X}$, formed by concatenating all snapshot vectors $\mathbf{v}_i$ as its columns, naturally forms a Hankel matrix:

$$\begin{aligned}\mathbf{X} &= [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N-1}] \\ &= \begin{bmatrix} x[0] & x[1] & x[2] & \cdots & x[N-1] \\ x[1] & x[2] & x[3] & \cdots & x[N] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[L-1] & x[L] & x[L+1] & \cdots & x[N+L-2] \end{bmatrix}\end{aligned}$$

(Note: This is a non-square Hankel matrix, which aligns with the usage in `scipy.linalg.hankel`.)

Using this Hankel matrix $\mathbf{X}$, the sample covariance matrix can be calculated concisely with a single matrix multiplication, which is much more efficient than an explicit `for` loop:

2.4. Improving Estimation with Forward-Backward Averaging

To further enhance the estimation accuracy, especially with a limited number of samples, a technique called Forward-Backward Averaging [5] can be applied. This method leverages the property that the statistical characteristics of a sinusoidal signal remain unchanged when it is time-reversed.

The forward-backward averaged covariance matrix $\hat{\mathbf{R}}_{fb}$ is computed by averaging the standard forward covariance matrix $\hat{\mathbf{R}}_f$ (calculated above) and a backward covariance matrix $\hat{\mathbf{R}}_b$. The backward matrix is derived from the time-reversed, complex-conjugated signal, and can be shown to be $\mathbf{J}\overline{\hat{\mathbf{R}}}_f\mathbf{J}$, where $\mathbf{J}$ is the exchange matrix.

$$\hat{\mathbf{R}}_{fb} = \frac{\hat{\mathbf{R}}_f + \mathbf{J}\overline{\hat{\mathbf{R}}}_f\mathbf{J}}{2}$$

Using $\hat{\mathbf{R}}_{fb}$ instead of $\hat{\mathbf{R}}_f$ yields a more statistically stable estimate, which in turn improves the accuracy of the MUSIC algorithm, particularly for short signals or in low SNR conditions.

3. Eigendecomposition and Subspace Separation

The covariance matrix $\mathbf{R}_x$ encapsulates the temporal correlation information of the signal. The next step is to use a powerful mathematical tool, **Eigenvalue Decomposition (EVD)**, to separate the signal and noise components from within this matrix.

3.1. What is Eigendecomposition? Revealing the Matrix’s Essence

Any square matrix can be decomposed into its fundamental properties: its **“essential directions” (eigenvectors)** and the **“strength of influence” in those directions (eigenvalues)**. This is what eigenvalue decomposition provides.

$$\mathbf{R}_x = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H$$

- **$\mathbf{E}$ (Eigenvector Matrix)**: The columns of this matrix are the eigenvectors $\mathbf{e}_i$, which form an orthonormal basis representing the “essential directions” of $\mathbf{R}_x$. These eigenvectors can be thought of as the “fundamental patterns” or basis waveforms that can be combined to represent the signal components.
- **$\mathbf{\Lambda}$ (Diagonal Matrix of Eigenvalues)**: The diagonal elements of this matrix are the eigenvalues λ_i, corresponding to the “strength” of each eigenvector. For a covariance matrix, each eigenvalue represents how much power (energy) its corresponding eigenvector (fundamental pattern) contributes to the total power of the signal.

3.2. Separating Signal from Noise: The Power Disparity

Let us recall our signal model: $x[n] = (\text{signal}) + (\text{noise})$.

- **Signal Components**: These are structured waveforms with specific frequencies. Their energy should be concentrated in the directions of a few specific eigenvectors.
- **Noise Component (White Noise)**: This is a completely random waveform containing all frequency components equally. Its energy is not concentrated in any particular direction but should be distributed evenly across all eigenvector directions.

Consequently, the eigenvalues of the covariance matrix $\mathbf{R}_x$ will distinctly separate into two groups.

- **Large Eigenvalues**: The $2M$ largest eigenvalues correspond to the M sinusoids (one pair for each real sinusoid). They represent the combined power of **"signal plus noise"** and thus have large magnitudes.
- **Small Eigenvalues**: The remaining $L - 2M$ eigenvalues correspond to "noise only". They will all be small and have magnitudes on the order of the noise variance (σ^2).

3.3. The Birth of Subspaces

This distinct "cliff" in the magnitude of the eigenvalues allows us to partition the set of all eigenvectors into two groups. This is the act of **"separating into subspaces."**

- **Signal Subspace $\mathbf{E}_s$**: The subspace spanned by the $2M$ eigenvectors corresponding to the largest eigenvalues. This space is the domain where the signal components "live"; all essential information about the signal is contained within this subspace.
- **Noise Subspace $\mathbf{E}_n$**: The subspace spanned by the $L - 2M$ eigenvectors corresponding to the smallest eigenvalues. This space is a domain of pure noise, where no signal components reside.

Crucially, the signal subspace and the noise subspace are, by definition of EVD, **mutually orthogonal**. This orthogonality is a key property that MUSIC-family algorithms exploit.

(Note: In implementation, the signal subspace $\mathbf{E}_s$ is stored as an $L \times 2M$ matrix whose columns are the corresponding eigenvectors. The dimension L is typically chosen to be around 1/3 to 1/2 of the signal frame length.)

4. The Principle of Orthogonality

The foundation of the MUSIC algorithm is the principle that the signal subspace $\mathbf{E}_s$ and the noise subspace $\mathbf{E}_n$ can be considered approximately orthogonal. This section explains the reasoning behind this **"principle of orthogonality."**

Let's first consider the noise-free covariance matrix, $\mathbf{R}_s$.

$$\mathbf{R}_s = \mathbf{A}\mathbf{S}\mathbf{A}^H$$

Here, $\mathbf{A}$ is the $L \times M$ steering matrix, whose columns are the steering vectors.

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \ldots, \mathbf{a}(\omega_M)] \\ \mathbf{a}(\omega_k) &= [1, \exp(j\omega_k), \exp(j2\omega_k), ..., \exp(j(L-1)\omega_k)]^\top \end{aligned}$$

And $\mathbf{S}$ is the $M \times M$ source covariance matrix.

$$\mathbf{S} = \mathbb{E}[\mathbf{s}(n)\mathbf{s}(n)^H]$$

where $\mathbf{s}(n) = [s_1(n), s_2(n), ..., s_M(n)]^\top$ is the vector of source signals at time n.

The range space (or column space) of this matrix, $\mathcal{R}(\mathbf{R}_s)$, is the space spanned by the columns of $\mathbf{A}$—that is, the space spanned by the steering vectors $\mathbf{a}(\omega_k)$. This is the true signal subspace (see Appendix A for a proof).

By definition, the left null space of $\mathbf{R}_s$, $\mathcal{N}(\mathbf{R}_s^H)$, is orthogonal to its range space $\mathcal{R}(\mathbf{R}_s)$. Any vector $\mathbf{v} \in \mathcal{N}(\mathbf{R}_s^H)$ satisfies $\mathbf{v}^H\mathbf{R}_s = \mathbf{0}$. Because the columns of $\mathbf{A}$ are linearly independent (rank M), this implies $\mathbf{v}^H\mathbf{a}(\omega_k) = 0$ for all k. In other words, the null space $\mathcal{N}(\mathbf{R}_s^H)$ is the space orthogonal to all signal steering vectors.

Now, consider the covariance matrix with noise, $\mathbf{R}_x = \mathbf{R}_s + \sigma^2\mathbf{I}$. The eigenvectors $\mathbf{e}_i$ of this matrix behave as follows:

- The eigenvectors corresponding to the signal ($i \leq 2M$) approximately span the range space $\mathcal{R}(\mathbf{R}_s)$ (the signal subspace).
- The eigenvectors corresponding only to noise ($i > 2M$) approximately span the null space $\mathcal{N}(\mathbf{R}_s^H)$ (the noise subspace).

Therefore, the eigenvectors that span the noise subspace $\mathbf{E}_n$ are approximately orthogonal to the signal steering vectors $\mathbf{a}(\omega_k)$:

$$\mathbf{a}(\omega_k)^H\mathbf{e}_i \approx 0;;; (\text{for}; i > 2M)$$

This can be compactly written as $\mathbf{a}(\omega_k)^H\mathbf{E}_n \approx \mathbf{0}$. Since $\mathbf{a}(\omega_k)$ belongs to the signal subspace, this demonstrates that $\mathbf{E}_s$ and $\mathbf{E}_n$ are approximately orthogonal. This orthogonality is the key property exploited by the MUSIC algorithm.

5. Two Approaches to Frequency Estimation

Once the noise subspace $\mathbf{E}_n$ has been estimated, there are two primary approaches to using its orthogonality property to estimate the signal frequencies.

5.1. Spectral MUSIC: The Peak-Picking Approach

The Spectral MUSIC approach involves computing a **pseudospectrum** over a predefined grid of discrete frequency points and then searching for its peaks. This pseudospectrum is defined to evaluate the orthogonality:

$$\hat{P}_{MU}(\omega) = \frac{1}{\mathbf{a}(\omega)^H \mathbf{E}_n \mathbf{E}_n^H \mathbf{a}(\omega)}$$

At a true normalized angular frequency ω_k, the steering vector $\mathbf{a}(\omega_k)$ is orthogonal to the noise subspace $\mathbf{E}_n$. Consequently, the denominator approaches zero, causing a sharp peak to appear in the MUSIC spectrum $\hat{P}_{\text{MU}}(\omega)$. By searching for the locations of these peaks, we can obtain estimates of the normalized angular frequencies, $\hat{\omega}_k$.

Finally, these are converted to estimates of the physical frequencies in Hz, $\hat{f}_k$:

$$\hat{f}_k = \frac{F_s}{2\pi} \hat{\omega}_k$$

- **Advantages**:
  - The concept is intuitive and easy to understand.
  - The resulting spectrum can be visualized, which is useful for analysis.
  - As implemented in this project, the denominator can be computed very efficiently for all frequency grid points at once using the Fast Fourier Transform (FFT).
- **Disadvantages**:
  - There is a trade-off between estimation accuracy and computational cost, which depends on the density of the frequency grid.
  - A finer grid yields higher accuracy but requires more computation.
  - Since the estimate is chosen from a set of predefined frequency candidates (the grid), an inherent estimation error (quantization error) occurs if the true frequency lies between grid points.

5.2. Root-MUSIC: The Polynomial Rooting Approach

Root-MUSIC avoids the spectral search by reformulating the problem algebraically. The denominator of the MUSIC spectrum, $D(z) = \mathbf{a}(z)^H \mathbf{C} \mathbf{a}(z)$ (where $\mathbf{C} = \mathbf{E}_n \mathbf{E}_n^H$), can be expressed as a polynomial in z, where $z = e^{j\omega}$.

Root-MUSIC directly computes the roots of this polynomial. The roots corresponding to the signal frequencies will lie on or very close to the unit circle in the complex plane.

The solve method finds the roots of the polynomial $F(z) = z^{L-1} D(z) = 0$, which is a standard polynomial form without negative powers. The roots are of the form $z_k = e^{j\omega_k}$, so the normalized angular frequencies $\hat{\omega}_k$ are obtained by calculating their phase angles. These are then converted to physical frequencies $\hat{f}_k$ in the same way as in Spectral MUSIC.

- **Advantages**:
  - It avoids a spectral search, so the computational cost does not depend on the desired frequency resolution.
  - By computing the roots, it estimates frequencies as continuous values, allowing for theoretically higher accuracy (an “off-grid” estimate).
- **Disadvantages**:
  - When noise is present, practical implementations require careful heuristics to stably select the true signal roots from the many extraneous roots that are also computed.
  - The computational cost of standard polynomial solvers (like numpy.roots, which uses an eigenvalue-based approach) is high, typically on the order of $O(L^3)$, where L is the polynomial degree. Since L is the subspace dimension, it can be quite large (e.g., ~1500 in our experiments), making this step a significant bottleneck.

In the ideal, noise-free case, all signal roots lie precisely on the unit circle. In practice, noise and finite-sample effects cause the roots to deviate from the unit circle.

One effective heuristic, adopted in our implementation, is to first select an- oversized set of candidate roots—for instance, the $4M$ roots closest to the unit circle. This over-selection ensures that all true signal roots are included, even in the presence of spurious roots caused by noise. Subsequent filtering steps then isolate the desired M positive frequency components from this candidate set.

Appendix A: Proofs and Supplements

A.1. Proof: Equivalence of Covariance Range and Steering Vector Space

Here we prove that $\mathcal{R}(\mathbf{A}\mathbf{S}\mathbf{A}^H) = \mathcal{R}(\mathbf{A})$.

1. **Proof of $\mathcal{R}(\mathbf{A}\mathbf{S}\mathbf{A}^H) \subseteq \mathcal{R}(\mathbf{A})$**: Let $\mathbf{y}$ be an arbitrary vector in $\mathcal{R}(\mathbf{A}\mathbf{S}\mathbf{A}^H)$. By definition, there exists a vector $\mathbf{x}$ such that $\mathbf{y} = (\mathbf{A}\mathbf{S}\mathbf{A}^H)\mathbf{x}$. If we let $\mathbf{z} = \mathbf{S}\mathbf{A}^H\mathbf{x}$, we can write $\mathbf{y} = \mathbf{A}\mathbf{z}$. This shows that $\mathbf{y}$ is a linear combination of the columns of $\mathbf{A}$, which means $\mathbf{y} \in \mathcal{R}(\mathbf{A})$.

2. **Proof of $\mathcal{R}(\mathbf{A}) \subseteq \mathcal{R}(\mathbf{A}\mathbf{S}\mathbf{A}^H)$**: Let $\mathbf{y}$ be an arbitrary vector in $\mathcal{R}(\mathbf{A})$. Then there exists a vector $\mathbf{x}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. The source covariance matrix $\mathbf{S}$ is a diagonal matrix with positive diagonal elements (the power of each source, see Appendix A.2) and is therefore invertible. The steering matrix $\mathbf{A}$ is full-rank, as its columns are linearly independent steering vectors. We can write $\mathbf{y} = \mathbf{A}\mathbf{S}\mathbf{S}^{-1}\mathbf{x}$. Let $\mathbf{w} = \mathbf{S}^{-1}\mathbf{x}$. Since $\mathbf{A}^H$ has full row rank, there exists a vector $\mathbf{v}$ such that $\mathbf{A}^H\mathbf{v} = \mathbf{w}$. Substituting this, we get $\mathbf{y} = \mathbf{A}\mathbf{S}(\mathbf{A}^H\mathbf{v}) = (\mathbf{A}\mathbf{S}\mathbf{A}^H)\mathbf{v}$. This shows that $\mathbf{y} \in \mathcal{R}(\mathbf{A}\mathbf{S}\mathbf{A}^H)$.

A.2. Supplement: Why the Source Covariance Matrix is Diagonal

For a sinusoidal signal $s_k(n) = A_k \exp(j(\omega_k n + \phi_k))$, we can treat it as a stationary stochastic process by assuming that the amplitude A_k and initial phase ϕ_k are random variables. We further assume that for each sinusoid k, A_k and ϕ_k are independent, and the parameters (A_k, ϕ_k) are mutually independent from (A_l, ϕ_l) for $k \neq l$. Under these assumptions, the cross-correlation for $k \neq l$ is:

$$\mathbb{E}[s_k(n)s_l(n)^H] = \mathbb{E}[A_k]\mathbb{E}[A_l]e^{j(\omega_k - \omega_l)n}\mathbb{E}[e^{j\phi_k}]\mathbb{E}[e^{-j\phi_l}]$$

If we assume the phases ϕ_k are uniformly distributed in $[-\pi, \pi]$, then $\mathbb{E}[e^{j\phi_k}] = 0$, making the entire expression zero. Thus, the sinusoids are uncorrelated.

For $k = l$, the autocorrelation is the signal power $P_k = \mathbb{E}[|s_k(n)|^2] = \mathbb{E}[|A_k|^2]$. Therefore, the source covariance matrix $\mathbf{S} = \mathbb{E}[\mathbf{s}(n)\mathbf{s}(n)^H]$ is a diagonal matrix with the power of each sinusoid on its diagonal.