Direction Of Arrival Estimation Algorithms

N. P. Waweru, D. B. O. Konditi, P. K. Langat

Kobi Rahimi

01

Introduction

Performance dependencies of DOA estimation algorithms

01. Size of array

    Number of sensors

02. Size of array

    Spacing between sensors

03. Angular distance of impinging signals

04. Number of impinging signals

05. Number of snapshots

06. SNR

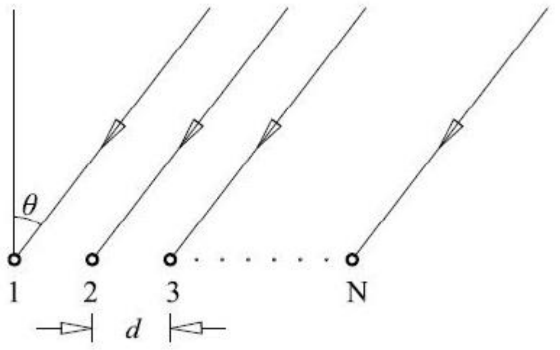
Parameters and array configuration

Parameters:

- f – main frequency of impinging signals.
- λ − wavelength.
- $d = \frac{\lambda}{2}$ − distance between adjoint sensors.
- N – number of sensors.
- M – number of impinging signals.

Array configuration:

- ULA



ULA with N sensors & 1 signal

- Assumption – the signal is in far field.
- Reference point – sensor number 1.
- The delay in which each sensor will receive the signal with respect to sensor 1 is: $\tau_i = \frac{(i-1)\, dsin(\theta)}{c}$
- Assuming $s(t)$ is a narrow band/ monochromatic signal, then:

  1. $x_1(t) = s(t)$

  2. $x_i(t) = e^{-j\omega\tau_i} \cdot \underbrace{x_1(t)}_{s(t)} |_{\frac{\omega}{c}=k} = e^{-jkd(i-1)\, sin(\theta)} \cdot s(t)|_{\psi=kdsin(\theta)} = e^{-j(i-1)\psi} \cdot s(t)$

Where $s(t)$ is the signal the first sensor receives and $e^{-j(i-1)\psi} \cdot s(t)$ is the signal the i_{th} sensor receives.

ULA with N sensors & M signals

- Vector form for a single signal:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ x_N(t) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j\psi} \\ e^{-2j\psi} \\ . \\ . \\ e^{-j(N-1)\psi} \end{bmatrix} \cdot s(t) = \boldsymbol{a}(\theta) \cdot s(t) \qquad \boldsymbol{a}(\theta) - steering\ vector$$

- For M signals impinging on the N sensor array we get M different steering vectors:

$$\underbrace{\boldsymbol{x}(t)}_{NX1} = \underbrace{\boldsymbol{A}}_{NXM} \cdot \underbrace{\boldsymbol{s}(t)}_{MX1}$$

where:

$$\boldsymbol{A} = [a(\theta_1)\ a(\theta_2) \dots .\ a(\theta_M)] = \begin{bmatrix} 1 & 1 & \dots .. & 1 \\ e^{-j\psi_1} & e^{-j\psi_2} & \dots .. & e^{-j\psi_M} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ e^{-j(N-1)\psi_1} & e^{-j(N-1)\psi_2} & & e^{-j(N-1)\psi_M} \end{bmatrix} \qquad \psi_i = kdsin(\theta_i), 1 \le i \le M$$

- $\boldsymbol{s}(t)$ is now a vector of M signals: $\boldsymbol{s}(t)^T = [s_1(t)\ s_2(t)\ \dots s_M(t)]$

02

MUSIC

Multi Signal Classification

Definitions & Assumptions

- Assume we have:

  1. ULA with N sensors.

  2. M signals : $s_1(t), s_2(t) \dots . s_M(t)$ arriving from directions: $\theta_1, \theta_2 \dots . \theta_M$ respectively.

- In the presence of noise, the received signal is:

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

- Defining the NXN autocorrelation matrix of the received signal $\boldsymbol{R}_{xx}$ as:

  $\boldsymbol{R}_{xx} = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)] = E\left[\left(\boldsymbol{A} \cdot \boldsymbol{s}(t) + \boldsymbol{n}(t)\right) \cdot \left(\boldsymbol{s}^H(t) \cdot \boldsymbol{A}^H + \boldsymbol{n}^H(t)\right)\right] = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \boldsymbol{R}_{nn}$

  assuming the noise and the signals are uncorrelated.

- For white noise with the same power received at each sensor σ_0^2 we get:

  $\boldsymbol{R}_{xx} = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\mathbf{I}.$

- $\boldsymbol{R}_{ss}$ is the source cross correlation matrix which is : $\boldsymbol{R}_{ss} = E[\boldsymbol{s}(t) \cdot \boldsymbol{s}^H(t)]$.

- Assuming uncorrelated signals we get: $\boldsymbol{R}_{ss} = diag(\sigma_1^2, \sigma_2^2 \dots . \sigma_M^2)$.

First step – without the presence of noise

- $\boldsymbol{R}_{xx} = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H$
- $\boldsymbol{R}_{ss}$ - full rank matrix of M eigenvectors corresponding to the M eigenvalues - $\sigma_1^2, \sigma_2^2 \dots . \sigma_M^2$.
- $\boldsymbol{A} = [a(\theta_1)\ a(\theta_2) \dots .\ a(\theta_M)]$ is a Vandermonde matrix where $x = e^{-j\psi_i}$ and as long as $\theta_i \neq \theta_j\ \forall\ i \neq j$ then each column is independent.
- $rank(\boldsymbol{A}) = rank(\boldsymbol{A}^H) = \min(N, M) = M$ since $N \geq M$
- $rank(\boldsymbol{A}\boldsymbol{R}_{ss}) = \min\big(rank(\boldsymbol{A}),\ rank(\boldsymbol{R}_{ss})\big) = M$
- $rank(\boldsymbol{R}_{xx}) = rank(\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H) = \min\big(rank(\boldsymbol{A}\boldsymbol{R}_{ss}),\ rank(\boldsymbol{A}^H)\big) = M$
- $\boldsymbol{R}_{xx} = \underbrace{\boldsymbol{A}}_{NXM} \cdot \underbrace{\boldsymbol{R}_{ss}}_{MXM} \cdot \underbrace{\boldsymbol{A}^H}_{MXN} \rightarrow NxN$ matrix. Since $N \geq M$ the matrix is rank deficient.
- In practice we'll need to estimate $\boldsymbol{R}_{xx}$: $\widetilde{\boldsymbol{R}}_{xx} = \frac{1}{J}\sum_{j=1}^{J}\boldsymbol{x}(j)\boldsymbol{x}^H(j)$, $\lim_{j\to\infty}\widetilde{\boldsymbol{R}}_{xx} = \boldsymbol{R}_{xx}$ where J is the number of snapshots.

Properties of Hermitian matrix

- $\boldsymbol{A}$ is Hermitian $\Leftrightarrow \boldsymbol{A} = \boldsymbol{A}^H$
- eigenvalues of Hermitian matrix are real – proof:

  Let λ be an eigenvalue of vector v and $\boldsymbol{A}$ is Hermitian matrix thus, $\boldsymbol{A}v = \lambda v \Rightarrow v^H \boldsymbol{A} v = v^H \lambda v = \lambda \left|\left|v\right|\right|^2$

  the conjugate transpose of both sides of the equation: $v^H \boldsymbol{A}^H v = \lambda^* \left|\left|v\right|\right|^2$.

  since $\boldsymbol{A} = \boldsymbol{A}^H$: $v^H \boldsymbol{A}^H v = v^H \boldsymbol{A} v = \lambda \left|\left|v\right|\right|^2 \Rightarrow \lambda \left|\left|v\right|\right|^2 = \lambda^* \left|\left|v\right|\right|^2 \Rightarrow \lambda = \lambda^* \Rightarrow \boldsymbol{\lambda}$ **is real**.
- eigenvectors of Hermitian matrix are orthogonal – Proof:

  Let λ_i and λ_j be the eigenvalues of v_i and v_j respectively and $\boldsymbol{A}$ is Hermitian matrix thus,

  $$\boldsymbol{A}v_i = \lambda_i v_i \Rightarrow v_i^H \boldsymbol{A}^H = \lambda_i^* v_i^H \Rightarrow v_i^H \underbrace{\boldsymbol{A}^H}_{\boldsymbol{A}} v_j = \lambda_i^* v_i^H v_j \Rightarrow \lambda_j v_i^H v_j = \lambda_i^* v_i^H v_j \Rightarrow v_i^H v_j \left(\lambda_i^* - \lambda_j\right) = 0$$

  since $\lambda_i^* \neq \lambda_j$ we get $v_1^H v_2 = 0 \Rightarrow \boldsymbol{v_i} \perp \boldsymbol{v_j}$ for $i \neq j$
- $\boldsymbol{A}$ is positive definite $\Leftrightarrow$ all its eigenvalues are positive.
- $\boldsymbol{A}$ is positive semi-definite $\Leftrightarrow$ all its eigenvalues are non-negative.

Autocorrelation matrix $\boldsymbol{R_{xx}}$

- $\boldsymbol{R}_{xx}^H = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)]^H = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)] = \boldsymbol{R}_{xx} \rightarrow \boldsymbol{R}_{xx}$ is Hermitian, thus maintain all the properties in the previous slide.
- $\boldsymbol{R}_{ss}^H = E[\boldsymbol{s}(t) \cdot \boldsymbol{s}^H(t)]^H = E[\boldsymbol{s}(t) \cdot \boldsymbol{s}^H(t)] = \boldsymbol{R}_{ss} \rightarrow \boldsymbol{R}_{ss}$ is also Hermitian.
- Definition of positive definite matrix: $\forall\, \bar{x} \neq \bar{0}$: $\bar{x}^H \boldsymbol{A} \bar{x} > 0$
- $\bar{x}^H \boldsymbol{R}_{ss} \bar{x} = \bar{x}^H diag(\sigma_1^2, \sigma_2^2 \ldots .\ \sigma_M^2)\bar{x} = |x_1|^2\sigma_1^2 + |x_2|^2\sigma_2^2 + |x_M|^2\sigma_M^2 > 0 \Rightarrow \boldsymbol{R}_{ss}$ is a positive definite MxM matrix.
- Since $\boldsymbol{R}_{ss}$ is Hermitian and positive definite matrix, all its eigenvalues are positive.
- Since $\boldsymbol{R}_{xx} = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H$ is NxN Hermitian matrix with rank M and $\boldsymbol{R}_{ss}$ is positive definite matrix then $\boldsymbol{R}_{xx}$ is a positive semi-definite matrix with M positive eigenvalues and $N - M$ zero eigenvalues.

Autocorrelation matrix $\boldsymbol{R_{xx}}$

- $\boldsymbol{R}_{xx}^H = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)]^H = E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)] = \boldsymbol{R}_{xx} \rightarrow \boldsymbol{R}_{xx}$ is Hermitian, thus maintain all the properties in the previous slide.
- $\boldsymbol{R}_{ss}^H = E[\boldsymbol{s}(t) \cdot \boldsymbol{s}^H(t)]^H = E[\boldsymbol{s}(t) \cdot \boldsymbol{s}^H(t)] = \boldsymbol{R}_{ss} \rightarrow \boldsymbol{R}_{ss}$ is also Hermitian.
- Definition of positive definite matrix: $\forall\, \bar{x} \neq \bar{0}$: $\bar{x}^H \boldsymbol{A} \bar{x} > 0$
- $\bar{x}^H \boldsymbol{R}_{ss} \bar{x} = \bar{x}^H diag(\sigma_1^2, \sigma_2^2 \ldots .\ \sigma_M^2)\bar{x} = |x_1|^2\sigma_1^2 + |x_2|^2\sigma_2^2 + |x_M|^2\sigma_M^2 > 0 \Rightarrow \boldsymbol{R}_{ss}$ is a positive definite MxM matrix.
- Since $\boldsymbol{R}_{ss}$ is Hermitian and positive definite matrix, all its eigenvalues are positive.
- Since $\boldsymbol{R}_{xx} = \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H$ is NxN Hermitian matrix with rank M and $\boldsymbol{R}_{ss}$ is positive definite matrix then $\boldsymbol{R}_{xx}$ is a positive semi-definite matrix with M positive eigenvalues and $N - M$ zero eigenvalues.

Why is $\boldsymbol{R}_{xx}$ positive semi-definite matrix

- Since $\boldsymbol{R}_{xx}$ is NxN matrix and its rank is M, it means that that it has $N-M$ zero eigenvalues.
- For $\boldsymbol{R}_{xx}$ to be a positive semi-definite matrix we need to show that $\forall\, \bar{x} \neq \bar{0}$: $x^H \boldsymbol{R}_{xx} x \geq 0$.

- $x^H \boldsymbol{R}_{xx} x = \underbrace{x^H \boldsymbol{A}}_{v^H} \boldsymbol{R}_{ss} \underbrace{\boldsymbol{A}^H x}_{v} = v^H \boldsymbol{R}_{ss} v > 0$ ($\boldsymbol{R}_{ss}$ is positive definite matrix).
- Having said that, the eigenvalues of $\boldsymbol{R}_{xx}$ that are not zero are positive.
- Since we showed that $\boldsymbol{R}_{xx}$ is Hermitian positive semi-definite, its eigenvalues are non-negative.

Second step – with presence of noise

- $\boldsymbol{R}_{xx} = \ \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\ \mathbf{I}$.
- Since all the eigenvalues of $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H$ are real non-negative and the eigenvalues of $\sigma_0^2\mathbf{I}$ are real positive (σ_0^2) we get a full rank matrix $\boldsymbol{R}_{xx}$ with only real positive eigenvalues.
- Explanation: we saw that $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H$ has M real positive eigenvalues, so they satisfy:

  $\det(\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H - \lambda\boldsymbol{I}) = 0$, they will be: $\lambda = [\lambda_1, \lambda_2 \ldots \lambda_M \underbrace{0, 0 \ldots . 0}_{N-M}]$.

  the eigenvalues of $\boldsymbol{R}_{xx}$ need to satisfy $\det(\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\ \mathbf{I} - \mu\boldsymbol{I}) = 0 \Rightarrow \det(\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H - (\mu - \sigma_0^2)\mathbf{I})$, in that case: $\mu = [\underbrace{\lambda_1 + \sigma_0^2, \lambda_2 + \sigma_0^2, \ldots \lambda_M + \sigma_0^2}_{M\ eigenvalues\ relevant\ to\ signal},\ \underbrace{\sigma_0^2, \sigma_0^2 \ldots . \sigma_0^2}_{\substack{N-M\ eigenvalues \\ of\ noise}}\ ]$.

- Since $\sigma_0^2 > 0$ and $\forall\, i$: $\lambda_i > 0$ the smallest eigenvalues of $\boldsymbol{R}_{xx}$ are σ_0^2 which are related only to the noise and there are $N - M$ of them.
- The other M eigenvalues are related to the signals and are bigger than σ_0^2.

Importance of real positive eigenvalues

- If $AR_{ss}A^H$ had negative eigenvalues, for example $\lambda_i = -\sigma_0^2$ we would get that at least one eigenvalue of $\boldsymbol{R}_{xx}$ is zero thus $\boldsymbol{R}_{xx}$ is not full rank.
- For $\lambda_i < -\sigma_0^2$ $\boldsymbol{R}_{xx}$ is still a full rank matrix but we wouldn't be able to classify that the M largest eigenvalues belong to the signals and that the other $N - M$ belong to the noise since it's not true anymore.

Implementation of MUSIC algorithm

- After eigenvalue decomposition of $\boldsymbol{R}_{xx}$ we'll sort the eigenvalues from highest to lowest:

  $\mu_1 \geq \mu_2 \geq \cdots \mu_N > 0$ and the first M highest eigenvalues correspond to the signal while the other

  $N - M$ eigenvalues correspond to the noise .

- Letting $[v_1, v_2 \ldots v_N]$ be the eigenvectors of the eigenvalues $\mu_1, \mu_2 \ldots \mu_N$ we get that $[v_1, v_2 \ldots v_M]$ correspond to the signal while $[v_{M+1}, v_{M+2} \ldots v_N]$ correspond to the noise.

- Let μ_i be the i_{th} eigenvalue of matrix $\boldsymbol{R}_{xx}$ and v_i the corresponding eigenvector, then:

  $\boldsymbol{R}_{xx} v_i = \mu_i v_i \Rightarrow (\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2 \boldsymbol{I}) v_i = \mu_i v_i$.

- Let $\mu_i = \sigma_0^2$ (corresponding to the noise): $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H v_i + \sigma_0^2 v_i = \sigma_0^2 v_i \Rightarrow \boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H v_i = 0$

- $rank(\boldsymbol{A}^H\boldsymbol{A}) = \min\big(rank(\boldsymbol{A}^H), rank(\boldsymbol{A})\big) = M$ and $\boldsymbol{A}^H\boldsymbol{A}$ is MxM matrix thus $\boldsymbol{A}^H\boldsymbol{A}$ is full rank $\Rightarrow \boldsymbol{A}^H\boldsymbol{A}$ has an inverse: $(\boldsymbol{A}^H\boldsymbol{A})^{-1}$.
- We saw that $\boldsymbol{R}_{ss}$ is also MxM full rank matrix $\Rightarrow \boldsymbol{R}_{ss}$ has an inverse: $\boldsymbol{R}_{ss}^{-1}$.
- $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H v_i = 0 \Rightarrow \boldsymbol{R}_{ss}^{-1}(\boldsymbol{A}^H\boldsymbol{A})^{-1}\boldsymbol{A}^H(\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H v_i) = 0 \Rightarrow \boldsymbol{A}^H vi = 0\,, i = M+1, M+2, \ldots. N$

$$\boldsymbol{A}^H vi = 0 \Rightarrow \begin{bmatrix} \leftarrow a(\theta_1)^* \rightarrow \\ \leftarrow a(\theta_2)^* \rightarrow \\ . \\ . \\ . \\ \leftarrow a(\theta_M)^* \rightarrow \end{bmatrix} \cdot \begin{bmatrix} v_{i1} \\ v_{i2} \\ . \\ . \\ . \\ v_{iN} \end{bmatrix} = \begin{bmatrix} a(\theta_1)^H v_i \\ a(\theta_2)^H v_i \\ . \\ . \\ . \\ a(\theta_M)^H v_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

- This equation indicates that the eigenvector v_i corresponding to the noise is orthogonal to the columns of the matrix $\boldsymbol{A}$.

Implementation of MUSIC algorithm

- Constructing a noise matrix from the eigenvectors corresponding to the noise will yield $\boldsymbol{E}_n = [v_{M+1}\ v_{M+1}\ \ldots .v_N]$ and since each column vector in $\boldsymbol{E}_n$ is orthogonal to the columns of $\boldsymbol{A}$ we get that the inner product is : $\left|\left|a(\theta)^H \boldsymbol{E}_n\right|\right|^2 = a(\theta)^H \boldsymbol{E}_n \boldsymbol{E}_n^H a(\theta) = 0$

- Taking the inverse, we get the spatial spectrum : $P_{MUSIC} = \frac{1}{a(\theta)^H \boldsymbol{E}_n \boldsymbol{E}_n^H a(\theta)}$ which indicates that a specific θ is an angle of arrival if we see a peak at that angle.

- By scanning every angle we'll get high peaks only from angles of arriving signals because the vectors $a(\theta)$ are the steering vectors of the signals impinging the array.

MUSIC - Summery

- Obtain the following estimation of the autocorrelation matrix based on the N received signal vectors: $\widetilde{\boldsymbol{R}}_{xx} = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{x}(j)\boldsymbol{x}^H(j) \;\approx A\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\,\mathbf{I}$
- Take the M highest eigenvalues and the M eigenvectors belonging to those eigenvectors as signal part of space.
- Take the rest, $N - M$ eigenvalues and eigenvectors $(v_{M+1}\; v_{M+1}\; \dots .\, v_N)$, as noise part of space.
- Construct the noise matrix $\boldsymbol{E}_n = [v_{M+1}\; v_{M+1}\; \dots .\, v_N]$ which assures $\boldsymbol{A}^H vi = 0\,, i = M + 1 \dots .\, N$
- Scan θ according to the formula $P_{MUSIC} = \frac{1}{a(\theta)^H \boldsymbol{E}_n \boldsymbol{E}_n^H a(\theta)}$ and calculate the spatial spectrum.
- Obtain the estimated M values of DOAs by searching for the M highest peaks.

03

ROOT MUSIC

Root MUSIC

- Variant of MUSIC algorithm.
- MUSIC involves plotting the spatial spectrum against the angles and searching for the peaks which requires human interaction or a searching algorithm to decide on the M largest peaks. This is an extremely computationally intensive process.
- We would like a method that results directly in numeric values for estimated DOA.
- Root MUSIC involves finding the roots of the denominator of P_{MUSIC}.
- The denominator's roots will give us the desired peaks.

Implementation of Root MUSIC algorithm

- $P_{MUSIC} = \dfrac{1}{a(\theta)^H \boldsymbol{E}_n \boldsymbol{E}_n^H a(\theta)}$
- Defining $\boldsymbol{C} = \boldsymbol{E}_n \boldsymbol{E}_n^H$ we get $P_{MUSIC} = \dfrac{1}{a(\theta)^H \boldsymbol{C} a(\theta)}$
- The m_{th} element of the array steering vector is defined as $a_m(\theta) = e^{-jkdmsin(\theta)}$, $0 \leq m \leq N-1$.

Implementation of Root MUSIC algorithm

$$a(\theta)^H \boldsymbol{C} a(\theta) = [a_0(\theta)^*, a_1(\theta)^* \dots a_{N-1}(\theta)^*] \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & \dots c_{0,N-1} \\ c_{1,0} & c_{1,1} & c_{1,2} & \dots c_{1,N-1} \\ . & & . & . \\ . & & . & . \\ . & & . & . \\ c_{N-1,0} & c_{N-1,1} & \dots & c_{N-1,N-1} \end{bmatrix} \begin{bmatrix} a_0(\theta) \\ a_1(\theta) \\ . \\ . \\ . \\ a_{N-1}(\theta) \end{bmatrix} =$$

$$= a_0(\theta) \cdot \sum_{n=0}^{N-1} a_n(\theta)^* c_{n,0} + a_1(\theta) \cdot \sum_{n=0}^{N-1} a_n(\theta)^* c_{n,1} + \dots + a_{N-1}(\theta) \cdot \sum_{n=0}^{N-1} a_n(\theta)^* c_{n,N-1} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} a_m(\theta) a_n(\theta)^* c_{n,m} =$$

$$= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} c_{n,m} \cdot e^{jkd(n-m)sin(\theta)} \underbrace{=}_{l=m-n} \sum_{n=0}^{N-1} \sum_{l=-(N-1)}^{N-1} c_{n,n+l} \cdot e^{-jkdlsin(\theta)} = \sum_{l=-(N-1)}^{N-1} e^{-jkdlsin(\theta)} \cdot \underbrace{\sum_{n=0}^{N-1} c_{n,n+l}}_{\substack{c_l - sum\ of\ the\ l^{th} \\ diagonal\ of\ matrix\ \boldsymbol{C}}} =$$

$$= \sum_{l=-(N-1)}^{N-1} c_l \cdot e^{-jkdlsin(\theta)} = P_{MUSIC}^{-1}$$

Implementation of Root MUSIC algorithm

- Finding the roots of $P_{MUSIC}^{-1} = \sum_{l=-(N-1)}^{N-1} c_l \cdot e^{-jkdlsin(\theta)}$ will estimate the angles of arrivals.
- Setting $z = e^{-jkdsin(\theta)}$ will give us the Laurent polynomial $P_{MUSIC}^{-1} = \sum_{l=-(N-1)}^{N-1} c_l \cdot z^l$
- The degree of the polynomial is $2(N-1)$ thus we have $2(N-1)$ zeros for this equation.

- Not all zeros are independent. If z is a zero of P_{MUSIC}^{-1} then $\frac{1}{z^*}$ is also a zero:

$$P_{MUSIC}^{-1}(z) = \sum_{l=-(N-1)}^{N-1} c_l \cdot z^l = c_{-(N-1)} z^{-(N-1)} + \cdots + c_0 + \cdots + c_{N-1} z^{N-1} = 0$$

Since $\boldsymbol{C} = \boldsymbol{E}_n \boldsymbol{E}_n^H$ its a Hermitian matrix, $c_l = c_{-l}^*$.

$$P_{MUSIC}^{-1}\left(\frac{1}{z^*}\right) = \sum_{l=-(N-1)}^{N-1} c_l \cdot {z^*}^{-l} \underbrace{=}_{c_l = c_{-l}^*} \sum_{l=-(N-1)}^{N-1} c_{-l}^* \cdot {z^*}^{-l} = c_{-(N-1)}^* z^{-(N-1)^*} + \cdots + c_0^* + \cdots + c_{N-1}^* z^{N-1^*} =$$

$$= \left(c_{-(N-1)} z^{-(N-1)} + \cdots + c_0 + \cdots + c_{N-1} z^{N-1}\right)^* = \left(\sum_{l=-(N-1)}^{N-1} c_l \cdot z^l\right)^* = \Big(\underbrace{P_{MUSIC}^{-1}(z)}_{0}\Big)^* = 0$$

Implementation of Root MUSIC algorithm

- The zeros of P_{MUSIC}^{-1} comes in pairs.
- $z\ ,\frac{1}{z^*}$ have the same phase but different magnitude thus, one is inside the unit circle and the other outside the unit circle.
- Ideally, with no noise, M zeros should be on the unit circle since we are looking for $\theta's$ such that $P_{MUSIC}^{-1}(\theta) = 0$ and since $z = e^{-jkdsin(\theta)}$ we are looking for solutions that look like that – find θ such that $z = e^{-jkdsin(\theta)}$ gets $P_{MUSIC}^{-1}(z) = 0$.
- $|z| = \left|e^{-jkdsin(\theta)}\right| = 1$ → laying on the unit circle.

Implementation of Root MUSIC algorithm

- Since we are looking for θ , all the information is in the phase of the root.
- Representing the roots as magnitude and phase we get: $z_i = |z_i|e^{jarg(z_i)}$ where $arg(z_i)$ is the $kdsin(\theta)$ we are looking for.
- From $N-1$ roots that were found, we'll take the M roots that are closest to the unit circle ($|z_i| \approx 1$) and compare them to $z = e^{-jkdsin(\theta_i)}$ and get:

$$-kdsin(\theta_i) = \arg(z_i) \Rightarrow \theta_i = \sin^{-1}\left(\frac{-\arg(z_i)}{kd}\right)$$

$$\Downarrow$$

$$\theta_i = -\sin^{-1}\left(\frac{\arg(z_i)}{kd}\right), 1 \le i \le M$$

Root MUSIC - Summery

- Obtain the following estimation of the autocorrelation matrix based on the N received signal vectors: $\widetilde{\boldsymbol{R}}_{xx} = \frac{1}{J}\sum_{j=1}^{J}\boldsymbol{x}(j)\boldsymbol{x}^H(j) \approx A\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\,\mathbf{I}$.
- Take the $N - M$ smallest eigenvalues and their eigenvectors, as noise part of space.
- Get the noise matrix $\boldsymbol{E}_n = [v_{M+1}\ v_{M+1}\ \ldots .\ v_N]$.
- Construct $P_{MUSIC}^{-1} = a(\theta)^H\boldsymbol{C}a(\theta)$ where $\boldsymbol{C} = \boldsymbol{E}_n\boldsymbol{E}_n^H$.
- Obtain c_l by summing the elements on the l^{th} diagonal of $\boldsymbol{C}$.
- Find the zeros of $\sum_{l=-(N-1)}^{N-1} c_l \cdot z^l$ in terms of $N - 1$ pair roots.
- Out of $N - 1$ roots inside the unit circle, choose the M roots closest to the unit circle.
- Obtain the DOA of the M signals by the equation $\theta_i = -\sin^{-1}\left(\frac{\arg(z_i)}{kd}\right)$, $0 \le i \le M - 1$

04

ESPRIT

Estimation of Signal Parameters using Rotational Invariance Techniques

Implementation of ESPRIT algorithm

- Different technique of finding numeric values for estimated DOA.
- We will consider $z_i = e^{-jkdsin(\theta_i)}$
- The estimation of $\boldsymbol{R}_{xx}$ is $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\,\mathbf{I}$
- In this case $\boldsymbol{A} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_M \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ z_1^{N-1} & z_2^{N-1} & \dots & z_M^{N-1} \end{bmatrix}$ (we don't know $\boldsymbol{A}$ since we only have the estimation of $\boldsymbol{A}\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\,\mathbf{I}$)
- Based on $\boldsymbol{A}$ we'll define two $(N-1)xM$ matrices $\boldsymbol{A_0}$ and $\boldsymbol{A_1}$

$$\boldsymbol{A_0} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_M \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ z_1^{N-2} & z_2^{N-2} & \dots & z_M^{N-2} \end{bmatrix} \qquad \boldsymbol{A_1} = \begin{bmatrix} z_1 & z_2 & \dots & z_M \\ z_1^2 & z_2^2 & \dots & z_M^2 \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ z_1^{N-1} & z_2^{N-1} & \dots & z_M^{N-1} \end{bmatrix}$$

- Note that $\boldsymbol{A_1} = \boldsymbol{A_0}\boldsymbol{\Phi}$ where $\boldsymbol{\Phi}$ is the MxM rotational matrix: $\boldsymbol{\Phi} = \begin{bmatrix} z_1 & 0 & \dots & 0 \\ 0 & z_2 & \dots & 0 \\ . & . & & . \\ . & . & & . \\ . & & . & . \\ 0 & 0 & \dots & z_M \end{bmatrix}$

- $\boldsymbol{\Phi}$ is a diagonal matrix whose entries are exactly the phase shift from one sensor to the adjacent one due to each individual signal.

- $z_i = e^{-jkdsin(\theta_i)}$ - the phase shift from sensor j and it's adjacent of the i^{th} signal. If we have z_i we can extract θ_i since all the other parameters are known (k, d).

Implementation of ESPRIT algorithm

- We are interested in the diagonal values of $\boldsymbol{\Phi}$ which are his eigenvalues.
- Of course, if $\boldsymbol{A_0}$, $\boldsymbol{A_1}$ are known then we can find $\boldsymbol{\Phi}$ ($\boldsymbol{\Phi} = \mathbf{A_0^{-1}A_1}$) but they are unknown just like the matrix $\boldsymbol{A}$ is unknown.
- Recall that from $\boldsymbol{R_{xx}}$ we can extract $\boldsymbol{E_s} = [v_1, v_2 \dots v_M]$ which is NxM matrix constructed from the eigenvectors that span the signal subspace.
- The matrix $\boldsymbol{A}$ also spans the same subspace as $\boldsymbol{E_s}$ because her columns are the steering vectors of the M different signals, and they are independent.
- Since both these matrices span the same subspace, one matrix can be represented by the other : $\boldsymbol{E_s} = \boldsymbol{AC}$ where $\boldsymbol{C}$ is invertible.

Implementation of ESPRIT algorithm

- Defining $\boldsymbol{E_0}, \boldsymbol{E_1}$ just like $\boldsymbol{A_0}, \boldsymbol{A_1}$ we get that $\boldsymbol{E_0}$ is the first $N-1$ rows of $\boldsymbol{E_s}$ and $\boldsymbol{E_1}$ is the last $N-1$ rows of $\boldsymbol{E_s}$.
- Using the last equation from the last slide we get that $\boldsymbol{E_0} = \boldsymbol{A_0 C}, \boldsymbol{E_1} = \boldsymbol{A_1 C} = \boldsymbol{A_0 \Phi C}$
- Consider $\boldsymbol{E_1 C^{-1} \Phi^{-1} C} = \boldsymbol{A_0 \Phi C C^{-1} \Phi^{-1} C} = \boldsymbol{A_0 C} = \boldsymbol{E_0}$ $(*)$
- Now let $\boldsymbol{\Psi^{-1}} = \boldsymbol{C^{-1} \Phi^{-1} C} \underbrace{\Rightarrow}_{(*)} \boldsymbol{E_1 \Psi^{-1}} = \boldsymbol{E_0} \Rightarrow \boldsymbol{E_1} = \boldsymbol{E_0 \Psi} \Rightarrow \boldsymbol{\Psi} = \boldsymbol{E_0^{-1} E_1}$
- $\boldsymbol{\Psi^{-1}} = \boldsymbol{C^{-1} \Phi^{-1} C} \Rightarrow \boldsymbol{\Psi} = \boldsymbol{C^{-1} \Phi C}$ and this is the pattern of eigenvalue decomposition , so this equation implies that $\boldsymbol{\Phi}$ is a diagonal matrix whose diagonal elements are $\boldsymbol{\Psi}$'s eigenvalues.

- Recall that we are looking for the elements of $\boldsymbol{\Phi}$ and since we have $\boldsymbol{\Psi} = \boldsymbol{E_0^{-1}E_1}$ and $\boldsymbol{E_s}$ that we know how to compute, all that is left is to find the M eigenvalues of $\boldsymbol{\Psi}$ and we have the solutions.
- Now that we have $z_1, z_2 \;\dots. z_M$ which are the eigenvalues $\lambda_1, \lambda_2 \dots \lambda_M$ of $\boldsymbol{\Psi}$ we can find the angle of arrival by comparing λ_i to $e^{-jkdsin(\theta_i)}$.
- $\lambda_i = |\lambda_i| e^{jarg(\lambda_i)}, |\lambda_i| \approx 1 \Rightarrow \lambda_i \approx e^{jarg(\lambda_i)}$ and comparing to $e^{-jkdsin(\theta_i)}$ we get:

$$\theta_i = sin^{-1}\left(\frac{-arg(\lambda_i)}{kd}\right)$$
$$\Downarrow$$
$$\theta_i = -sin^{-1}\left(\frac{arg(\lambda_i)}{kd}\right), 1 \le i \le M$$

ESPRIT - Summery

- Obtain the following estimation of the autocorrelation matrix based on the N received signal vectors: $\widetilde{\boldsymbol{R}}_{xx} = \frac{1}{J}\sum_{j=1}^{J}\boldsymbol{x}(j)\boldsymbol{x}^H(j) \approx A\boldsymbol{R}_{ss}\boldsymbol{A}^H + \sigma_0^2\,\mathbf{I}$.
- Take the M highest eigenvalues and the M eigenvectors belonging to those eigenvalues as signal part of space and build $\boldsymbol{E_s} = [v_1, v_2 \ldots v_M]$.
- Obtain $\boldsymbol{E_0}, \boldsymbol{E_1}$ to estimate the MxM matrix $\boldsymbol{\Psi} = \boldsymbol{E_0^{-1}E_1}$.
- Find the eigenvalues of $\boldsymbol{\Psi}$ which are the values that estimate $z_i = e^{-jkdsin(\theta_i)}$.
- Obtain DOA by $\theta_i = -sin^{-1}\left(\frac{arg(\lambda_i)}{kd}\right), 1 \le i \le M$
- Note: ESPRIT has greater computation load than MUSIC. This is because we need two eigen decompositions , one from the autocorrelation matrix to find $\boldsymbol{E_s}$ and the second for estimating the eigenvalues of $\boldsymbol{\Psi}$.

05

Simulation

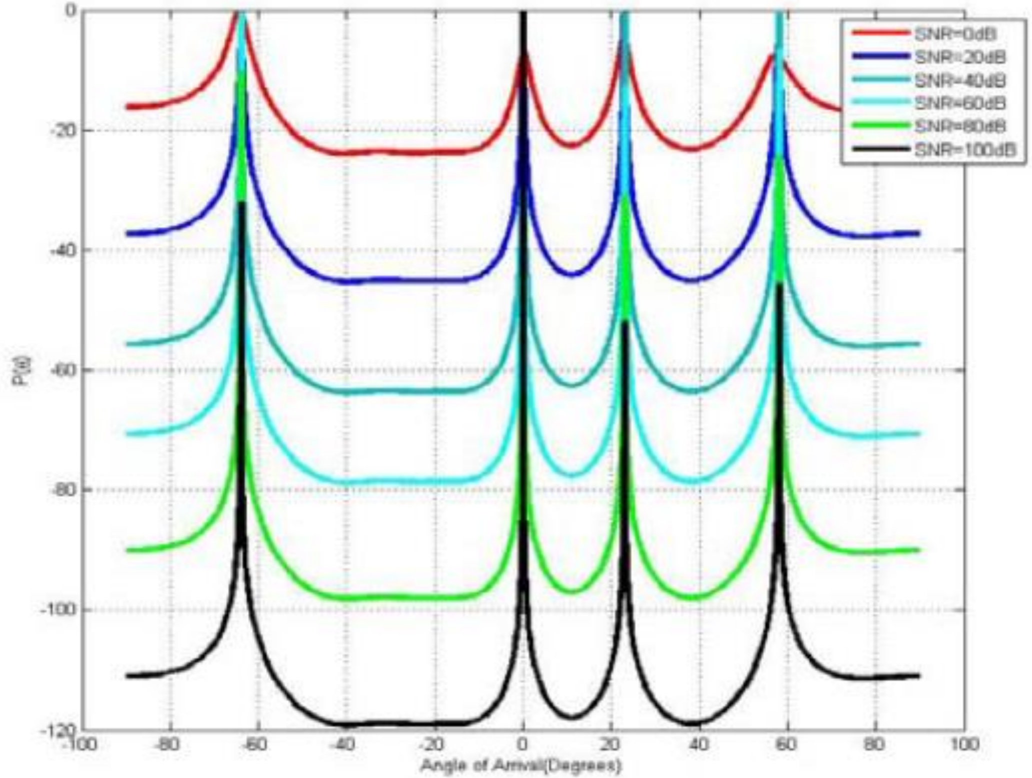
Simulation & Analysis

- Simulations in the article were done in MATLAB for angles of arrival $\theta_1 = -64° , \theta_2 = 0° , \theta_3 = 23° , \theta_4 = 58°$.
- $f = 2.4\ [GHz] -$ main frequency of impinging signals.
- The array size was held to 8 sensors as the values of SNR were varied from 0-100 dB in steps of 20 dB.
- This was repeated holding SNR to 50dB and varying the array size from 5-100 sensors.
- Results for varying snapshots are represented as well.
- Simulation of two closely spaced angles of arrival is also shown.

06

Results

MUSIC for varying SNR



MUSIC for low SNR

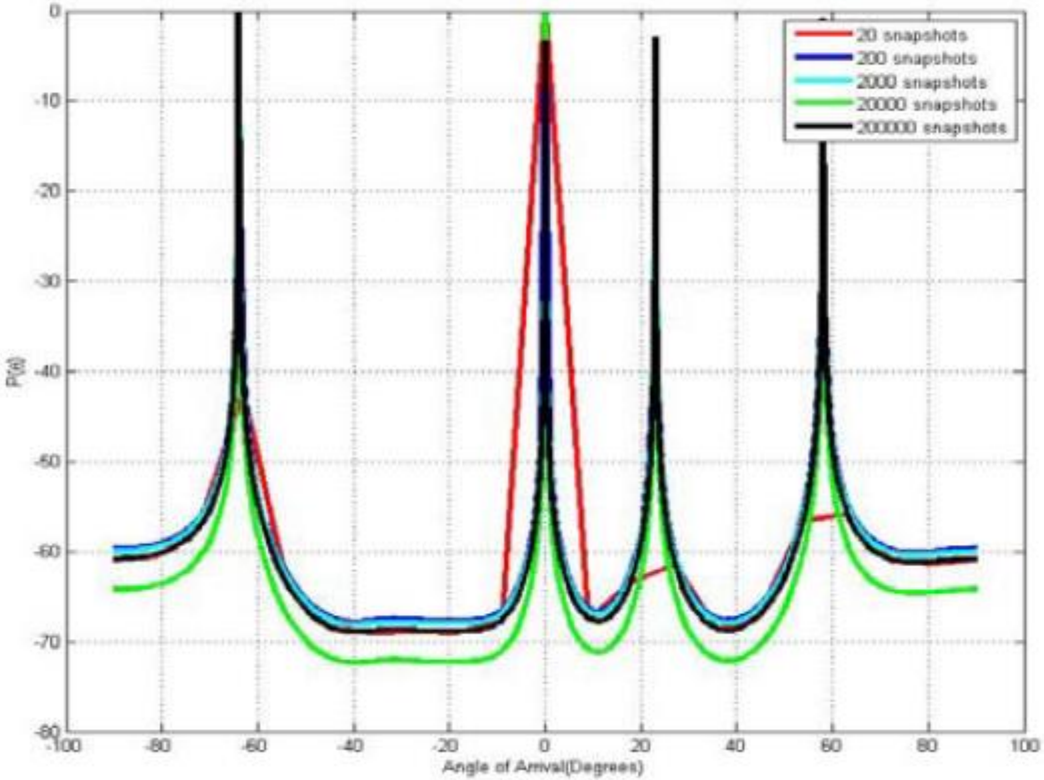
- For low SNR – 0dB the spikes depicting the arrival of a signal from certain directions are small and the response is almost flat. It is thus difficult to exactly extract the angles of arrival.
- This is attributed to the fact that for low SNR the difference between the eigenvalues associated with the signal $(\lambda_1 + \sigma_0^2, \lambda_2 + \sigma_0^2 \ldots \lambda_M + \sigma_0^2)$ and those associated with the noise (σ_0^2) became smaller and the peaks therefore became smaller with respect to the noise levels.
- Low SNR makes it difficult to estimate the eigenvectors of the signal since the eigenvalues of the signal and noise are close to each other. This may add /subtract noise eigenvectors to $\boldsymbol{E_n}$ therefore smearing the spatial spectrum and getting low resolution.

MUSIC for high SNR

- With increase in SNR, the difference between the eigenvalues corresponding to the signal and noise increase which causes the peaks to be bigger with respect to the noise levels.

  $(\lambda_i + \sigma_0^2 \gg \sigma_0^2)$.
- As the values of SNR increase the resolution of the algorithm is observed to improve considerably and the spikes became more definite.
- The cause of this is because it’s easier to separate the eigenvectors corresponding to the signal and the ones corresponding to the noise with small number of errors so the set $\boldsymbol{E_n}$ is what it should be, so only real angles of arrival will be orthogonal to $\boldsymbol{E_n}$ which can’t be promised when SNR is low.

MUSIC for varying snapshots



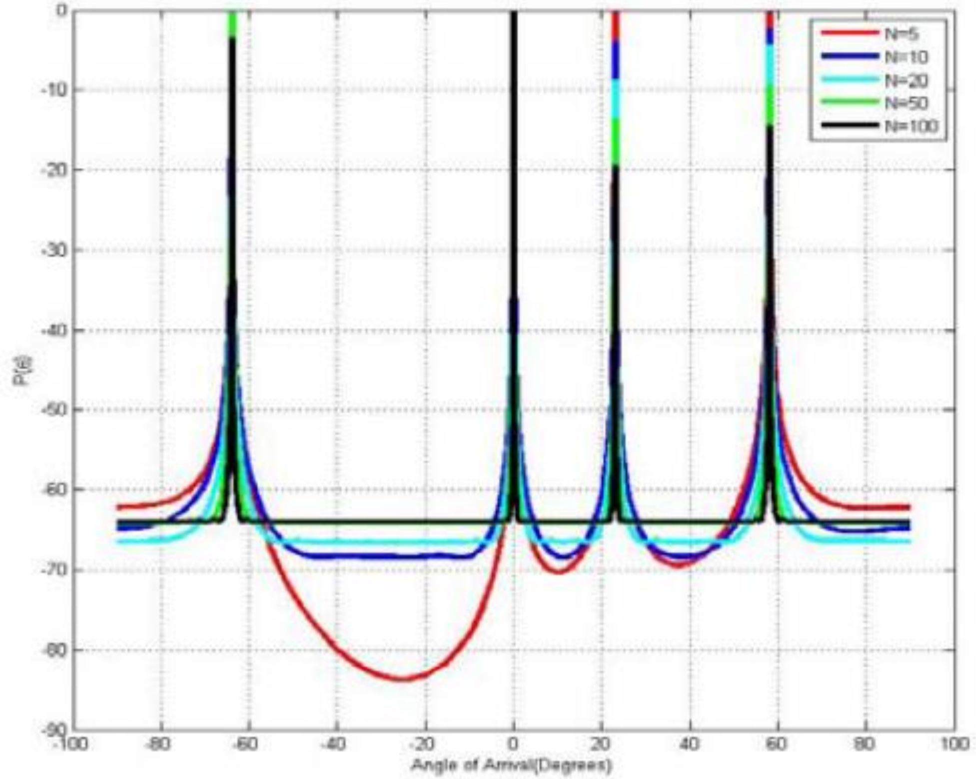
MUSIC for low amount of snapshots

- As expected, since we need to estimate $\boldsymbol{R}_{xx}$ in order to perform MUSIC algorithm by $\widetilde{\boldsymbol{R}}_{xx} = \frac{1}{J}\sum_{j=1}^{J}\boldsymbol{x}(j)\boldsymbol{x}^H(j)$ where $\widetilde{\boldsymbol{R}}_{xx}$ is the ML estimate of $\boldsymbol{R}_{xx}$ we need large number of snapshots in order to get a good estimation of $\boldsymbol{R}_{xx}$: $\widetilde{\boldsymbol{R}}_{xx} \rightarrow \boldsymbol{R}_{xx}$ for $J \rightarrow \infty$ where J is the number of snapshots.
- For small number of snapshots $\widetilde{\boldsymbol{R}}_{xx} \neq \boldsymbol{R}_{xx}$ and the eigenvectors of the signal and noise subspaces aren't the expected ones which causes poor resolution and estimation of DOA since we work with $\widetilde{\boldsymbol{R}}_{xx}$ that doesn't give enough information of about the steering vectors $a(\theta_i)$.
- It can be seen from the figure that for 20 snapshots the response has less spikes.
- The number of snapshots affect the correlation between the received signals. For less snapshots, the received signals seem more correlated making it difficult to distinguish between them.

MUSIC for high amount of snapshots

- For large number of snapshots, $\boldsymbol{R}_{xx}$ is well estimated ($\widetilde{\boldsymbol{R}}_{xx} \rightarrow \boldsymbol{R}_{xx}$) and the construction of the two subspaces is much more accurate.
- Large number of snapshots gives more information about the uncorrelated signals which makes it easier to distinguish between them.
- Since $\boldsymbol{R}_{xx}$ is well estimated when the number of snapshots is high, the resolution is seen to improve with increase in the number of snapshots.

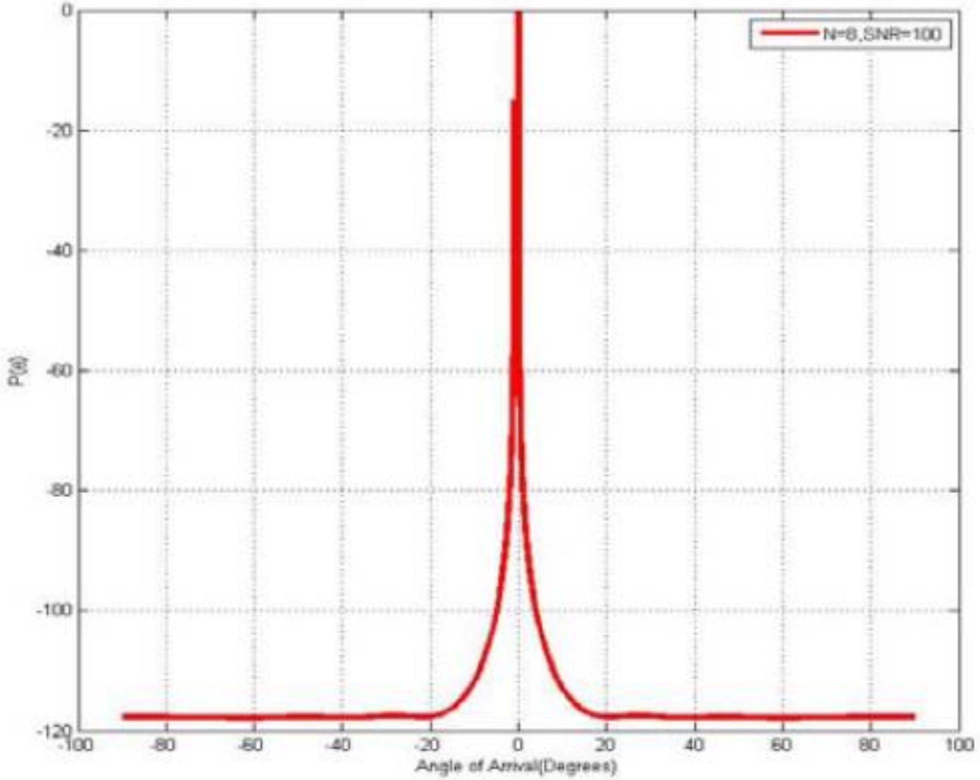
MUSIC for varying array size



MUSIC for varying array size

- As we can see, for 5 elements the spikes are very definite and exactly correspond to the angles of arrival and that is because we have 4 signals, so this problem is an overdetermined problem (more equations than unknown angles of arrival).
- As the array size increases, we still get M eigenvalues correspond to the number of impinging signals but $N - M$ eigenvalues for the noise. When N increases, the noise subspace $\boldsymbol{E_n}$ has more eigenvectors thus for a steering vector to be orthogonal to the noise subspace it has to maintain orthogonality for more vectors (each column vector in $\boldsymbol{E_n}$).
- That means the number of steering vectors maintaining this condition decreases and equal to the number of steering vectors of the impinging signals which are represented in the matrix $\boldsymbol{A}$ and maintain that condition. In conclusion, more elements gives better resolution.

MUSIC for two closely spaced signals



MUSIC for two closely spaced signals

- This simulation was tested for $N = 8,\ SNR = 100dB,\ \theta_1 = -1°, \theta_2 = 0°$.
- We have two signals impinging from two different close directions.
- Since the noise subspace is constructed from 6 eigenvectors and there are only two angles of arrival, and the SNR is very high it’s easier for MUSIC to construct the exact noise subspace and exactly extract the angles of arrival making MUSIC a high-resolution DOA algorithm.

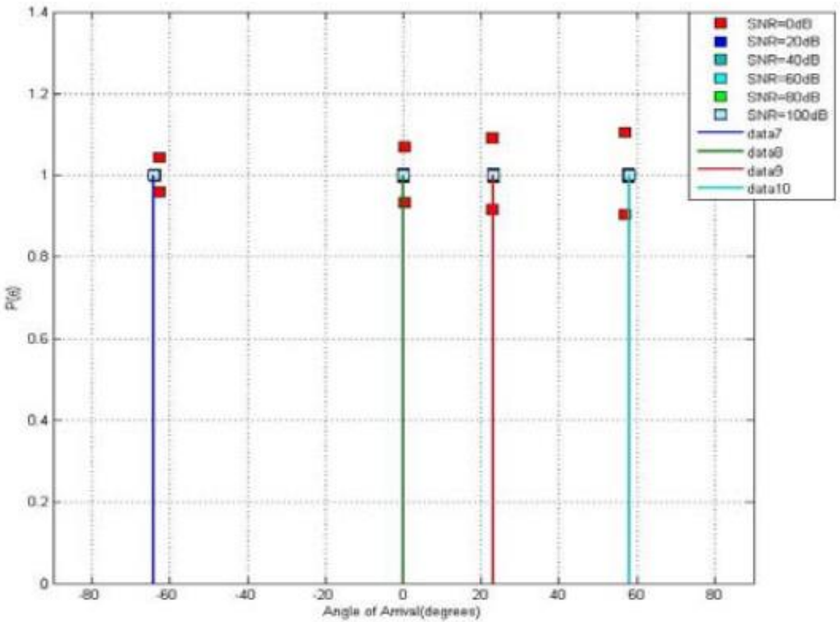


Figure 1

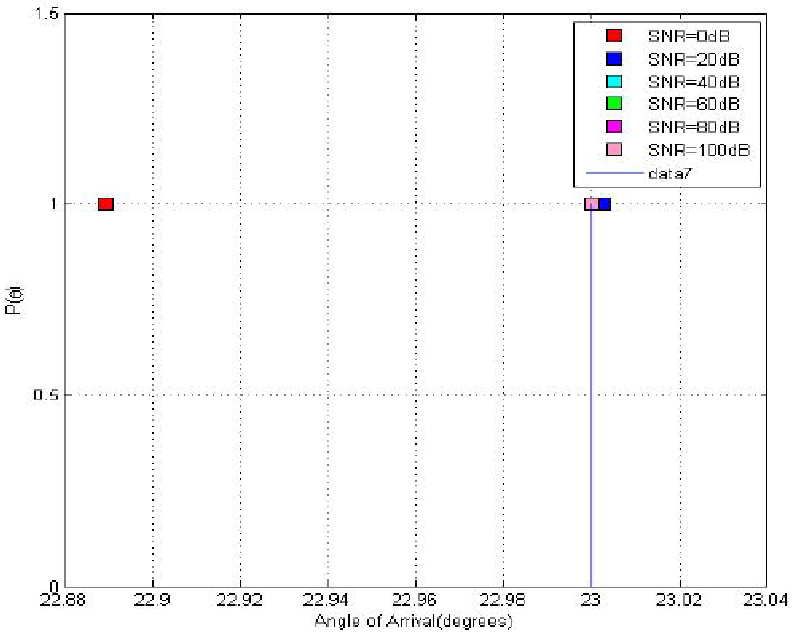


Figure 2

- Fig 1 - Squares represent $2(N-1)$ roots of the equation $a(\theta)^H \boldsymbol{C} a(\theta)$. $(N = 5)$
- Fig 1 - blue, green, red, azure lines represent what the actual angles of arrival are.

Root MUSIC for varying SNR

- As we can see, for low SNR we get that the roots of the equation $a(\theta)^H \boldsymbol{C} a(\theta)$ are not on the unit circle because of the dominance of the noise (marked as the red squares).
- Out of the $2N - 2$ roots inside the unit circle, choosing the M roots closest to the unit circle shows us that the phase of those roots are not giving an accurate angle of arrival with respect to the four lines in the figure above (the roots are shifted from the lines).
- For high SNR, the noise is neglectable and we get that all roots of $a(\theta)^H \boldsymbol{C} a(\theta)$ lay on the unit circle and each pair is equal (which is obtained only when on the unit circle: $z = \frac{1}{z^*} \Leftrightarrow |z|^2 = 1$)
- With increase in SNR, we saw that the resolution of MUSIC improves which affect and improves the resolution of Root MUSIC since it relies on MUSIC.

Root MUSIC for varying array size

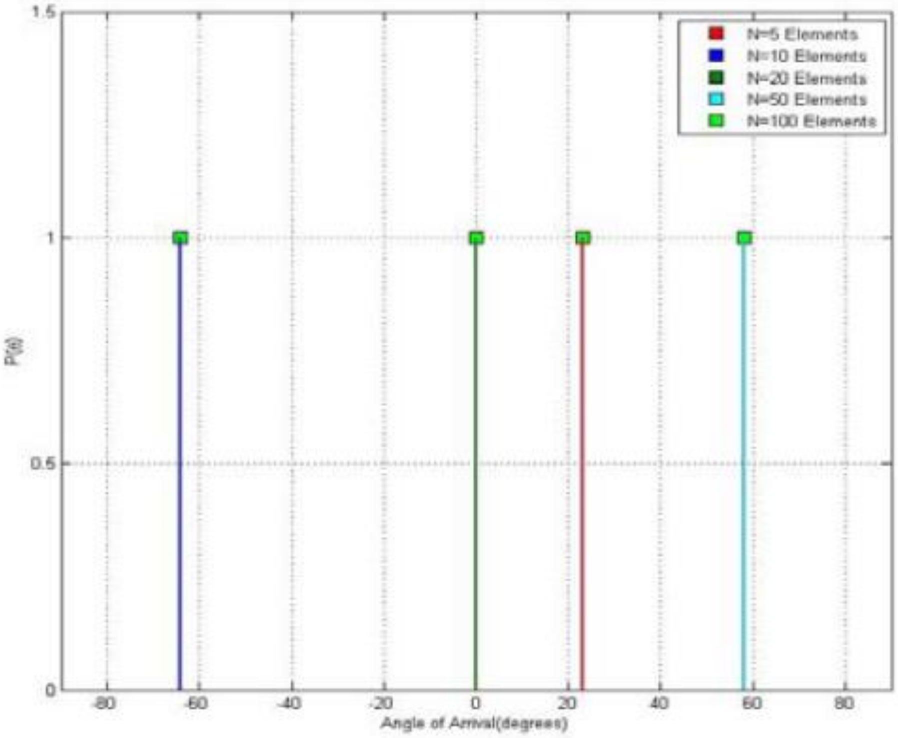


Figure 1

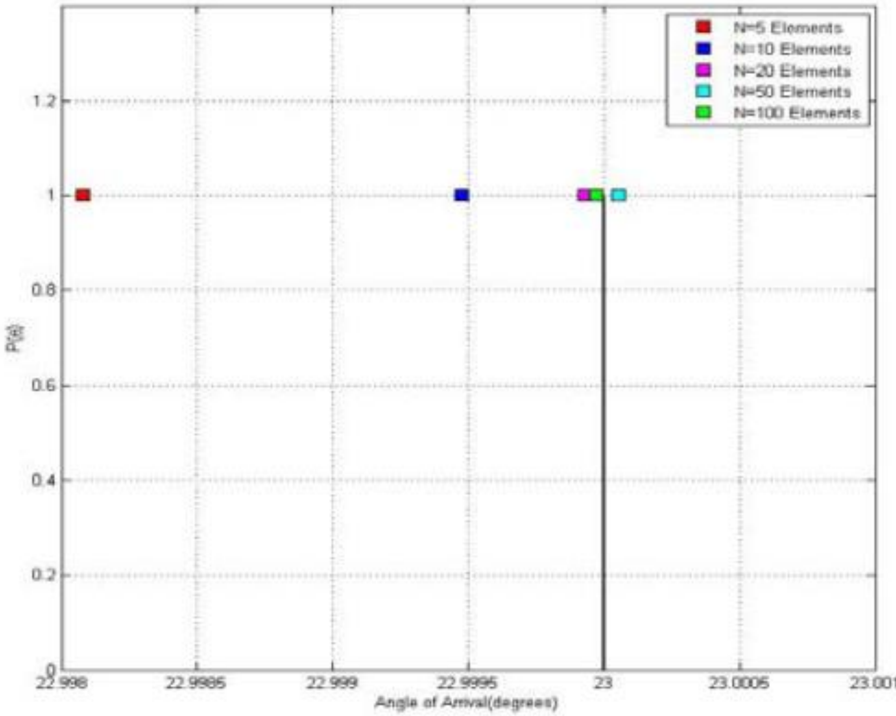
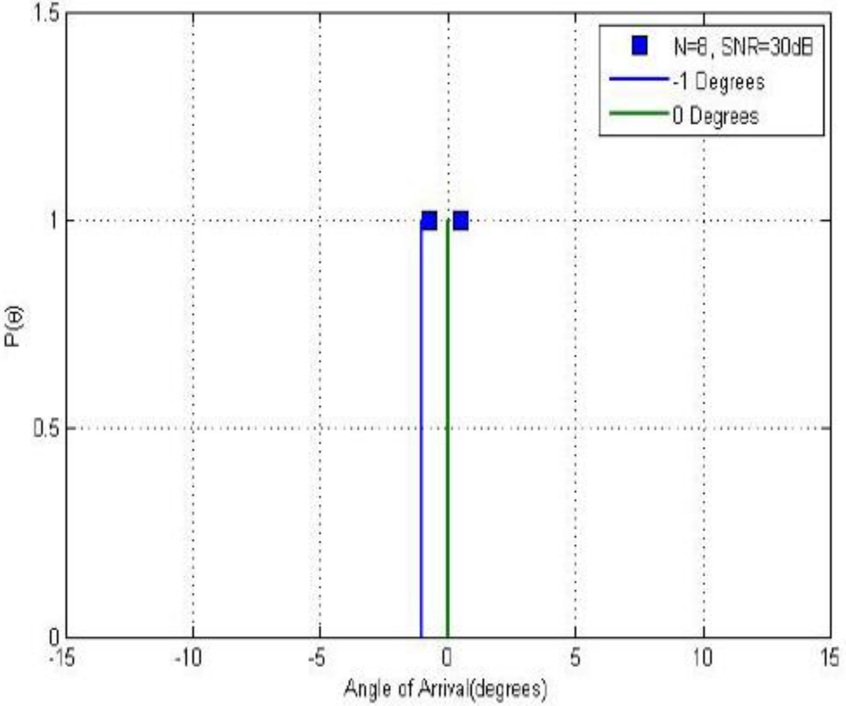


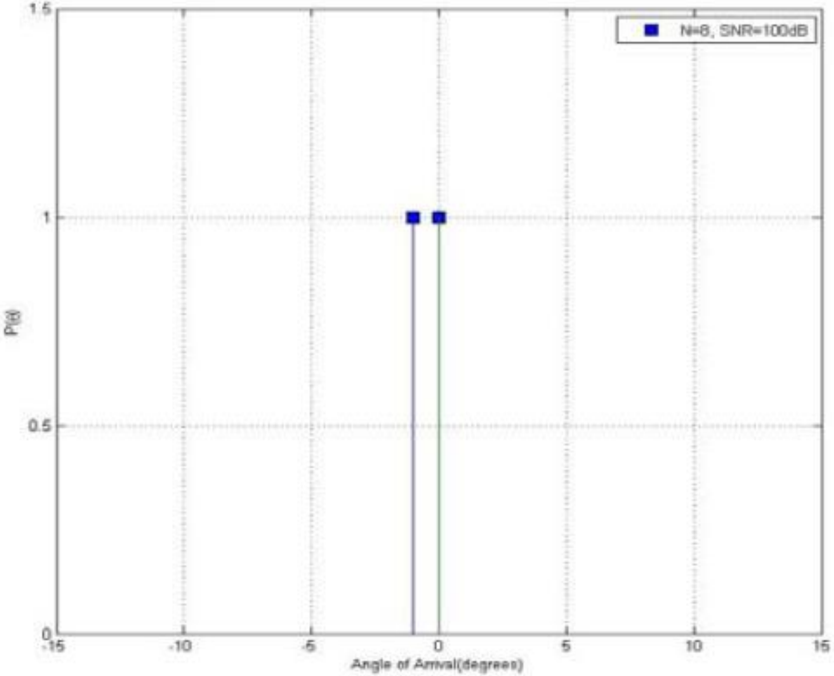
Figure 2

Root MUSIC for varying array size

- Just like in MUSIC , five elements was enough to determine four angles of arrival, and that is why we barely see any difference between the estimation (roots of the equation) for 5 elements and 100 elements just like shown in figure 1 and more clearly in figure 2.
- In MUSIC the peaks don't move for varying elements, but the resolution gets higher for large number of elements (width of spike is smaller).
- Here, since number of elements don't cause movement of spikes, the roots don't move as well, and width of spike doesn't concern us since we are looking for a numeric value for the angle and not a graph representing the spatial spectrum like in MUSIC.

Root MUSIC for two closely spaced signals





Root MUSIC for two closely spaced signals

- This simulation was tested for $N = 8,\ SNR = 30, 100dB,\ \theta_1 = -1°, \theta_2 = 0°$.
- We have two signals impinging from two different close directions.
- From the results ROOT-MUSIC is a high-resolution DOA algorithm which can estimate two closely spaced angles of arrival as two angles and not one.
- Since the noise subspace is constructed from 6 eigenvectors and there are only two angles of arrival, and the SNR is high it's easier for MUSIC to construct the exact noise subspace and for ROOT-MUSIC to estimate the DOA.

08

Conclusions & summery

Conclusions & summery

- This presentation introduces three high-resolution DOA estimation algorithms: MUSIC, ROOT MUSIC and ESPRIT.
- Full implementations of the algorithms are provided at each chapter respectively.
- Simulations and performances were also introduced.
- Each algorithm has its advantages, disadvantages and complexity thus, each algorithm should be picked wisely.

Bibliography

Web:

- MATLAB explanation of MUSIC: https://www.mathworks.com/help/phased/ug/music-super-resolution-doa-estimation.html
- https://www.vocal.com/beamforming-2/music-algorithm/
- Proof that eigenvectors of Hermitian matrix are orthogonal : https://math.stackexchange.com/questions/762984/whats-the-proof-stategy-for-hermitian-matrix-has-orthogonal-eigenvectors-for-d
- Proof that eigenvalues of Hermitian matrix are real: https://yutsumura.com/eigenvalues-of-a-hermitian-matrix-are-real-numbers/
- https://en.wikipedia.org/wiki/MUSIC_(algorithm)
- https://en.wikipedia.org/wiki/Vandermonde_matrix
- https://en.wikipedia.org/wiki/Estimation_of_signal_parameters_via_rotational_invariance_techniques

Reference articles:

- Performance Analysis of MUSIC Root MUSIC and ESPRIT DOA Estimation Algorithm
- Application of DOA Estimation Algorithms in smart antennas systems (Tanuja S. Dhope (Shendkar), Dina Simunic, Marijan Djurek)
- Direction of Arrival Estimation using a Root-MUSIC Algorithm (H.K. Hwang, Anatoly Yakovlev)
- Direction of Arrival Estimation
- DOA estimation based on MUSIC algorithm
- Polynomial Root-MUSIC Algorithm for Efficient broadband direction of arrival estimation (William Coventry, Carmine Clemente, and John Soraghan)