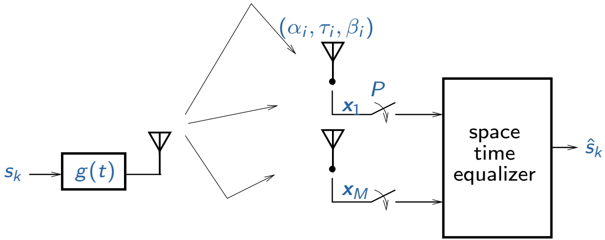
EE 4715 Array Processing

9. Joint diagonalization and Kronecker product structures

April 2022

Problem

We receive a signal over a multipath channel. Can we estimate *jointly* the angles, delays and fading parameters?



In this lecture, we look at **Kronecker product structures** to achieve this.

The vec operator

For a matrix, $\text{vec}(\cdot)$ denotes the stacking of the columns of a matrix into a vector:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \Rightarrow \quad \text{vec}(\boldsymbol{A}) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \hline a_{12} \\ a_{22} \\ a_{32} \\ \hline a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

The Kronecker product

For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the Kronecker product is defined a

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1N}\boldsymbol{B} \\ \vdots & & \vdots \\ a_{M1}\boldsymbol{B} & \cdots & a_{MN}\boldsymbol{B} \end{bmatrix},$$

Some properties:

$$\begin{aligned} (\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) &= \boldsymbol{AC} \otimes \boldsymbol{BD} \\ [\boldsymbol{a} \otimes \boldsymbol{b}][\boldsymbol{c} \otimes \boldsymbol{d}]^{\mathrm{H}} &= \boldsymbol{ac}^{\mathrm{H}} \otimes \boldsymbol{bd}^{\mathrm{H}} = \boldsymbol{a} \otimes \boldsymbol{bc}^{\mathrm{H}} \otimes \boldsymbol{d}^{\mathrm{H}} \\ \mathrm{tr}(\boldsymbol{A} \otimes \boldsymbol{B}) &= \mathrm{tr}(\boldsymbol{A})\mathrm{tr}(\boldsymbol{B}) \end{aligned}$$

$\mathrm{tr}(\cdot)$ is the trace operator: sum of the diagonal elements

The Kronecker product

A rank-one matrix has the form $\boldsymbol{a}\boldsymbol{b}^{\mathsf{T}}$.

- An important property:

$$\operatorname{vec}(\boldsymbol{a}\boldsymbol{b}^{\mathsf{T}}) = \boldsymbol{b} \otimes \boldsymbol{a} \quad \Leftrightarrow \quad \operatorname{vec}\begin{bmatrix} a_1 b_1 & a_1 b_2 \\ a_2 b_1 & a_2 b_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_2 b_1 \\ \hline a_1 b_2 \\ a_2 b_2 \end{bmatrix}$$

- For complex matrices:

$$\operatorname{vec}(\boldsymbol{a}\boldsymbol{b}^{\mathsf{H}}) = \boldsymbol{b}^* \otimes \boldsymbol{a}$$

The Kronecker product

More in general, for 3 matrices:

$$\text{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^{\mathsf{T}} \otimes \boldsymbol{A})\text{vec}(\boldsymbol{B})$$

- Prove by writing $\boldsymbol{ABC}^{\mathsf{T}} = \sum_{ij} b_{ij}\boldsymbol{a}_i\boldsymbol{c}_j^{\mathsf{T}}$ and using the previous result.
- Interpretation: $\boldsymbol{ABC}$ is linear in the entries of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$.

This implies that we can write $\text{vec}(\boldsymbol{ABC})$ in terms of a matrix times $\text{vec}(\boldsymbol{A})$, $\text{vec}(\boldsymbol{B})$ or $\text{vec}(\boldsymbol{C})$, respectively:

$$\begin{aligned}\text{vec}(\boldsymbol{ABC}) &= [(\boldsymbol{BC})^{\mathsf{T}} \otimes \boldsymbol{I}]\text{vec}(\boldsymbol{A}) \\ \text{vec}(\boldsymbol{ABC}) &= [\boldsymbol{I} \otimes \boldsymbol{AB}]\text{vec}(\boldsymbol{C})\end{aligned}$$

The Khatri-Rao product

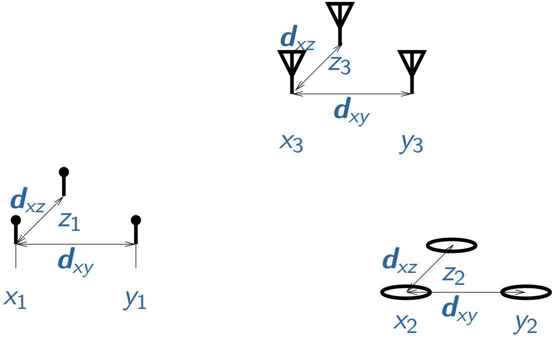
$\circ$ denotes the Khatri-Rao product, i.e., a column-wise Kronecker product:

$$\boldsymbol{A} \circ \boldsymbol{B} := [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \quad \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \quad \cdots]$$

- This forms a submatrix of $\boldsymbol{A} \otimes \boldsymbol{B}$.
- If $\boldsymbol{B} = \mathrm{diag}(\boldsymbol{b})$ is a diagonal matrix formed from $\boldsymbol{b}$, then

$$\mathrm{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^\mathsf{T} \circ \boldsymbol{A})\boldsymbol{b}$$

The extended ESPRIT algorithm



Consider M *triplets*: three identical but displaced subarrays.

The extended ESPRIT algorithm

Data model (d narrowband point sources):

$$\begin{cases} \boldsymbol{X} = \boldsymbol{A}_x\boldsymbol{S} = \boldsymbol{A}\boldsymbol{S} \\ \boldsymbol{Y} = \boldsymbol{A}_y\boldsymbol{S} = \boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{S} \\ \boldsymbol{Z} = \boldsymbol{A}_z\boldsymbol{S} = \boldsymbol{A}\boldsymbol{\Theta}\boldsymbol{S} \end{cases} \Leftrightarrow \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}\boldsymbol{\Phi} \\ \boldsymbol{A}\boldsymbol{\Theta} \end{bmatrix} \boldsymbol{S}\,.$$

$\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ are diagonal matrices with entries

$$\phi_k = e^{-j\frac{\omega_0}{c}\boldsymbol{d}_{xy}\cdot\zeta_k} \qquad \theta_k = e^{-j\frac{\omega_0}{c}\boldsymbol{d}_{xz}\cdot\zeta_k}$$

The DOA problem is to estimate $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ from $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$. This can be done from $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{X}, \boldsymbol{Z})$ separately, but how to find the pairs of angles (θ_i, ϕ_i)?

The extended ESPRIT algorithm

- Preprocessing: compute (truncated) SVD

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{H}}$$

- Partition $\boldsymbol{U}$ similar to $\boldsymbol{K}$:

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_x \\ \boldsymbol{U}_y \\ \boldsymbol{U}_z \end{bmatrix} \Sigma\boldsymbol{V}^{\mathrm{H}} \quad \text{but also} \quad \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}\Phi \\ \boldsymbol{A}\Theta \end{bmatrix} \boldsymbol{S}\,.$$

  The column spans must match: there is an invertible matrix $\boldsymbol{T}$ such that

$$\begin{bmatrix} \boldsymbol{U}_x \\ \boldsymbol{U}_y \\ \boldsymbol{U}_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}\Phi \\ \boldsymbol{A}\Theta \end{bmatrix} \boldsymbol{T}$$

Joint diagonalization

$\boldsymbol{A} = \boldsymbol{U}_x \boldsymbol{T}^{-1}$ implies

$$\begin{cases} \boldsymbol{U}_y &= \boldsymbol{U}_x \boldsymbol{T}^{-1} \boldsymbol{\Phi} \boldsymbol{T} \\ \boldsymbol{U}_z &= \boldsymbol{U}_x \boldsymbol{T}^{-1} \boldsymbol{\Theta} \boldsymbol{T} . \end{cases}$$

Define $\boldsymbol{M}_y = \boldsymbol{U}_x^\dagger \boldsymbol{U}_y$ and $\boldsymbol{M}_z = \boldsymbol{U}_x^\dagger \boldsymbol{U}_z$, then

$$\begin{cases} \boldsymbol{M}_y &= \boldsymbol{T}^{-1} \boldsymbol{\Phi} \boldsymbol{T} \\ \boldsymbol{M}_z &= \boldsymbol{T}^{-1} \boldsymbol{\Theta} \boldsymbol{T} . \end{cases}$$

The matrix $\boldsymbol{T}$ diagonalizes both $\boldsymbol{M}_y$ and $\boldsymbol{M}_z$ ($d \times d$ matrices derived from the data)

This is a **joint diagonalization problem**.

Computing the joint diagonalization

- Already one matrix specifies $\boldsymbol{T}$ (usual eigenvalue problem). The joint diagonalization problem gives redundancy $\Rightarrow$ more accurate results.

- We could solve one problem to find $(\boldsymbol{T}, \boldsymbol{\Phi})$ and apply $\boldsymbol{T}$ to $\boldsymbol{M}_y$ to find $\boldsymbol{\Theta}$.
  This fails if two values of $\boldsymbol{\Phi}$ are the same ($\boldsymbol{T}$ not uniquely defined) $\Rightarrow$ another reason for joint processing

- There are numerical algorithms to solve the joint (approximate) problem, e.g. using Jacobi rotations.

Connection to the Khatri-Rao product structure

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A\Phi} \\ \boldsymbol{A\Theta} \end{bmatrix} \boldsymbol{S}$$

Define a matrix $\boldsymbol{F}$ from the diagonals of $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ as

$$\boldsymbol{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \phi_1 & \phi_2 & \cdots & \phi_d \\ \theta_1 & \theta_2 & \cdots & \theta_d \end{bmatrix}$$

then we can write this compactly as

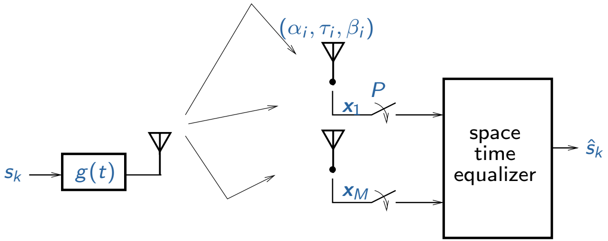
$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = (\boldsymbol{F} \circ \boldsymbol{A})\boldsymbol{S} \quad \text{and likewise} \quad \begin{bmatrix} \boldsymbol{U}_x \\ \boldsymbol{U}_y \\ \boldsymbol{U}_z \end{bmatrix} = (\boldsymbol{F} \circ \boldsymbol{A})\boldsymbol{T}$$

Connection to the Khatri-Rao product structure

This Khatri-Rao product structure is the *only* property that was needed to derive the joint diagonalization model

- Whenever we have this structure, we can transform it into joint diagonalization.
- We expanded on the rows of $\boldsymbol{F}$, but we can also expand on the rows of $\boldsymbol{A}$; even on $\boldsymbol{T}$.
- This is an example of a **canonical polyadic decomposition** (CPD) $\Rightarrow$ tensor decomposition framework

Joint angle-delay estimation



$$\boldsymbol{h}(t) = \sum_{i=1}^{r} \boldsymbol{a}(\alpha_i)\beta_i g(t - \tau_i)\,.$$

Sample $h(t)$ and stack N samples in $\boldsymbol{h}$:

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_0 \\ \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_{N-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}(0) \\ \boldsymbol{h}(T) \\ \vdots \\ \boldsymbol{h}((N-1)T) \end{bmatrix} = \sum_{i=1}^{r} [\boldsymbol{g}_{\tau_i} \otimes \boldsymbol{a}(\alpha_i)]\beta_i \;=\; [\boldsymbol{G} \circ \boldsymbol{A}]\boldsymbol{b}$$

Joint angle-delay estimation

$$\boldsymbol{h} = [\boldsymbol{G} \circ \boldsymbol{A}]\boldsymbol{b}$$

We approach this as we did for delay estimation.

- Apply a DFT to $\boldsymbol{h}$ and deconvolve $\boldsymbol{g}$ to obtain $\boldsymbol{z}$:

$$\boldsymbol{z} = [\boldsymbol{F} \circ \boldsymbol{A}]\boldsymbol{b}$$

with

$$\boldsymbol{F} = [\boldsymbol{f}(\phi_1), \cdots, \boldsymbol{f}(\phi_r)], \quad \boldsymbol{f}(\phi) = \begin{bmatrix} 1 \\ \phi \\ \phi^2 \\ \vdots \\ \phi^{N-1} \end{bmatrix}, \quad \phi := e^{-j\frac{2\pi}{N}\frac{\tau}{T}}$$

$\boldsymbol{F}$ contains the delay information.

Joint angle-delay estimation

We would like to apply joint diagonaliation to solve for (τ_i, α_i). But we only have a single vector $\boldsymbol{h}$.

We can expand $\boldsymbol{z}$ to a matrix using shift-invariance ("smoothing") of $\boldsymbol{A}$ (if we have a ULA) or $\boldsymbol{F}$ (after a DFT):

- Partition $\boldsymbol{z}$ and form block shifts: $\boldsymbol{Z} = [\boldsymbol{z}^{(0)}, \ \boldsymbol{z}^{(1)}, \ \cdots, \ \boldsymbol{z}^{(m-1)}]$

$$
\boldsymbol{z} = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_i \\ \vdots \\ z_{N-m+i} \\ \vdots \\ z_{N-1} \end{bmatrix} \qquad \left. \begin{matrix} z_i \\ \vdots \\ z_{N-m+i} \end{matrix} \right\} \boldsymbol{z}^{(i)}
$$

Joint angle-delay estimation

- Model: $\boldsymbol{Z} = [\boldsymbol{F}' \circ \boldsymbol{A}]\boldsymbol{B}$, with

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

  (The structure of $\boldsymbol{B}$ is not used.)

- First, compute an SVD (truncate to rank r):

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

- Model:

$$\boldsymbol{U} = (\boldsymbol{F}' \circ \boldsymbol{A})\boldsymbol{T}$$

- Various approaches are possible, e.g., expand on the rows of $\boldsymbol{F}'$.

Joint angle-delay estimation

Assuming a ULA, we can use shifts on both $\boldsymbol{A}$ and $\boldsymbol{F}$:

$$\boldsymbol{J}_{x\phi} := [\boldsymbol{I}_{N-m} \quad 0_1] \otimes \boldsymbol{I}_M \,, \qquad \boldsymbol{J}_{x\theta} := \boldsymbol{I}_{N-m+1} \otimes [\boldsymbol{I}_{M-1} \quad 0_1]$$
$$\boldsymbol{J}_{y\phi} := [0_1 \quad \boldsymbol{I}_{N-m}] \otimes \boldsymbol{I}_M \,, \qquad \boldsymbol{J}_{y\theta} := \boldsymbol{I}_{N-m+1} \otimes [0_1 \quad \boldsymbol{I}_{M-1}]$$

This will allow to estimate more sources than we have antennas.

- To estimate $\boldsymbol{\Phi}$, we take submatrices consisting of the first and last $M(N-m)$ rows of $\boldsymbol{U}$:

$$\boldsymbol{U}_{x\phi} = \boldsymbol{J}_{x\phi}\boldsymbol{U} \,, \qquad \boldsymbol{U}_{y\phi} = \boldsymbol{J}_{y\phi}\boldsymbol{U} \,,$$

- To estimate $\boldsymbol{\Theta}$ we stack, for all $N-m+1$ blocks, its first and respectively last $M-1$ rows:

$$\boldsymbol{U}_{x\theta} = \boldsymbol{J}_{x\theta}\boldsymbol{U} \,, \qquad \boldsymbol{U}_{y\theta} = \boldsymbol{J}_{y\theta}\boldsymbol{U} \,.$$

Joint angle-delay estimation

- Structure:

$$\begin{cases} \boldsymbol{U}_{x\phi} & = & \boldsymbol{A}'\boldsymbol{T} \\ \boldsymbol{U}_{y\phi} & = & \boldsymbol{A}'\boldsymbol{\Phi}\boldsymbol{T} \end{cases} \qquad \begin{cases} \boldsymbol{U}_{x\theta} & = & \boldsymbol{A}''\boldsymbol{T} \\ \boldsymbol{U}_{y\theta} & = & \boldsymbol{A}''\boldsymbol{\Theta}\boldsymbol{T} \,. \end{cases}$$

  Resulting joint diagonalization problem:

$$\begin{aligned} \boldsymbol{U}_{x\phi}^{\dagger}\boldsymbol{U}_{y\phi} &= \boldsymbol{T}^{-1}\boldsymbol{\Phi}\boldsymbol{T} \\ \boldsymbol{U}_{x\theta}^{\dagger}\boldsymbol{U}_{y\theta} &= \boldsymbol{T}^{-1}\boldsymbol{\Theta}\boldsymbol{T} \end{aligned}$$

- From $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$, we find the pairs (τ_i, α_i) of delays and angles.
- Possible extension to d sources each with a superposition of rays.

Exploiting fading diversity

If the fading parameters β_i are fast fading (with angles/delays constant over the observing interval), then

$$[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots] = [\boldsymbol{G} \circ \boldsymbol{A}]\boldsymbol{B}$$

Each $\boldsymbol{h}_k$ has the same model as before.

Due to fast fading, we do not need to use deconvolution by $\boldsymbol{g}$ followed by taking shifts to transform a single vector $\boldsymbol{h}$ into a matrix.

- Unvector each $\boldsymbol{h}_k$ gives

$$\boldsymbol{H}_k = \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{b}_k)\boldsymbol{G}^\top, \qquad k = 1, 2, \cdots$$

- Use joint diagonalization (non-symmetric) to solve for $\boldsymbol{A}$ and $\boldsymbol{G}$.

Joint angle and frequency estimation

In a wide frequency band, there are a number of narrowband sources, received by an antenna array. Find the angles and carrier frequencies.

- Assume narrowband signal can be sampled with $T = 1$ at Nyquist
- Sample the entire band at rate P. Without multipath:

$$\boldsymbol{x}(t) = \sum_{1}^{d} \boldsymbol{a}(\theta_i)\beta_i e^{j\frac{2\pi}{P}f_i t} s_i(t) \quad \Leftrightarrow \quad \boldsymbol{x}(t) = \boldsymbol{A}_\theta \boldsymbol{B} \boldsymbol{\Phi}^t \boldsymbol{s}(t)$$

Joint angle and frequency estimation

- If P is large, then subsample: take m samples at high rate, then wait:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}(0) & \boldsymbol{x}(1) & \cdots & \boldsymbol{x}(N-1) \\ \boldsymbol{x}(\frac{1}{P}) & \boldsymbol{x}(1+\frac{1}{P}) & \cdots & \boldsymbol{x}(N-1+\frac{1}{P}) \\ \vdots & \vdots & & \vdots \\ \boldsymbol{x}(\frac{m-1}{P}) & \boldsymbol{x}(1+\frac{m-1}{P}) & \cdots & \boldsymbol{x}(N-1+\frac{m-1}{P}) \end{bmatrix}$$

Model:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{s}(0) & \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{\Phi}^P \boldsymbol{s}(1) & \cdots \\ \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{s}(\frac{1}{P}) & \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{\Phi}^{P+1} \boldsymbol{s}(1+\frac{1}{P}) & \cdots \\ \vdots & \vdots & \\ \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{\Phi}^{m-1}\boldsymbol{s}(\frac{m-1}{P}) & \boldsymbol{A}_\theta \boldsymbol{B}\boldsymbol{\Phi}^{P+m-1} \boldsymbol{s}(1+\frac{m-1}{P}) & \cdots \end{bmatrix}$$

Joint angle and frequency estimation

- If $m \ll P$, then $\mathbf{s}(k) \approx \mathbf{s}(k + \frac{m-1}{P})$:

$$\begin{aligned} \mathbf{X} &\approx \begin{bmatrix} \mathbf{A}_\theta \\ \mathbf{A}_\theta \mathbf{\Phi} \\ \vdots \\ \mathbf{A}_\theta \mathbf{\Phi}^{m-1} \end{bmatrix} \mathbf{B} \left[\mathbf{s}_0 \quad \mathbf{\Phi}^P \mathbf{s}_1 \quad \cdots \quad \mathbf{\Phi}^{(N-1)P} \mathbf{s}_{N-1} \right] \\ &= (\mathbf{F}_\phi \circ \mathbf{A}_\theta) \, \mathbf{B} \, (\mathbf{F}_P \odot \mathbf{S}) \end{aligned}$$

We can now apply the same joint diagonalization algorithm as before.

Summary

If we have data with a Khatri-Rao structure

$$\boldsymbol{X} = (\boldsymbol{F} \circ \boldsymbol{A})\boldsymbol{S}$$

we can convert the problem to joint diagonalization, by expansion over the rows of $\boldsymbol{F}$ (or $\boldsymbol{A}$, or $\boldsymbol{S}$).

Joint diagonalization problems are of the form

$$\boldsymbol{M}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{A}^{\mathrm{H}}$$

(by congruence) but also $\boldsymbol{M}_k = \boldsymbol{T}^{-1}\Phi_k\boldsymbol{T}$ (by similarity) or $\boldsymbol{M}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{B}^{\mathrm{H}}$ (nonsymmetric).

This is an example of a canonical polyadic decomposition, a tensor decomposition.

Applications are joint estimation of azimuth-elevation, angle-delay, angle-frequency, multiple resolutions.