Microphone Array Processing

- **Estimation of the ATF**
- **Estimation of R_s**
- **GEVD and EVD**

Richard C. Hendriks

Delft University of Technology

Summary of Previous Lecture

Microphone Measurement Model

Single microphone model:

$$x[n] = \sum_{i=1}^{d} (h_i * s_i)[n] + n[n]$$

Assumptions: Sources are assumed to be

- Additive
- zero-mean and mutually uncorrelated, i.e., $E[s_i] = 0$, $E[n] = 0$, $E[s_i s_j] = 0\,\forall i, j$ and $E[s_i n] = 0\,\forall i$.
- short-time stationary.

Validity of these assumptions?

Short-Time Frequency Transform

Processing is often done in the so-called short-time frequency domain, i.e., FFT on short windowed time frames.

- Time frames should obey Short time WSS assumption.
- STFT makes convolutive model (aproximately) multiplicative AND helps to satisfy narrowband assumption.
- $x(k,l) = \sum_{i=1}^{d} a_i(k,l)s_i(k,l) + n(k,l)$
- For M microphones using stacked vector notation:

$$\mathbf{x}(k,l) = \sum_{i=1}^{d} \mathbf{a}_i(k,l)s_i(k,l) + \mathbf{n}(k,l)$$

- Notice: As all processing is often done independently per frequency band and time frame, time and frequency indices are usually neglected.

Problem formulation

$$\mathbf{x}(k,l) = \sum_{i=1}^{d} \mathbf{a}_i(k,l)s_i(k,l) + \mathbf{n}(k,l)$$

- Assuming a single target and considering remaining point sources as interferers, abusing notation we can write

$$\mathbf{x}(k,l) = \underbrace{\mathbf{a}_1(k,l)s_1(k,l)}_{target} + \underbrace{\sum_{i=2}^{d} \mathbf{a}_i(k,l)s_i(k,l) + \mathbf{n}'(k,l)}_{interferers+noise}$$

$$= \mathbf{a}(k,l)s(k,l) + \mathbf{n}(k,l)$$

- Goal: Estimate $s(k,l)$ given $\mathbf{x}(k,l)$: e.g. $\hat{s}(k,l) = E[s(k,l)|\mathbf{x}(k,l)]$

TU Delft

Summary of Previous Lecture

- Delay and sum beamformer

$$\mathbf{w}(k,l) = \frac{\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)\mathbf{a}(k,l)}$$

- MVDR beamformer

$$\mathbf{w}(k,l) = \frac{R_{\mathbf{x}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{x}}^{-1}(k,l)\mathbf{a}(k,l)} = \frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}$$

- Multi-Channel Wiener

$$\mathbf{w}(k,l) = \underbrace{\frac{\sigma_s^2(k,l)}{\sigma_s^2(k,l) + (\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l))^{-1}}}_{\text{Single-channel Wiener}} \underbrace{\frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}}_{MVDR}$$

TU Delft

Optimal Linear Multi-Channel Wiener

Signal model: $\mathbf{x}(k,l) = s(k,l)\mathbf{a}(k,l) + \mathbf{n}(k,l)$

Cost function: $J_{MSE}(\mathbf{w}(k,l)) = E[||s(k,l) - \mathbf{w}^H(k,l)\mathbf{x}(k,l)||_2^2]$

$$\begin{aligned}\frac{dJ_{MSE}(\mathbf{w}(k,l))}{d\mathbf{w}^H(k,l)} &= -E[s^H(k,l)\mathbf{x}(k,l)] + \mathbf{R}_{\mathbf{x}}(k,l)\mathbf{w}(k,l) \\ &= -\sigma_s^2(k,l)\mathbf{a}(k,l) + \mathbf{R}_{\mathbf{x}}(k,l)\mathbf{w}(k,l) = 0\end{aligned}$$

$$\mathbf{w}(k,l) = \mathbf{R}_{\mathbf{x}}^{-1}(l)\sigma_{S,k}^2\mathbf{a}(k,l)$$

TU Delft

Optimal Linear Multi-Channel Wiener

Using again the Matrix inversion lemma, it can be shown that

$$\mathbf{w}(k,l) = R_{\mathbf{x}}^{-1}(k,l)\sigma_s^2(k,l)\mathbf{a}(k,l)$$

can be written as

$$\mathbf{w}(k,l) = \underbrace{\frac{\sigma_s^2(k,l)}{\sigma_s^2(k,l) + (\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l))^{-1}}}_{\text{Single-channel Wiener}} \underbrace{\frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}}_{MVDR}$$

Optimal Linear Multi-Channel Wiener

matrix inversion lemma:

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

Matrix $\mathbf{R}_{\mathbf{x}}(k,l)$ can be written as $\mathbf{R}_{\mathbf{x}}(k,l) = \mathbf{R}_{\mathbf{n}}(k,l) + \mathbf{a}\mathbf{a}^H\sigma_s^2(k,l)$

$$\begin{aligned}
\mathbf{R}_{\mathbf{x}}^{-1}(k,l)\mathbf{a}(k,l)\sigma_s^2(k,l) &= \left(\mathbf{R}_{\mathbf{n}}(k,l) + \mathbf{a}\mathbf{a}^H\sigma_s^2(k,l)\right)^{-1}\mathbf{a}(k,l)\sigma_s^2(k,l) \\
&= \mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\sigma_s^2(k,l) \\
&- \mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\frac{\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{1+\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\sigma_s^2(k,l) \\
&= \mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\left(1 - \frac{\sigma_s^2(k,l)\mathbf{a}(k,l)^H\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{1+\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\right)\sigma_s^2(k,l)
\end{aligned}$$

Optimal Linear Multi-Channel Wiener

$$
\begin{aligned}
&= \mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\left(1-\frac{\sigma_s^2(k,l)\mathbf{a}(k,l)^H\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{1+\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\right)\sigma_s^2(k,l) \\
&= \mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\left(\frac{\sigma_s^2(k,l)}{1+\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\right) \\
&= \frac{\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\left(\frac{\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\sigma_s^2(k,l)}{1+\sigma_s^2(k,l)\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\right) \\
&= \frac{\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}\left(\frac{\sigma_s^2(k,l)}{\left(\mathbf{a}^H(k,l)\mathbf{R}_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)\right)^{-1}+\sigma_s^2(k,l)}\right)
\end{aligned}
$$

Optimal Linear Multi-Channel Wiener

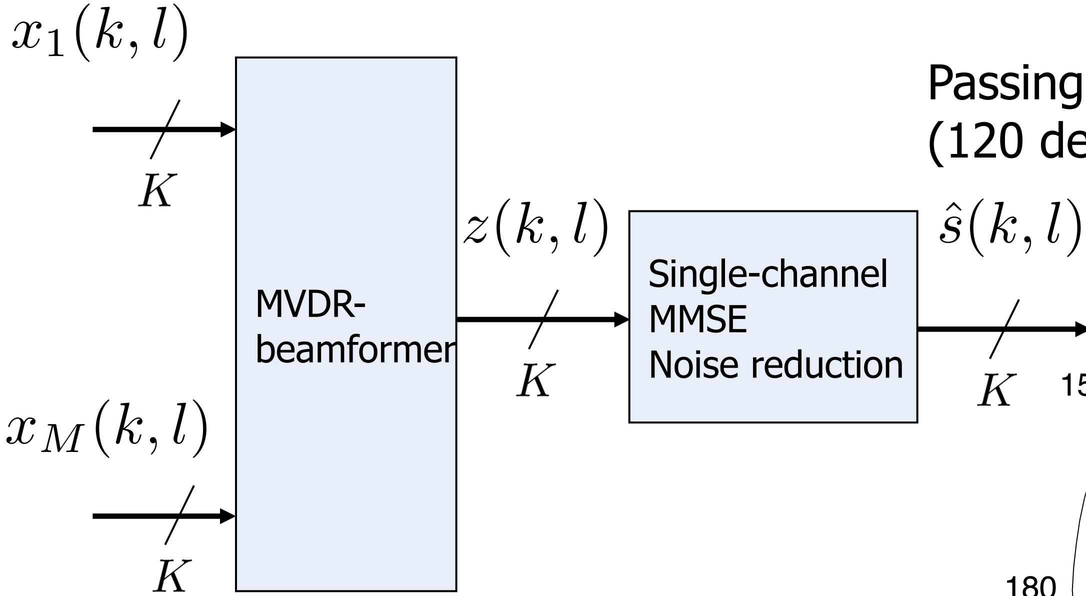
$$\mathbf{w}(k,l) = \underbrace{\frac{\sigma_s^2(k,l)}{\sigma_s^2(k,l) + (\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l))^{-1}}}_{\text{Single-channel Wiener}} \underbrace{\frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}}_{MVDR}$$

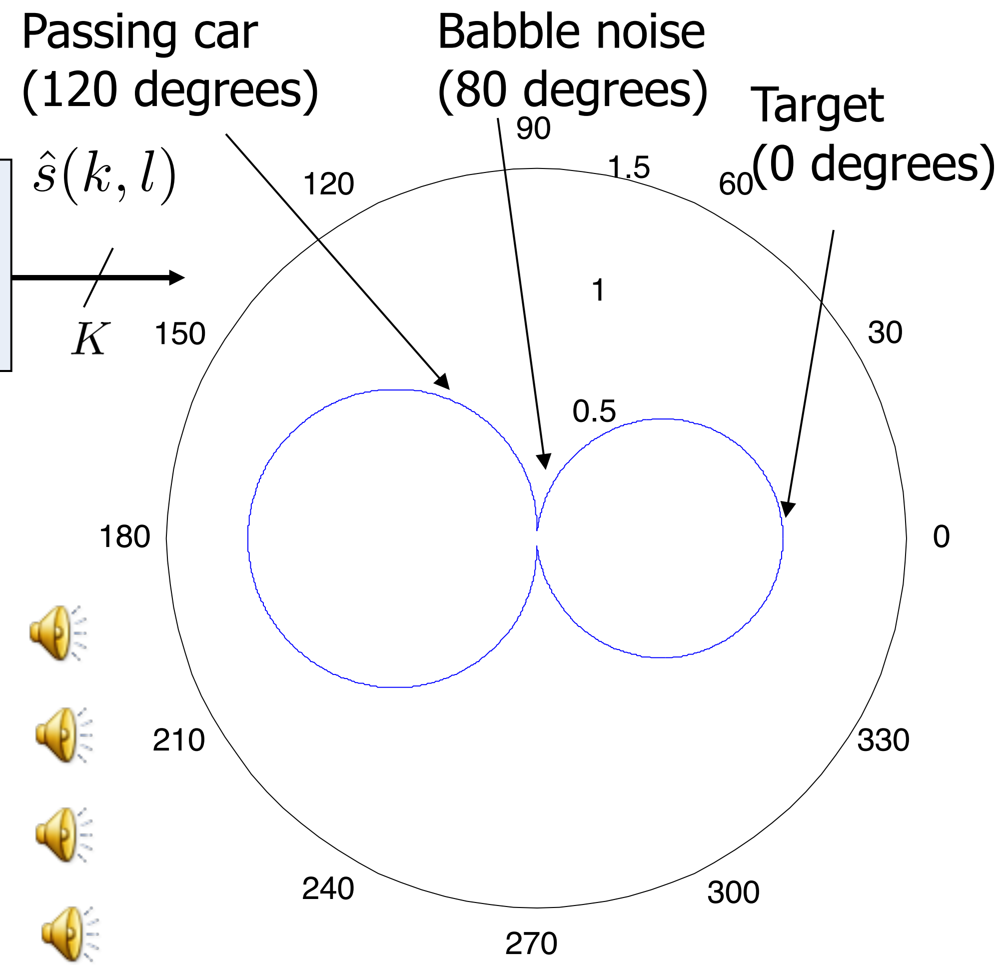
The multi-channel Wiener filter can thus be seen as a concatenation of two filters:

- An MVDR as spatial filter
- Single-Channel Wiener filter as post-processor where the noise variance is set to the remaining noise PSD after beamforming:

$$\mathbf{w}^H(k,l)R_{\mathbf{n}}(k,l)\mathbf{w}(k,l) = \mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)$$

Example: Multi-Channel Noise Reduction





- Clean input $s(k,l)$
- Noisy input at mic. 1 $x_1(k,l)$
- MVDR-beamformer output $z(k,l)$
- Output of total system $\hat{s}(k,l)$

Sufficient Statistics

- For $\mathbf{n}$ Gaussian distributed,

$$T\left(\mathbf{x}(k,l)\right) = \mathbf{w}_{\mathrm{MVDR}}^{H}(k,l)\mathbf{x}(k,l) = \frac{\mathbf{a}^{H}(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{x}(k,l)}{\mathbf{a}^{H}(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}$$

  is known to be a sufficient statistic for s.

- This means no information is lost on s by using $T\left(\mathbf{x}(k,l)\right)$ instead of $\mathbf{x}(k,l)$.

- This result holds in general for any prior distribution on $s(k,l)$ and any cost-function (e.g., quadratic (MSE), uniform (MAP), Absolute error (Median)) and any function of s (e.g., $|s|$, $|s|^2$, etc.)

Sufficient Statistics

- Let $f_S(s|\mathbf{x})$ denote the conditional pdf of random variable S. It then holds that for a sufficient statistics $f_S(s|\mathbf{x}) = f_S\left(s|T(\mathbf{x})\right)$.
- If $f_{\mathbf{x}}\left(\mathbf{x}|T(\mathbf{x};s)\right)$ is independent of s, $T(\mathbf{x})$ is a sufficient statistic for estimating s.
- Equivalent: $I\left(s;T(\mathbf{x})\right) = I\left(s;\mathbf{x}\right)$, i.e., we have equality in the data processing inequality and no information is lost.

Finding a sufficient statistic: if the pdf $f_{\mathbf{x}}\left(\mathbf{x};s\right)$ can be factorized as

$$f_{\mathbf{x}}\left(\mathbf{x};s\right) = g(T(\mathbf{x}),s)h(\mathbf{x}),$$

then $T(\mathbf{x})$ is a sufficient statistic for s.

LCMV - beamformer

Remember the MVDR: $J(\mathbf{w}(k,l)) = \mathbf{w}^H(k,l)\mathbf{R_x}(k,l)\mathbf{w}(k,l)$

$$\min_{\mathbf{w}(k,l)} J(\mathbf{w}(k,l))$$

$$s.t. \mathbf{w}(k,l)^H \mathbf{a}(k,l) = 1.$$

- The MVDR imposes one constraint.
- This can be generalised to having d constraints.

TU Delft

LCMV - beamformer

Cost function: $J(\mathbf{w}(k,l)) = \mathbf{w}^H(k,l)\mathbf{R}_{\mathbf{x}}(k,l)\mathbf{w}(k,l)$

$$\min_{\mathbf{w}(k,l)} J(\mathbf{w}(k,l))$$

$$s.t. \mathbf{w}^H(k,l)\mathbf{\Lambda}(k,l) = \mathbf{f}^H(k,l).$$

with $\mathbf{\Lambda} \in \mathbb{C}^{M \times d}$

When $d < M$, there is a closed form solution:

$$\mathbf{w}(k,l) = \mathbf{R}_{\mathbf{x}}^{-1}(k,l)\mathbf{\Lambda}(k,l)\left(\mathbf{\Lambda}^H(k,l)\mathbf{R}_{\mathbf{x}}^{-1}(k,l)\mathbf{\Lambda}(k,l)\right)^{-1}\mathbf{f}(k,l).$$

LCMV - beamformer

$$\mathbf{w}_k = \mathbf{R}_{\mathbf{x}}^{-1}(k,l)\mathbf{\Lambda}(k,l)\left(\mathbf{\Lambda}^{H}(k,l)\mathbf{R}_{\mathbf{x}}^{-1}(k,l)\mathbf{\Lambda}(k,l)\right)^{-1}\mathbf{f}(k,l).$$

How to use the multiple constraints?

- To steer zeros in the direction of certain noise sources.
- To maintain the signal from certain directions.
- To maintain the spatial cues of for hearing aids.

Notice that the more constraints are used, less degrees of freedom are left to control the noise reduction.

Overview of Discussed filters

- Delay and sum beamformer

$$\mathbf{w}(k,l) = \frac{\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)\mathbf{a}(k,l)}$$

- MVDR beamformer

$$\mathbf{w}(k,l) = \frac{R_{\mathbf{x}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{x}}^{-1}(k,l)\mathbf{a}(k,l)} = \frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}$$

- Multi-Channel Wiener

$$\mathbf{w}(k,l) = \underbrace{\frac{\sigma_s^2(k,l)}{\sigma_s^2(k,l) + (\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l))^{-1}}}_{\text{Single-channel Wiener}} \underbrace{\frac{R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}{\mathbf{a}^H(k,l)R_{\mathbf{n}}^{-1}(k,l)\mathbf{a}(k,l)}}_{MVDR}$$

Overview of Discussed filters

- LCMV beamformer
  $$\mathbf{w}(k,l) = \mathbf{R}_{\mathbf{x}}^{-1}(k,l)\boldsymbol{\Lambda}(k,l)\left(\boldsymbol{\Lambda}^{H}(k,l)\mathbf{R}_{\mathbf{x}}^{-1}(k,l)\boldsymbol{\Lambda}(k,l)\right)^{-1}\mathbf{f}(k,l).$$

Today

All beamformers discussed so far depend on the acoustic transfer function (ATF) $\mathbf{a}$.

Today:

- How to estimate $\mathbf{a}$
- How to estimate $R_{\mathbf{S}}$ (rank-r extention)
- Beamformers in terms of $R_{\mathbf{S}}$

TU Delft

Acoustic transfer function

The acoustic events in a room, under some assumptions, can be mathematically idealised as to be linear and time-invariant (LTI) so that the sound as it would be measured at the receiver can be calculated directly by convolution of the RIR and the source signal.

Assume that the target signal, say s, is a point source. Let h_m denote the RIR from the source s to microphone m. In that case, the signal x_m (the noise-free source signal received at the mth microphone) is given by

$$x_m[n] = (h_m * s)[n]$$

TUDelft

Acoustic transfer function

Using the STFT, we can write equivalently

$$x_m(k,l) = a_m(k,l)s(k,l)$$

- The function $a_m(k,l)$ is called the acoustic transfer function (ATF) from the source to the mth microphone.
- Notice that it is thus the temporal frequency domain representation of the room impulse response.
- Given a fixed frequency band k, we can collect the M microphone DFT coefficients in a vector $\mathbf{x}(k,l) = [x_1(k,l), ..., x_M(k,l)]^T$ such that $\mathbf{x}(k,l) = \mathbf{a}(k,l)s(k,l)$.

Relative Acoustic transfer function

- In many applications we are interested in the relative acoustic transfer function, which is then normalize with respect to a reference location, e.g., with respect to one of the microphones,

$$\mathbf{a}'(k,l) = [1, a_2(k,l)/a_1(k,l)..., a_M(k,l)/a_1(k,l)]^T .$$

TUDelft

Cross Power Spectral Density Matrices

Assuming that all sources ($s_i[n]$ and $n[n]$) are realizations of random processes, we can define the cross power spectral density matrix per frequency band k and time frame l:

$$E[\mathbf{x}(k,l)\mathbf{x}^H(k,l)] = \underbrace{E[\mathbf{s}(k,l)\mathbf{s}^H(k,l)]}_{\text{target source}} + \underbrace{E[\mathbf{n}(k,l)\mathbf{n}^H(k,l)]}_{\text{interferers/noise}}$$

often written as

$$\mathbf{R}_{\mathbf{x}}(k,l) = \mathbf{R}_{\mathbf{s}}(k,l) + \mathbf{R}_{\mathbf{n}}(k,l)$$

Cross Power Spectral Density Matrices

Assume that the target signal $s[n]$ is a point source, so that

$$\mathbf{s}(k,l) = \mathbf{a}(k,l)s_1(k,l),$$

where $\mathbf{a} \in \mathbb{C}^M$ is the (relative) acoustic transfer functions from the source to the microphones. With this, $\mathbf{R_s}$ can be expressed as

$$\mathbf{R_s} = \mathrm{E}[\mathbf{s}(k,l)\mathbf{s}(k,l)^H] = \sigma_{s_1}^2 \mathbf{a}(k,l)\mathbf{a}^H(k,l),$$

where $\sigma_{s_1}^2 = \mathrm{E}[|s_1(k,l)|]^2$, the variance of the clean signal as received at the reference microphone 1.

Note that in this case $\mathrm{rank}(\mathbf{R_s}) = 1$.

TU Delft

Cross Power Spectral Density Matrices

The beamformers derived so far assume that the (relative) acoustic transfer function $\mathbf{a}(k,l)$ is known a-priori

- In practice, $\mathbf{a}(k,l)$ is unknown and needs to be estimated
- Estimation errors in $\mathbf{a}(k,l)$ generally lead to severe performance degradation of the beamformer
- When there are multiple sources, the beamformers will be a function of a general correlation matrix $\mathbf{R_s}$

In the following, we will focus on 1) estimating $\mathbf{a}$, 2) estimating $\mathbf{R_s}$ and 3) give expressions for beamformers in terms of a general $\mathbf{R_s}$, not necessarily of rank 1

TUDelft

The Eigenvalue Problem

The eigenvalue problem for a square matrix $\mathbf{A}$ is

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \Leftrightarrow (\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0}.$$

Any λ that makes $\mathbf{A} - \lambda\mathbf{I}$ singular is called an eigenvalue, the corresponding $\mathbf{x}$ is the eigenvector. It has an arbitrary norm, usually set equal to 1.

We can collect the eigenvectors in a matrix:

$$\mathbf{A}\left[\mathbf{x}_1\,\mathbf{x}_2\,\ldots\right] = \left[\mathbf{x}_1\,\mathbf{x}_2\,\ldots\right] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} \Leftrightarrow \mathbf{AT} = \mathbf{T\Lambda}.$$

TU Delft

The Eigenvalue Problem

Assume $\mathbf{A}$ is *Hermitian*.

- Every eigenvalue is real:

$$\lambda\|\mathbf{x}\|^2 = \lambda\mathbf{x}^H\mathbf{x} = \mathbf{x}^H\mathbf{A}\mathbf{x} = \mathbf{x}^H\mathbf{A}^H\mathbf{x} = \overline{\lambda}\mathbf{x}^H\mathbf{x} = \overline{\lambda}\|\mathbf{x}\|^2$$

- The eigenvectors are mutually orthogonal $(\mathbf{T}^{-\mathbf{1}} = \mathbf{T}^H)$:

  Let $\mathbf{x}$ and $\mathbf{y}$ be eigenvectors of $\mathbf{A}$ corresponding to distinct eigenvalues λ and μ, respectively. Then

$$\lambda\mathbf{y}^H\mathbf{x} = \mathbf{y}^H\mathbf{A}\mathbf{x} = \mathbf{y}^H\mathbf{A}^H\mathbf{x} = \overline{\mu}\mathbf{y}^H\mathbf{x} = \mu\mathbf{y}^H\mathbf{x}$$

  Hence $(\lambda - \mu)\mathbf{y}^H\mathbf{x} = 0$ and thus $\mathbf{x} \perp \mathbf{y}$.

The Eigenvalue Problem

If, in addition, $\mathbf{A}$ is *positive semi-definite* ($\mathbf{x}^H \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{C}^n$), then

- Every eigenvalue is nonnegative:

$$\lambda \|\mathbf{x}\|^2 = \lambda \mathbf{x}^H \mathbf{x} = \mathbf{x}^H \mathbf{A} \mathbf{x} \geq 0 \;\Rightarrow\; \lambda \geq 0$$

The Eigenvalue Problem

If it exists, the eigenvalue decomposition of a square matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1},$$

with $\mathbf{T}$ invertible and $\mathbf{\Lambda}$ diagonal.

Notice that if $\mathbf{A}$ is Hermitian, that the eigenvalue are real and non-negative, and the eigenvectors are orthogonal. If the eigenvectors have norm one, $\mathbf{T}$ is unitary ($\mathbf{T}^{-1} = \mathbf{T}^H$) and thus

$$\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^H.$$

Calculating the ATF - No Noise

Assume that $\mathbf{R_s}$ is perfectly known, i.e.,

$$\mathbf{R_s} = \mathrm{E}[\mathbf{s}(k,l)\mathbf{s}(k,l)^H] = \sigma_{s_1}^2(k,l)\mathbf{a}(k,l)\mathbf{a}^H(k,l).$$

Since $\mathbf{R_s}$ is Hermitian and positive semi-definite, there exists a unitary matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_M)$, $\mathbf{u}_i \in \mathbb{C}^M$, such that

$$\mathbf{R_s} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H,$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$, $\lambda_i \geq 0$ for all i.

In this case, $\boldsymbol{\Lambda} = \mathrm{diag}\left(\sigma_{s_1}^2(k,l)\|\mathbf{a}(k,l)\|^2, 0, \ldots, 0\right)$ and the (scaled) ATF is given by $\mathbf{u}_1 = \mathbf{a}(k,l)/\|\mathbf{a}(k,l)\|$

Calculating the ATF – Spatially White Noise

Now suppose $\mathbf{R_x}(k,l)$ is known and $\mathbf{R_n}(k,l) = \sigma_n^2(k,i)\mathbf{I}$, i.e.,

$$\mathbf{R_x}(k,l) = \mathbf{R_s}(k,l) + \mathbf{R_n}(k,l) = \sigma_{s_1}^2(k,l)\mathbf{a}(k,l)\mathbf{a}^H(k,l) + \sigma_n^2(k,i)\mathbf{I}$$

As an identity matrix is diagonalizable by any unitary matrix, we have

$$\begin{aligned}\mathbf{R_x} &= \mathbf{R_s} + \sigma_n^2\mathbf{I}\\ &= \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H + \sigma_n^2\mathbf{I}\\ &= \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H + \sigma_n^2\mathbf{U}\mathbf{U}^H\\ &= \mathbf{U}(\boldsymbol{\Lambda} + \sigma_n^2\mathbf{I})\mathbf{U}^H\end{aligned}$$

which is the eigenvalue decomposition of $\mathbf{R_x}$.

TUDelft

Calculating the ATF

Conclusions:

- For spatially white noise, $\mathbf{R}_\mathbf{s}(k,l)$ (which is not available) and $\mathbf{R}_\mathbf{x}(k,l)$ (which we can estimate from the observed data) share the same eigenvectors
- Adding (spatially uncorrelated) noise to the desired speech data *only* affects the eigenvalues of $\mathbf{R}_\mathbf{s}(k,l)$
- Given $\mathbf{R}_\mathbf{x}(k,l)$, assuming that the noise is spatially white, the ATF can still be calculated by taking the principle eigenvector.

Remark: Notice that from now on, indices (k,l) will be neglected because of notational convenience.

Estimating R_s - Spatially White Noise

Let us assume that $\mathrm{rank}(\mathbf{R_s}) = r < M$. We can partition $\mathbf{R_x}$ as

$$\mathbf{R_x} = (\mathbf{U}_1 \ \mathbf{U}_2) \begin{pmatrix} \mathbf{\Lambda}_1 + \sigma_n^2 \mathbf{I}_r & O \\ O & \sigma_n^2 \mathbf{I}_{M-r} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{pmatrix},$$

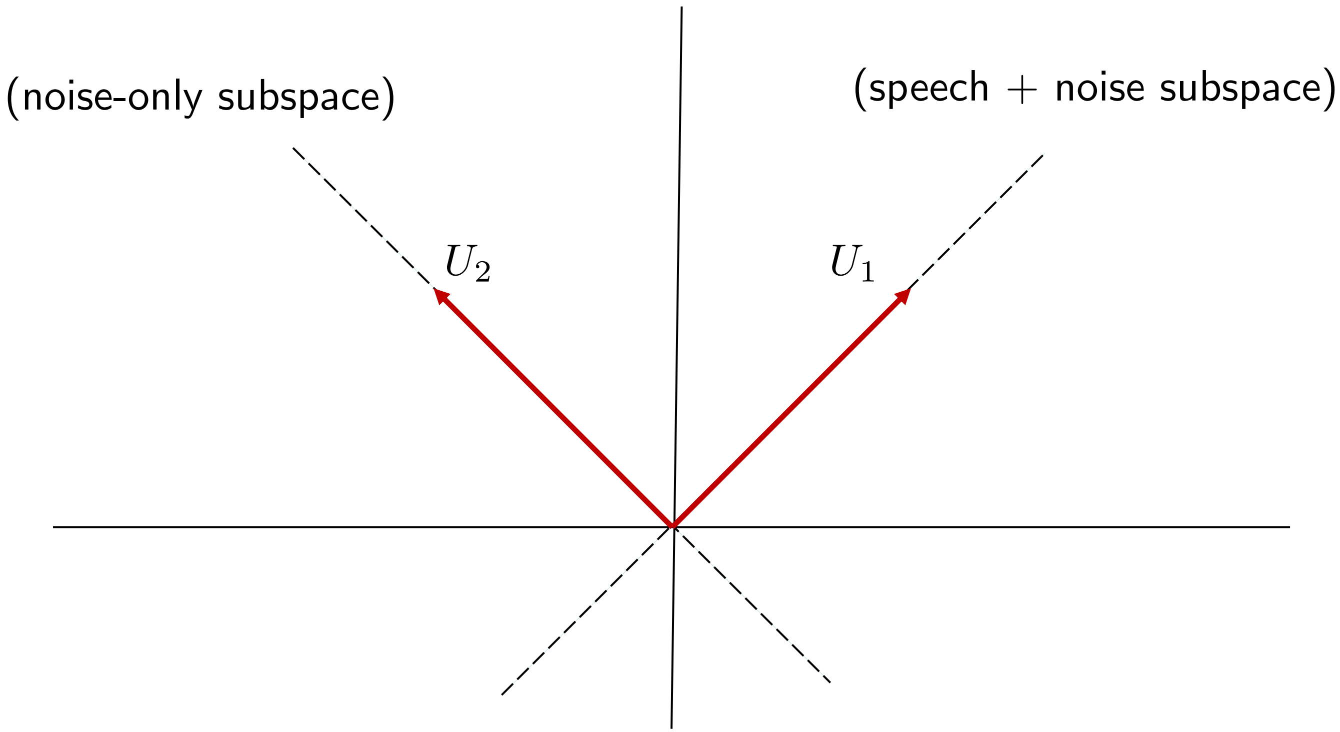
where $\mathbf{U}_1 \in \mathbb{C}^{M \times r}, \mathbf{U}_2 \in \mathbb{C}^{M \times (M-r)}$ and $\Lambda_1 \in \mathbb{C}^{r \times r}$.

Since $\mathbf{R_x} = \mathbf{U}_1(\mathbf{\Lambda}_1 + \sigma_n^2 \mathbf{I}_r)\mathbf{U}_1^H + \sigma_n^2 \mathbf{U}_2 \mathbf{U}_2^H$, we conclude that the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$ span the *speech (+ noise) subspace*, whereas $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_M$ span the *noise only* subspace.

Since $\mathbf{U}$ is unitary, we have $\mathbf{U}_1^H \mathbf{U}_2 = \mathbf{O}$ (orthogonal subspaces).

TUDelft

Geometric interpretation



TU Delft

Estimating R_s - Spatially White Noise

Despite the fact that we do not know what the signal subspace is a priori ($\mathbf{R_s}$ is unknown), we can compute (estimate) $\mathbf{R_s}$ from the EVD of $\mathbf{R_x}$.

Least-squares estimate is obtained by approximating $\mathbf{R_s}$ by

$$\hat{\mathbf{R}}_\mathbf{s} = \arg \min_{\text{rank}(\mathbf{R_s})=r} \|\mathbf{R_x} - \mathbf{R_s}\|_F^2$$

The solution is a classical result and follows by truncating the $M - r$ smallest eigenvalues. That is,

$$\hat{\mathbf{R}}_\mathbf{s} = \mathbf{U}_1(\mathbf{\Lambda}_1 + \sigma_n^2 \mathbf{I}_r)\mathbf{U}_1^H$$

TUDelft

Estimating R_s - Spatially White Noise

Since the last $M - r$ eigenvalues are given by σ_n^2, we can even do better by subtracting σ_n^2 from the largest r eigenvalues (results in a minimum variance estimator). That is,

$$\hat{\mathbf{R}}_\mathbf{s} = \mathbf{U}_1 \Lambda_1 \mathbf{U}_1^H,$$

which is *identical* to $\mathbf{R}_\mathbf{s}$.

Note that in practice we have to estimate $\mathbf{U}$ and $\boldsymbol{\Lambda}$ (and thus $\mathbf{U}_1$ and $\boldsymbol{\Lambda}_1$) from the noisy observations and for that reason the resulting estimator is *not* identical to $\mathbf{R}_\mathbf{s}$ although the above equation suggests so.

TUDelft

Estimating R_s – Pre-Whitening

If the noise process $\mathbf{n}$ is *not* white ($\mathbf{R_n} \neq \alpha \mathbf{I}_M$ for some $\alpha > 0$), we can pre-whiten the data, assuming that $\mathbf{R_n} \succ 0$ (positive definite)

Since $\mathbf{R_n}$ is Hermitian and positive definite, we have

$$\mathbf{R_n} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H = \mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^H = (\mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^H)(\mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^H) = \mathbf{R}_{\mathbf{n}}^{\frac{1}{2}}\mathbf{R}_{\mathbf{n}}^{\frac{1}{2}}$$

where $\mathbf{R}_{\mathbf{n}}^{\frac{1}{2}}$ is the (unique) Hermitian *square root* of $\mathbf{R_n}$.

Consider the transformed process $\tilde{\mathbf{n}} = \mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}}\mathbf{n}$. The process $\tilde{\mathbf{n}}$ is spatially white:

$$\mathbf{R}_{\tilde{\mathbf{n}}} = \mathrm{E}(\tilde{\mathbf{n}}\tilde{\mathbf{n}}^H) = \mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}}\mathrm{E}(\mathbf{n}\mathbf{n}^H)\mathbf{R}_{\tilde{\mathbf{n}}}^{-\frac{1}{2}} = \mathbf{I}_M$$

TU Delft

Estimating R_s – Pre-Whitening

Next consider the transformed process $\tilde{\mathbf{x}} = \mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}}\mathbf{x}$. Since this transformation transforms the original noise process into a spatially uncorrelated one, we have

$$\mathbf{R}_{\tilde{\mathbf{x}}} = \mathrm{E}(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H) = \mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}} E(\mathbf{x}\mathbf{x}^H)\mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}} = \mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}}\mathbf{R}_{\mathbf{s}}\mathbf{R}_{\mathbf{n}}^{-\frac{1}{2}} + \mathbf{I}_M.$$

Hence, we can apply the same techniques as discussed previously to the transformed process $\tilde{\mathbf{x}}$ and de-whiten the result thus obtained.

Estimating R_s – Pre-Whitening

Estimation of $\mathbf{R_s}$:

1. Compute $\mathbf{R_n^{\frac{1}{2}}}$ and pre-whiten the data: $\tilde{\mathbf{x}} = \mathbf{R_n}^{-\frac{1}{2}}\mathbf{x}$
2. Compute the EVD $\mathbf{R}_{\tilde{\mathbf{x}}} = \tilde{\mathbf{U}}\left(\tilde{\mathbf{\Lambda}} + \mathbf{I}_M\right)\tilde{\mathbf{U}}^H$, truncate the $M - r$ smallest eigenvalues and reduce the remaining ones by one.
3. Estimate $\hat{\mathbf{R}}_{\tilde{\mathbf{s}}} = \tilde{\mathbf{U}}_1\tilde{\mathbf{\Lambda}}_1\tilde{\mathbf{U}}_1^H$
4. De-whiten the result thus obtained so that
$$\hat{\mathbf{R}}_{\mathbf{s}} = \mathbf{R_n^{\frac{1}{2}}}\tilde{\mathbf{U}}_1\tilde{\mathbf{\Lambda}}_1\tilde{\mathbf{U}}_1^H\mathbf{R_n^{\frac{1}{2}}}$$

If $rank\left(\mathbf{R_s}(k,l)\right) = 1$, the ATF for spatially non-white noise can thus be obtained by selecting the principle eigenvector from $\hat{\mathbf{R}}_{\mathbf{s}}$ or from $\mathbf{R_n}^{1/2}\tilde{\mathbf{U}}_1$

TU Delft

Generalised eigenvalue decomposition

Remarks:

- The explicit use of $\mathbf{R}_{\mathbf{n}}^{\frac{1}{2}}$ may result in a loss of accuracy in the data
- Can be avoided by working directly with $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{R}_{\mathbf{n}}$
- In addition, when $\mathbf{R}_{\mathbf{n}}$ and/or $\mathbf{R}_{\mathbf{x}}$ are updated in a recursive way, it is generally very complicated to update $\mathbf{R}_{\tilde{\mathbf{x}}}$, while it is much simpler to calculate updates of $\mathbf{R}_{\mathbf{n}}^{-1}$ (using the matrix inversion lemma)

Another (in theory equivalent) method is given by the *generalised eigenvalue decomposition*

TU Delft