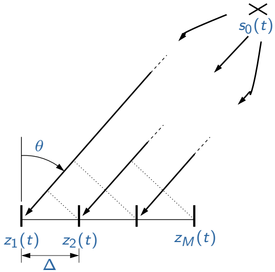
EE 4715 Array Processing

8. Direction Estimation using the ESPRIT Algorithm

April 2022

Problem

From the samples at the output of a **uniform linear antenna array**, estimate the angles of arrival of the impinging signals.



$$\boldsymbol{Z} = \boldsymbol{ABS} = \boldsymbol{a}(\theta_1)\beta_1\boldsymbol{s}_1 + \boldsymbol{a}(\theta_2)\beta_2\boldsymbol{s}_2 + \cdots$$

A general procedure for DOA estimation is parametric fitting, e.g.

$$\underset{\{\theta_i\},\boldsymbol{B},\boldsymbol{S}}{\arg\min}\ \|\boldsymbol{Z} - \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{BS}\|_F^2\,, \quad \text{where} \quad \boldsymbol{A}(\boldsymbol{\theta}) = [\boldsymbol{a}(\theta_1)\ \ \boldsymbol{a}(\theta_2)\cdots]$$

We have to know the array manifold $\boldsymbol{a}(\theta)$.

The ESPRIT algorithm

For a uniform linear array with element spacing Δ wavelengths

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{M-1} \end{bmatrix} \quad \begin{matrix} \text{first } M-1 \text{ entries: } \mathbf{a}_x \\ \text{last } M-1 \text{ entries: } \mathbf{a}_y \end{matrix} \qquad \phi = e^{j2\pi\Delta\sin(\theta)}$$

We will exploit the **shift-invariance property**:

$$\mathbf{a}_x := \begin{bmatrix} 1 \\ \phi \\ \vdots \\ \phi^{M-2} \end{bmatrix}, \quad \mathbf{a}_y := \begin{bmatrix} \phi \\ \phi^2 \\ \vdots \\ \phi^{M-1} \end{bmatrix}, \quad \text{so that} \quad \mathbf{a}_y = \mathbf{a}_x \phi$$

The ESPRIT algorithm

Let us group the first and last $M-1$ antennas:

$$\boldsymbol{x}(t) = \begin{bmatrix} z_1(t) \\ \vdots \\ z_{M-1}(t) \end{bmatrix}, \quad \boldsymbol{y}(t) = \begin{bmatrix} z_2(t) \\ \vdots \\ z_M(t) \end{bmatrix}, \qquad \begin{array}{rcl} \boldsymbol{X} & = & [\boldsymbol{x}(1) \ \cdots \ \boldsymbol{x}(N)] \\ \boldsymbol{Y} & = & [\boldsymbol{y}(1) \ \cdots \ \boldsymbol{y}(N)] \end{array}$$

Due to shift-invariance, we have the models

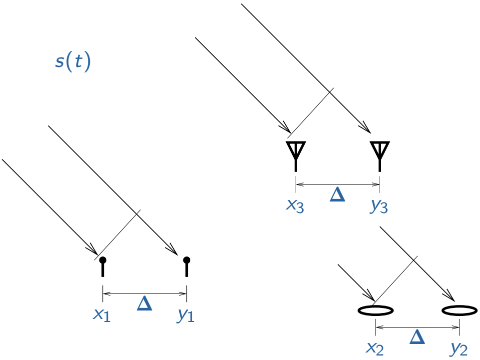
$$\begin{array}{rcll} \boldsymbol{x}(t) & = & \sum_{k=1}^{d} \boldsymbol{a}_x(\theta_k)\beta_k s_k(t) & \Rightarrow \ \boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} \\ \boldsymbol{y}(t) & = & \sum_{k=1}^{d} \boldsymbol{a}_y(\theta_k)\beta_k s_k(t) = \sum_{k=1}^{d} \boldsymbol{a}_x(\theta_k)\phi_k\beta_k s_k(t) & \Rightarrow \ \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Theta}\boldsymbol{S} \end{array}$$

where

$$\boldsymbol{A} = [\boldsymbol{a}_x(\theta_1) \ \cdots \ \boldsymbol{a}_x(\theta_d)], \quad \boldsymbol{\Theta} = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_d \end{bmatrix}, \quad \phi_k = e^{j2\pi\Delta \sin(\theta_k)}$$

The ESPRIT algorithm

This model also holds for the more general "doublet" antenna structure



$$x_k(t) = a_k \beta_k s(t)\,, \quad y_k = a_k e^{j2\pi\Delta\sin(\theta_k)}\beta_k s(t)$$

$$\boldsymbol{z}(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{y}(t) \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A\Theta} \end{bmatrix} \boldsymbol{BS}$$

The ESPRIT algorithm

Given the data matrix $\boldsymbol{Z}$ from all antennas

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} = \boldsymbol{A}_z \boldsymbol{B} \boldsymbol{S}\,, \quad \boldsymbol{A}_z = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}\boldsymbol{\Theta} \end{bmatrix}$$

- Note that $\boldsymbol{Z}$ has rank d. Compute an SVD of $\boldsymbol{Z}$:

  $$\boldsymbol{Z} = \hat{\boldsymbol{U}}_z \hat{\Sigma}_z \hat{\boldsymbol{V}}_z^{\mathrm{H}}$$

  $\hat{\boldsymbol{U}}_z : 2M \times d$ has d columns that span the column space of $\boldsymbol{Z}$.

- $\hat{\boldsymbol{U}}_z$ spans the same space as $\boldsymbol{A}_z$, hence there exists a $d \times d$ matrix $\boldsymbol{T}$:

  $$\hat{\boldsymbol{U}}_z = \boldsymbol{A}_z \boldsymbol{T} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{T} \\ \boldsymbol{A}\boldsymbol{\Theta}\boldsymbol{T} \end{bmatrix}$$

- Split $\boldsymbol{Z}$ into $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $\hat{\boldsymbol{U}}_z$ accordingly into $\hat{\boldsymbol{U}}_x$ and $\hat{\boldsymbol{U}}_y$, then

  $$\begin{cases} \hat{\boldsymbol{U}}_x = \boldsymbol{A}\boldsymbol{T} \\ \hat{\boldsymbol{U}}_y = \boldsymbol{A}\boldsymbol{\Theta}\boldsymbol{T} \end{cases}$$

The ESPRIT algorithm

- Note that $\hat{\boldsymbol{U}}_x^\dagger = (\boldsymbol{T}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{T})^{-1}\boldsymbol{T}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{T}^{-1}\boldsymbol{A}^\dagger$ so that

$$\hat{\boldsymbol{U}}_x^\dagger \hat{\boldsymbol{U}}_y = \boldsymbol{T}^{-1}\boldsymbol{\Theta}\boldsymbol{T}\,.$$

  Thus, $\boldsymbol{T}^{-1}$ and $\boldsymbol{\Theta}$ are given by the eigendecomposition of $\hat{\boldsymbol{U}}_x^\dagger \hat{\boldsymbol{U}}_y$. From $\boldsymbol{\Theta}$ we find $\{\phi_k\}$ and hence $\{\theta_k\}$.

- From $\boldsymbol{T}$ we can compute a zero-forcing beamformer on $\boldsymbol{Z}$ (such that $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{Z} = \boldsymbol{B}\boldsymbol{S}$) as

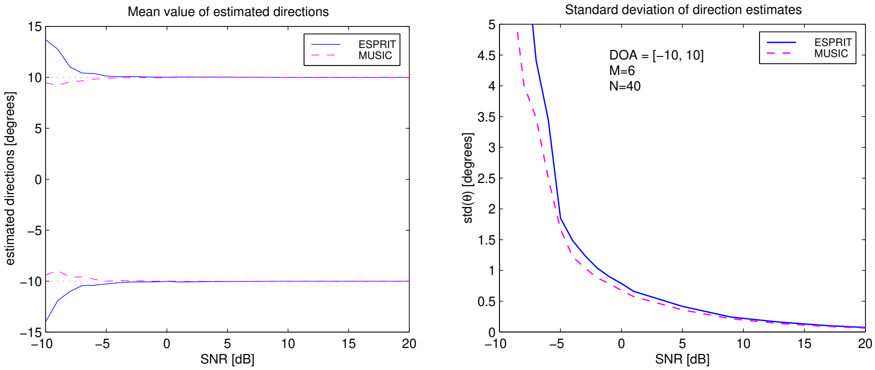
$$\boldsymbol{W} = \hat{\boldsymbol{U}}_z \boldsymbol{T}^{\mathrm{H}}$$

Proof:

$$\begin{aligned}
& \boldsymbol{Z} = \hat{\boldsymbol{U}}_z \cdot \hat{\boldsymbol{\Sigma}}_z \hat{\boldsymbol{V}}_z^{\mathrm{H}}\,, \qquad \boldsymbol{Z} = \boldsymbol{A}_z\boldsymbol{B}\boldsymbol{S} = \boldsymbol{A}_z\boldsymbol{T}\cdot\boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{S} \\
\Rightarrow\quad & \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{S} = \hat{\boldsymbol{\Sigma}}_z \hat{\boldsymbol{V}}_z^{\mathrm{H}} = \hat{\boldsymbol{U}}_z^{\mathrm{H}}\boldsymbol{Z} \\
\Rightarrow\quad & \boldsymbol{B}\boldsymbol{S} = \boldsymbol{T}\hat{\boldsymbol{U}}_z^{\mathrm{H}}\boldsymbol{Z}
\end{aligned}$$

The ESPRIT algorithm

Performance (varying SNR)

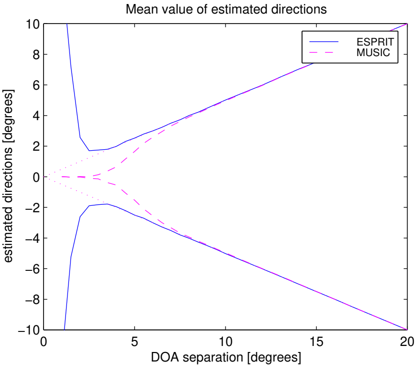


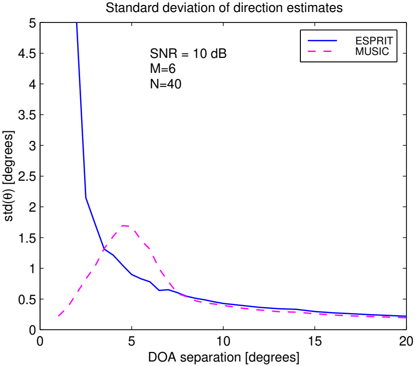
$d = 2$ sources, $M = 6$ antennas, $N = 40$ samples, 20° separation.

Mean and standard deviations of ESPRIT and MUSIC estimates as function of SNR. ESPRIT is slightly worse but much easier to compute (no search)

The ESPRIT algorithm

Performance (varying separation)





Mean and standard deviations of ESPRIT and MUSIC estimates as function of DOA separation.

Delay estimation using ESPRIT

Principle

Assume a vector $\boldsymbol{g}_0$ corresponding to an FIR pulse shape function $g(t)$, nonzero on interval $[0, L_g)$. Sample at rate P times the symbol rate:

$$g(t) \quad \leftrightarrow \quad \boldsymbol{g}_0 = \begin{bmatrix} g(0) \\ g(\frac{1}{P}) \\ \vdots \\ g(L - \frac{1}{P}) \end{bmatrix}$$

Let $\boldsymbol{g}_\tau$ denote the sample vector from a delayed version of $g(t)$:

$$g(t-\tau) \quad \leftrightarrow \quad \boldsymbol{g}_\tau = \begin{bmatrix} g(0-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(L - \frac{1}{P} - \tau) \end{bmatrix}$$

Given $\boldsymbol{g}_\tau$ and knowing $\boldsymbol{g}_0$, how do we estimate τ?

Delay estimation using ESPRIT

Use fact that a Fourier transform maps a delay to a phase shift

Apply the DFT to $\boldsymbol{g}_0$ and to $\boldsymbol{g}_\tau$ (LP samples):

$$\tilde{g}_0(\omega_i) := \sum_k e^{-j\omega_i k} g(\frac{k}{P}) \,, \quad \omega_i = i\frac{2\pi}{LP}$$

$$\tilde{\boldsymbol{g}}_0 := \mathcal{F}\,\boldsymbol{g}_0 \,, \quad \mathcal{F} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \phi & \cdots & \phi^{LP-1} \\ \vdots & \vdots & & \vdots \\ 1 & \phi^{LP-1} & \cdots & \phi^{(LP-1)^2} \end{bmatrix} , \quad \phi = e^{-j\frac{2\pi}{LP}} \,.$$

Property ($\odot$ represents entrywise multiplication):

$$\tilde{\boldsymbol{g}}_\tau := \mathcal{F}\,\boldsymbol{g}_\tau \; = \; \tilde{\boldsymbol{g}}_0 \odot \begin{bmatrix} 1 \\ \phi^{\tau P} \\ (\phi^{\tau P})^2 \\ \vdots \\ (\phi^{\tau P})^{LP-1} \end{bmatrix} \; = \; \mathrm{diag}(\tilde{\boldsymbol{g}}_0) \cdot \begin{bmatrix} 1 \\ \phi^{\tau P} \\ (\phi^{\tau P})^2 \\ \vdots \\ (\phi^{\tau P})^{LP-1} \end{bmatrix}$$

Delay estimation using ESPRIT

- Invert the pulse (deconvolution)

$$\boldsymbol{z} := \{\text{diag}(\tilde{\boldsymbol{g}}_0)\}^{-1}\tilde{\boldsymbol{g}}_\tau$$

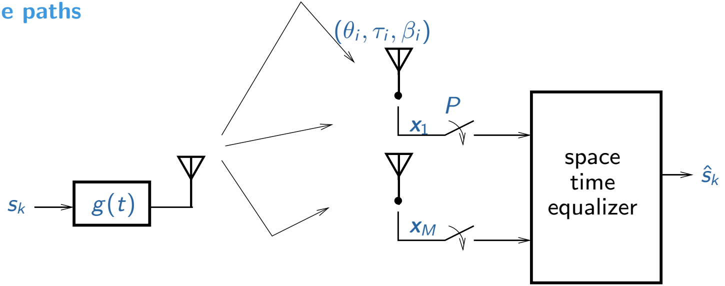
- The vector $\boldsymbol{z}$ has model

$$\boldsymbol{z} = \boldsymbol{f}(\phi)\,, \quad \boldsymbol{f}(\phi) := \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{LP-1} \end{bmatrix}, \quad \phi := e^{j2\pi\tau/L}$$

Now apply ESPRIT to compute ϕ and then τ.

Delay estimation using ESPRIT

Multiple paths



Consider a multipath channel which consists of r delayed copies of $g(t)$

$$h(t) = \sum_{i=1}^{r} \beta_i g(t - \tau_i) \ \Leftrightarrow\ \boldsymbol{h} = \sum_{i=1}^{r} \boldsymbol{g}_{\tau_i} \beta_i = [\boldsymbol{g}_{\tau_1}, \cdots, \boldsymbol{g}_{\tau_r}] \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_r \end{bmatrix} =: \boldsymbol{G}_\tau \boldsymbol{\beta}$$

Assume that the pulse shape $g(t)$ is known and that $\boldsymbol{h}$ has been estimated

Delay estimation using ESPRIT

As before:

$$\boldsymbol{z} := \{\mathrm{diag}(\tilde{\boldsymbol{g}}_0)\}^{-1}\mathcal{F}\boldsymbol{h}\,, \quad (LP \times 1)$$

- The vector $\boldsymbol{z}$ has model

$$\boldsymbol{z} = \boldsymbol{F}\beta\,,\ \boldsymbol{F} = [\boldsymbol{f}(\phi_1)\,,\cdots,\boldsymbol{f}(\phi_r)]\,,\ \boldsymbol{f}(\phi) := \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{LP-1} \end{bmatrix}$$

- Construct a matrix $\boldsymbol{Z} = [\boldsymbol{z}^{(0)}, \cdots, \boldsymbol{z}^{(m-1)}]$ out of shifts of $\boldsymbol{z}$:

$$\boldsymbol{z}^{(i)} := \begin{bmatrix} z_{i+1} \\ z_{i+2} \\ \vdots \\ z_{LP-m+i} \end{bmatrix},\ \boldsymbol{f}(\phi)^{(i)} = \begin{bmatrix} \phi^i \\ \phi^{i+1} \\ \phi^{i+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \end{bmatrix} \phi^i =: \boldsymbol{f}'(\phi)\phi^i$$

Delay estimation using ESPRIT

Thus, $\boldsymbol{Z}$ has the model

$$\boldsymbol{Z} = \boldsymbol{F}'\boldsymbol{B}\,, \quad \boldsymbol{F}' = [\boldsymbol{f}'(\phi_1)\,, \cdots, \boldsymbol{f}'(\phi_r)]$$

$$\boldsymbol{B} = [\beta \quad \Phi\beta \quad \Phi^2\beta \quad \cdots \quad \Phi^{m-1}\beta]\,, \quad \Phi = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_r \end{bmatrix}$$

Model: $\boldsymbol{X} = \boldsymbol{F}''\boldsymbol{B}\,, \quad \boldsymbol{Y} = \boldsymbol{F}''\Phi\boldsymbol{B}$.

Now we can apply ESPRIT to estimate Φ and hence all $\{\tau_i\}$.

1. Compute the SVD: $\boldsymbol{Z} = \hat{\boldsymbol{U}}_z\hat{\Sigma}_z\hat{\boldsymbol{V}}_z^{\mathrm{H}}$
2. Split $\hat{\boldsymbol{U}}_z$ into $\hat{\boldsymbol{U}}_x$ and $\hat{\boldsymbol{U}}_y$ (shift over 1 row):

   $$\hat{\boldsymbol{U}}_z = \left[\frac{\hat{\boldsymbol{U}}_x}{*\,*\,*}\right] = \left[\frac{*\,*\,*}{\hat{\boldsymbol{U}}_y}\right]$$

   Model: $\hat{\boldsymbol{U}}_x = \boldsymbol{F}''\boldsymbol{T}\,, \quad \hat{\boldsymbol{U}}_y = \boldsymbol{F}''\Phi\boldsymbol{T}$ for some $r \times r$ matrix $\boldsymbol{T}$.
3. Compute the eigenvalue decomposition: $\hat{\boldsymbol{U}}_x^{\dagger}\hat{\boldsymbol{U}}_y = \boldsymbol{T}^{-1}\Phi\boldsymbol{T}$.

Frequency estimation using ESPRIT

Given that signal $x(t)$ is the sum of d harmonic components,

$$x(t) = \sum_{i=1}^{d} \beta_i e^{j\omega_i t},$$

estimate ω_i and β_i.

Collect N samples in a data matrix $\boldsymbol{Z}$ with m rows,

$$\boldsymbol{Z} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots \\ x_2 & x_3 & x_4 & \cdots \\ \iddots & \iddots & & \iddots \\ x_m & x_{m+1} & \cdots & x_N \end{bmatrix}, \quad x_k = x(kT)$$

$\boldsymbol{Z}$ has model

$$\boldsymbol{Z} = \boldsymbol{A}(\omega)\boldsymbol{S} := \begin{bmatrix} 1 & \cdots & 1 \\ \phi_1 & \cdots & \phi_d \\ \phi_1^2 & \cdots & \phi_d^2 \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \beta_1\phi_1 & \beta_1\phi_1^2 & \cdots \\ \vdots & \vdots & \\ \beta_d\phi_d & \beta_d\phi_d^2 & \cdots \end{bmatrix}, \; \phi_i = e^{j\omega_i T}$$

Summary

The ESPRIT algorithm is a computationally efficient way to exploit *shift invariance* in a model

It is often not optimal, but still offers quite good performance

It has two steps: subspace estimation and an eigenvalue problem. Many source separation algorithms consist of such steps

The algorithm can be extended to jointly estimate two angles, angle-frequency, angle-delay etc.