

# Advanced Spatial Audio and Speech Processing

## LVA/ICA Summer School

Emmanuel Vincent and Emanuël Habet

August 25, 2015



# Outline

Introduction

Fundamental Acoustics

Signal Models

Fundamental Array Processing

Data-Independent Beamforming

Data-Dependent Beamforming

Data-Dependent Source Separation

Summary and Perspectives

# Outline

## Introduction

- Considered Problem
- Applications
- General Approach
- Focus and Overview

## Fundamental Acoustics

## Signal Models

## Fundamental Array Processing

## Data-Independent Beamforming

## Data-Dependent Beamforming

## Data-Dependent Source Separation

## Summary and Perspectives

# Introduction

## Considered Problem

Commercial applications of speech and audio processing are already available for, e.g., speech recorded by a close-talk microphone in a quiet environment.

But audio scenes are often more complicated due to

- ▶ reverberation,
- ▶ noise,
- ▶ multiple sound sources.



# Introduction

## Considered Problem

A general problem is to analyze such sound scenes in order to

1. describe the environment,
2. localize the sources,
3. describe them,
4. enhance or separate them.

Humans are able to perform the three first tasks above in many situations.

In this lecture, we focus on the problem of **speech enhancement or source separation** for **multichannel signals**.

# Introduction

## Applications

Three categories of applications:

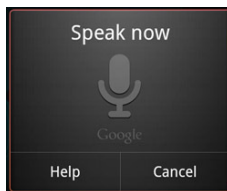
- ▶ separation per se,
- ▶ remixing,
- ▶ information retrieval from multisource audio.

Some practical examples follow.

# Introduction

## Applications

Spoken communication and personal assistants: simple noise reduction techniques already available in today's phones/hearing aids.



# Introduction

## Applications

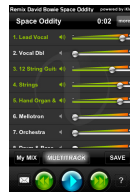
3D audio: upmixing of mono/stereo formats or new 3D formats (SAOC).



# Introduction

## Applications

Creative & interactive audio: similar to 3D audio but finer-grained separation and control for professionals/general public.





# Introduction

## Applications

Monitoring and surveillance: similar to smart homes but healthcare/security market.



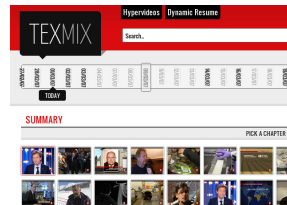
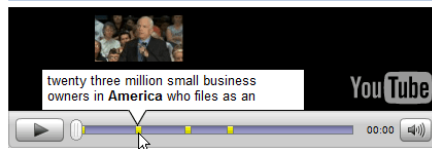
# Introduction

## Applications

Audiovisual content management: index speech and music documents with robust, detailed information.

What did the candidates say?

All Politicians | [McCain](#) | [Obama](#)






# Introduction


## Applications

### Sound examples


Music 



Vocals 


Speech in bus 



Speech 

TV series 



Speech 

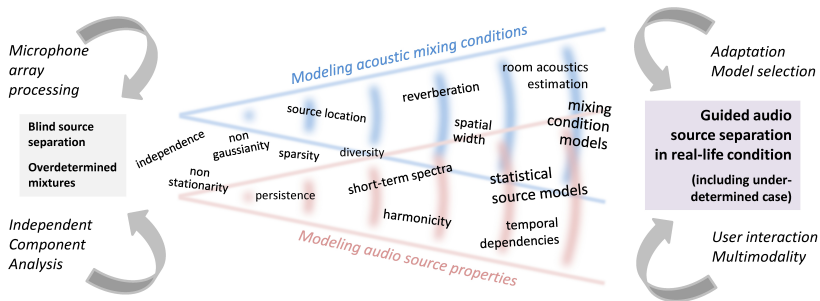
# Introduction

## General Approach

Solving the speech enhancement and source separation problem requires building models of:

- ▶ audio source properties aka **spectral models**,
- ▶ acoustic mixing conditions aka **spatial models**.

Increasingly complex models have been proposed over time.



In the following, we focus on:

- ▶ **microphone array recordings of speech**

→ similar principles apply to artificial multichannel mixes and to music

- ▶ **spatial modeling and estimation**

→ for state-of-the-art spectral modeling and estimation techniques, see

- ▶ T. Cemgil's course on nonnegative matrix and tensor factorizations (Aug 24)
- ▶ D. Wang's keynote and special session on deep neural networks (Aug 26)

# Introduction

## Focus and Overview

Research in the field is typically categorized either as microphone array processing or source separation.

Rather than opposing them, this lecture seeks to provide

- ▶ an overview of their common foundations,
- ▶ more details about the most usual algorithms,
- ▶ a summary of their common perspectives.

# Outline

Introduction

**Fundamental Acoustics**

Physics

Deterministic Perspective

Statistical Perspective

Signal Models

Fundamental Array Processing

Data-Independent Beamforming

Data-Dependent Beamforming

Data-Dependent Source Separation

Summary and Perspectives

# Fundamental Acoustics

## Physics

At usual loudness levels, the wave equation that governs the propagation of sound in air is linear:

1. the sound field at any time is the sum of the sound fields resulting from each source at that time;
2. the sound field emitted by a given source propagates over space and time according to a linear operation.

Unless clipping occurs, microphones also operate linearly.

**The overall mixing process is therefore linear.**

# Fundamental Acoustics

## Physics

In the free field, the recorded waveform differs from the emitted waveform by

- ▶ a **delay** of  $\ell/c$ ,
- ▶ an **attenuation** factor of  $1/\sqrt{4\pi\ell}$ .

$$x(t) = \frac{1}{\sqrt{4\pi\ell}} s\left(t - \frac{\ell}{c}\right)$$

$x(t)$ : recorded

$s(t)$ : emitted

$\ell$ : source-to-microphone distance

$c$ : speed of sound = 343 m/s

**Figure :** The spherical wave. Points on each sphere correspond to the same pressure.

# Fundamental Acoustics

## Physics

In the presence of obstacles, the wave is subject to different phenomena depending on its wavelength  $\lambda$  (from 17 mm at 20 kHz to 17 m at 20 Hz):

- ▶ reflection on surfaces of larger size (walls),
- ▶ diffraction on obstacles of similar size (furniture, ear+head+torso).

Smaller objects have little effect.

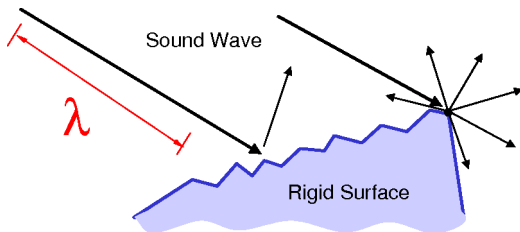


Figure : Reflection (left) and diffraction (right).



# Fundamental Acoustics

## Physics

The amount of reflected energy is as high as 85% for a carpeted floor and 99% for a tiled floor.

This induces **thousands to millions of propagation paths** between each source and each microphone.

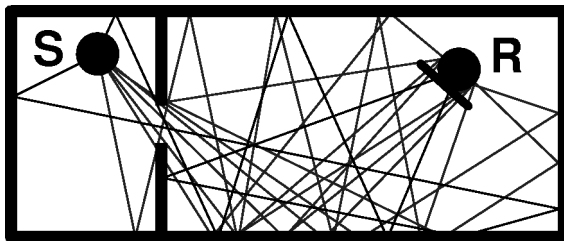


Figure : A few of the propagation paths.

# Fundamental Acoustics

## Deterministic Perspective

The summation of the propagation paths at each microphone results in an **acoustic impulse response**  $a(t)$ .

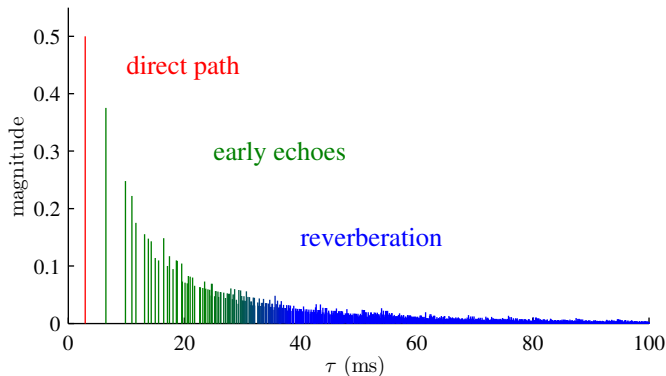
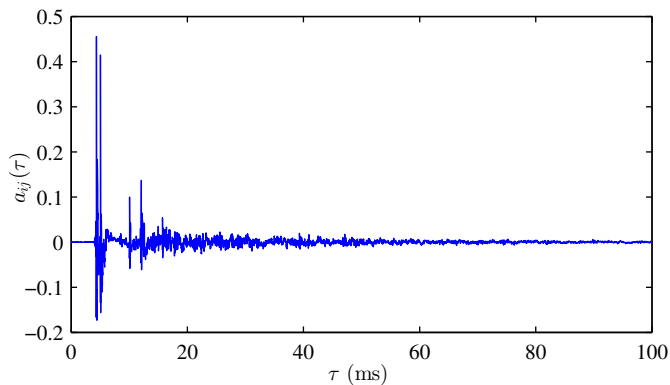


Figure : Illustration of the shape of an impulse response with  $RT = 250$  ms.

# Fundamental Acoustics

## Deterministic Perspective



**Figure :** Real impulse response recorded in a meeting room with  $RT = 230$  ms and a source-to-microphone distance of 1.45 m.

# Fundamental Acoustics

## Deterministic Perspective

The overall shape of acoustic impulse responses is often described by

- ▶ the reverberation time (RT)
  - ▶ time it takes for the reverberant tail to decay by 60 decibels (dB)
  - ▶ depends solely on the room
- ▶ the direct-to-reverberant ratio (DRR)
  - ▶ ratio of the power of the direct path to the rest of the impulse response
  - ▶ depends on the room and the source-to-microphone distance

The RT varies from 50 ms in a car to 1 s or more in an auditorium.

# Fundamental Acoustics

## Statistical Perspective

Since it results from the superposition of many acoustic paths, the reverberant tail of the impulse response is well described statistically.

More precisely:

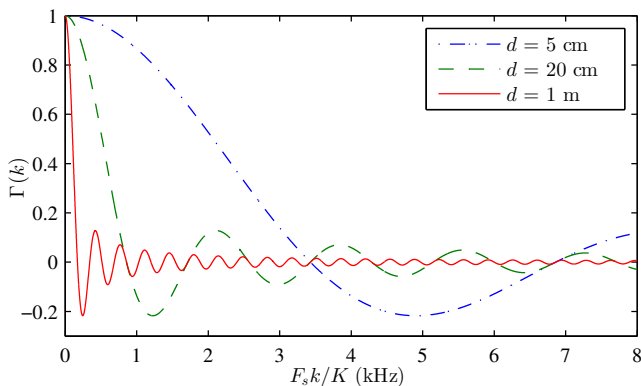
- ▶ it can be modeled as a zero-mean Gaussian noise signal whose amplitude decays exponentially over time according to the RT;
- ▶ its correlation over frequency decays quickly;
- ▶ it is approximately **diffuse**, i.e., it has similar power in all directions.

# Fundamental Acoustics

## Statistical Perspective

In a perfectly diffuse sound field, the correlation between two microphones in the Fourier domain is given by the **sine cardinal model**

$$\Gamma(k) = \text{sinc}\left(\frac{2\pi F_s k d}{cK}\right) = \frac{\sin(2\pi F_s k d / cK)}{2\pi F_s k d / cK} \quad d: \text{microphone distance}$$



# Outline

Introduction

Fundamental Acoustics

**Signal Models**

- Time-Domain Free-Field Model

- Time-Domain Reverberant-Field Model

- Time-Frequency Analysis and Synthesis

- STFT-Domain Multiplicative and Convolutional Models

- STFT-Domain Additive Model

- Extensions

Fundamental Array Processing

Data-Independent Beamforming

Data-Dependent Beamforming

Data-Dependent Source Separation

Summary and Perspectives

# Signal Models

## Time-Domain Free-Field Model

- In a free-field the direct path signal at the  $n$ -th microphone can be written as

$$x_n(t) = a_n(t) * s(t) = \int a_n(t') s(t - t') \, dt'$$

with

$$a_n(t) = \frac{1}{\sqrt{4\pi\ell_n}} \delta\left(t - \frac{\ell_n}{c}\right)$$

where  $t$  is the time index,  $s(t)$  is the anechoic source signal, and  $\ell_n$  is the distance between  $n$ -th microphone and the source.



# Signal Models

## Time-Domain Reverberant-Field Model

- ▶ Mathematically, we can formulate the room impulse response (RIR) as

$$a(t) = \sum_{i=1}^{\infty} r_i(t) * \delta(t - \tau_i),$$

where  $*$  denotes the convolution operation,  $\tau_i$  denotes the time-of-arrival of the  $i$ -th reflection and  $r_i(t)$  denotes the impulse response of the  $i$ -th reflection.

- ▶ The received signal at the  $n$ -th microphone can then be defined as

$$x_n(t) = \int a_n(t') s(t - t') \, dt'$$

where  $s(t)$  is the anechoic signal.

# Signal Models

## Time-Frequency Analysis and Synthesis

- ▶ We commonly work in the short-time Fourier transform (STFT) to exploit the temporal and spectral properties of the source signal.
- ▶ STFT Analysis:

$$X(m, k) = \sum_{r=0}^{L-1} x(mR + r)w_a(r)e^{-j\omega_k r} \quad \text{with} \quad \omega_k = \frac{2\pi k}{K},$$

$k = 0, 1, \dots, K-1$  is the frequency index (with  $K \geq L$ ),  $m$  is the time frame index, and  $R$  denotes the number of samples between two successive time frames.

- ▶ STFT Synthesis:

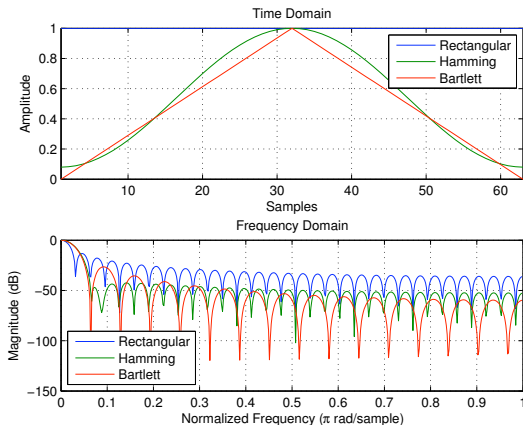
$$x(u) = \sum_m w_s(u - mR) \sum_{k=0}^{K-1} X(m, k)e^{j\omega_k(u - mR)},$$

where  $u$  is the discrete time index.

- ▶ The spectrogram is given by  $|X(m, k)|^2$ .

# Signal Models

## Time-Frequency Analysis and Synthesis - Window Functions



**Figure :** Rectangular, Hamming, and Bartlett windows. Note that an increased tapering of the window reduces the sidelobe level and increased the width of the main lobe.

# Signal Models

## Time-Frequency Analysis and Synthesis

- ▶ Completeness condition for analysis window ( $w_a$ ) and synthesis window ( $w_s$ ):

$$\sum_m w_a(u - mR)w_s(u - mR) = \frac{1}{L} \quad \text{for all } u. \quad (1)$$

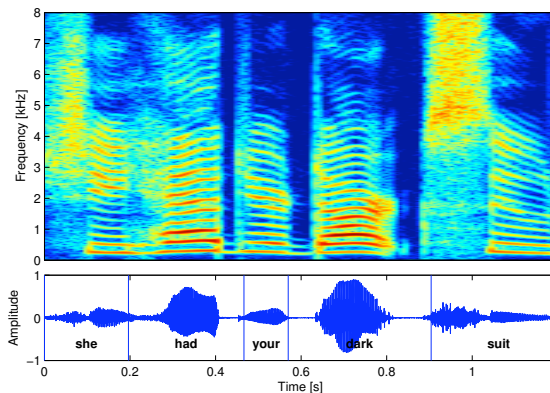
- ▶ Given analysis and synthesis windows that satisfy (1) we can reconstruct  $x(u)$  from its STFT coefficients  $X(m, k)$ .
- ▶ In practice, a Hamming window is often used for the synthesis window.
- ▶ A reasonable choice for the analysis window is the one with minimum energy (Wexler and Raz 1990), given by

$$w_a(u) = \frac{w_s(u)}{L \sum_m w_s^2(u - mR)}.$$

- ▶ The inverse STFT is efficiently implemented using the weighted overlap-add method (Crochiere and Rabiner 1983).

# Signal Models

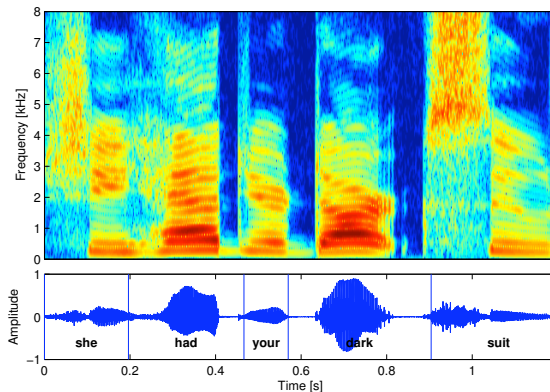
## Time-Frequency Analysis and Synthesis - Spectrogram



**Figure :** Spectrogram ( $10 \log(|X(m, k)|^2)$ ) of a speech signal (sample frequency 16 kHz, DFT length  $K = 1024$ , window length  $L = 512$  (32 ms), hamming window).

# Signal Models

## Time-Frequency Analysis and Synthesis - Spectrogram



**Figure :** Spectrogram ( $10 \log(|X(m, k)|^2)$ ) of a speech signal (sample frequency 16 kHz, DFT length  $K = 1024$ , window length  $L = 64$  (4 ms), hamming window).

# Signal Models

## STFT-Domain Multiplicative and Convolutional Models

- **Multiplicative Model:** When the STFT analysis frames are much longer than the RIR, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = A_n(k) S(m, k),$$

where  $A_n(k)$  is the Fourier transform of  $a_n(u)$  and  $S(m, k)$  is the STFT of the anechoic signal  $s(u)$ . As a consequence, the covariance matrix of  $\mathbf{x}(m, k) = [X_1(m, k), X_2(m, k), \dots, X_N(m, k)]^T$  is of rank-one.

- **Convolutional Model:** More generally, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = \sum_{k'} \sum_{m'=0}^{L'} A_n(m', k, k') S(m - m', k').$$

# Signal Models

## STFT-Domain Multiplicative and Convolutional Models

- **Multiplicative Model:** When the STFT analysis frames are much longer than the RIR, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = A_n(k) S(m, k),$$

where  $A_n(k)$  is the Fourier transform of  $a_n(u)$  and  $S(m, k)$  is the STFT of the anechoic signal  $s(u)$ . As a consequence, the covariance matrix of  $\mathbf{x}(m, k) = [X_1(m, k), X_2(m, k), \dots, X_N(m, k)]^T$  is of rank-one.

- **Convolutional Model:** More generally, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = \sum_{m'=0}^{L'} A_n(m', k) S(m - m', k).$$



# Signal Models

## STFT-Domain Multiplicative and Convolutional Models

- **Multiplicative Model:** When the STFT analysis frames are much longer than the RIR, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = A_n(k) S(m, k),$$

where  $A_n(k)$  is the Fourier transform of  $a_n(u)$  and  $S(m, k)$  is the STFT of the anechoic signal  $s(u)$ . As a consequence, the covariance matrix of  $\mathbf{x}(m, k) = [X_1(m, k), X_2(m, k), \dots, X_N(m, k)]^T$  is of rank-one.

- **Convolutional Model:** More generally, the received signal at the  $n$ -th microphone can be written as

$$X_n(m, k) = \underbrace{\sum_{m'=0}^{L'_d-1} A_n(m', k) S(m - m', k)}_{X_n^d(m, k)} + \underbrace{\sum_{m'=L'_d}^{L'} A_n(m', k) S(m - m', k)}_{X_n^r(m, k)}.$$

# Signal Models

## STFT-Domain Additive Model

- Alternatively, the  $n$ -th microphone can also be modelled in the STFT domain as the sum of a direct component and reverberant component:

$$X_n(k) = \underbrace{A_n^d(k, \theta)S(m, k)}_{X_n^d(m, k)} + X_n^r(m, k),$$

where  $A_n^d(k, \theta)$  models the direct path with  $\theta$  being the DOA of the direct sound, and  $X_n^r(m, k)$  models the reverberant signal component.

- In contrast to the convolutive model, it is commonly assumed that

$$\mathbb{E}\{X_n^d(m, k) (X_{n'}^r(m', k'))^*\} = 0 \quad \forall n, n', m, m', k, k'.$$

- The resulting covariance (PSD) matrix equals

$$\mathbb{E}\{\mathbf{x}(k)\mathbf{x}^H(k)\} = \mathbf{\Phi}_{\mathbf{x}_d}(k) + \mathbf{\Phi}_{\mathbf{x}_r}(k)$$

with

$$\mathbf{\Phi}_{\mathbf{x}_d}(m, k) = \phi_S(m, k)\mathbf{a}_d(k)\mathbf{a}_d^H(k) \text{ and } \mathbf{\Phi}_{\mathbf{x}_r}(m, k) = \mathbb{E}\{\mathbf{x}_r(m, k)\mathbf{x}_r^H(m, k)\}$$

which is an example of a **full-rank covariance matrix**.

# Signal Models

## STFT-Domain Additive Model (continued)

- ▶ The covariance matrix of the reverberant component

$$\Phi_{\mathbf{x}_r}(m, k) = E\{\mathbf{x}_r(m, k)\mathbf{x}_r^H(m, k)\}.$$

is often modelled as

$$\Phi_{\mathbf{x}_r}(m, k) = \phi_r(m, k) \mathbf{\Gamma}(k)$$

where in a perfectly homogenous and spherically isotropic sound field

$$[\mathbf{\Gamma}(k)]_{nn'} = \text{sinc}\left(\frac{2\pi F_s k d_{nn'}}{c K}\right),$$

where  $n$  and  $n'$  are microphone indices and  $d_{nn'}$  is the distance between these microphones.

- ▶ Hence, the spatial properties of the reverberant component are time-invariant and the temporal-spectral properties are time-variant.

- ▶ The  $n$ -th microphone signal can be written as

$$Y_n(m, k) = X_n(m, k) + V_n(m, k),$$

where  $V_n(m, k)$  denotes the additive noise as received by the  $n$ -th sensor.

- ▶ In the case of  $J$  directional sources, the  $n$ -th microphone signal can be written as

$$Y_n(m, k) = \sum_{j=1}^J X_{nj}(m, k) + V_n(m, k),$$

where  $X_{nj}(m, k)$  denotes the  $j$ -th source as received by the  $n$ -th microphone.

# Outline

Introduction

Fundamental Acoustics

Signal Models

**Fundamental Array Processing**

Spatial Sampling and Spatial Aliasing

Array Constellations

Near-field versus Far-Field

Beamforming

Design Criteria

Data-Independent Beamforming

Data-Dependent Beamforming

Data-Dependent Source Separation

Summary and Perspectives

# Fundamental Array Processing

## Spatial Sampling and Spatial Aliasing

- ▶ In practice, we can only sample the sound field at discrete positions, i.e., we use a discrete aperture rather than a continuous aperture.
- ▶ Temporal aliasing: signals with different temporal frequencies become indistinguishable when sampled across time.
- ▶ Spatial aliasing: waves with different spatial frequencies become indistinguishable when sampled across space.

**Example:** The smallest inter-microphone distance determines the highest frequency at which plane waves from different directions ( $0^\circ - 180^\circ$ ) result in a unique inter-microphone phase difference.

In this case, spatial aliasing does not occur when

$$d < \frac{c}{2f},$$

where  $c \text{ m s}^{-1}$  is the sound velocity and  $f$  is the frequency of the wave.

The spatial aliasing frequency is therefore  $f_{\text{sa}} = \frac{c}{2d}$ .

# Fundamental Array Processing

## Array Constellations



Figure : Cylindrical microphone arrays (top) and linear microphone array (bottom)

# Fundamental Array Processing

## Array Constellations

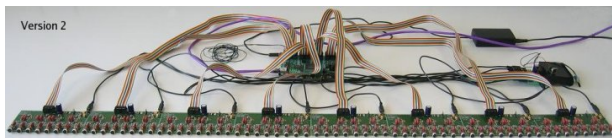


Figure : Spherical microphone arrays by mh acoustics.



# Fundamental Array Processing

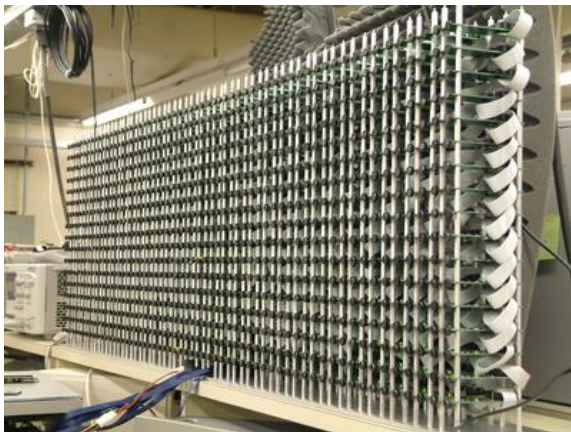
## Array Constellations



**Figure :** The NIST Mark-III microphone array (consists of 64 microphones)

# Fundamental Array Processing

## Array Constellations



**Figure :** The LOUD (Large acOUSTic Data) array is an array with 1020 microphones.  
See <http://groups.csail.mit.edu/cag/mic-array>

# Fundamental Array Processing

## Near-field versus Far-Field

- ▶ In the **near-field** the sound pressure of a wave measured at different positions differs both in amplitude and phase.
- ▶ When the source is far from the array we have

$$\frac{1}{\ell_1} \approx \frac{1}{\ell_2} \approx \dots \approx \frac{1}{\ell_N} \approx \frac{1}{r_s},$$

where  $\ell_n$  is the distance between the  $n$ -th microphone and the source and  $r_s$  is the distance from the reference point of the array to the source.

- ▶ The acoustic transfer function in a free-field then simplifies to

$$a_n(t) = \frac{1}{\sqrt{4\pi r_s}} \delta \left( t - \frac{r_s}{c} + \frac{\mathbf{r}_n^T \mathbf{u}_s}{c} \right),$$

where  $\mathbf{r}_n$  denotes the position vector of the  $n$ -th microphone and  $\mathbf{u}_s = \mathbf{r}_s / \|\mathbf{r}_s\|$  is a unit-norm vector pointing in the direction of the sound source.

- ▶ Hence, in the **far-field** the sound pressure of a wave measured at different positions differs only in phase. The wave can be modelled as a plane wave.

# Fundamental Array Processing

## Beamforming

- ▶ The task of the **beamforming algorithm** is to combine the microphone signals such that a desired, and possibly time-varying, spatial selectivity is achieved.
- ▶ Two major challenges:
  1. To design the microphone array
  2. To design the beamforming algorithm

# Fundamental Array Processing

## Beamforming - Delay and Sum Beamforming

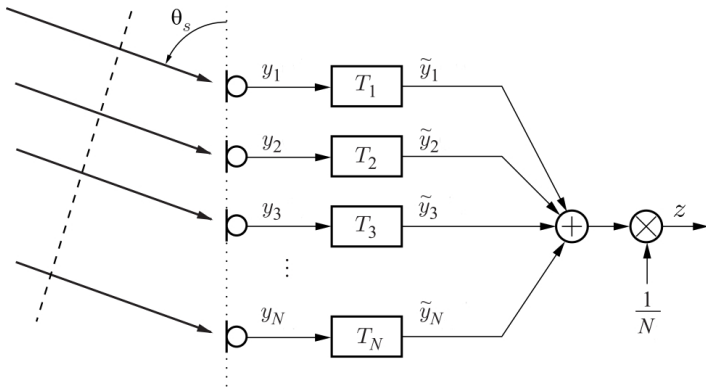


Figure : Block diagram of a delay-and-sum beamformer

# Fundamental Array Processing

## Beamforming - Delay and Sum Beamforming

- The microphone signals can be expressed as

$$y_n(t) = x_n(t) + v_n(t) = x_1(t - \tau_{n1}) + v_n(t).$$

where  $\tau_{n1}$  denotes the time difference of arrival w.r.t. the first microphone.

- After applying the channel dependent delay  $T_n$  we have

$$\tilde{y}_n(t) = x_1(t - \tau_{n1} - T_n) + v_n(t - T_n).$$

where  $T_n$  is the delay applied to the  $n$ -th microphone signal. Note that we need to ensure that all delays are positive! Therefore, a channel independent delay  $T_G$  is included such that  $T_n \geq 0 \forall n$ .

- The output of the **delay-and-sum beamformer** is computed using

$$\begin{aligned} z(t) &= \frac{1}{N} \sum_{n=1}^N \tilde{y}_n(t) \\ &= \frac{1}{N} \sum_{n=1}^N x_1(t - \tau_{n1} - T_n) + \frac{1}{N} \sum_{n=1}^N v_n(t - T_n) \\ &= \frac{1}{N} \sum_{n=1}^N x_1(t - T_G) + \frac{1}{N} \sum_{n=1}^N v_n(t - T_n) \end{aligned}$$

# Fundamental Array Processing

## Beamforming - Filter and Sum Beamforming

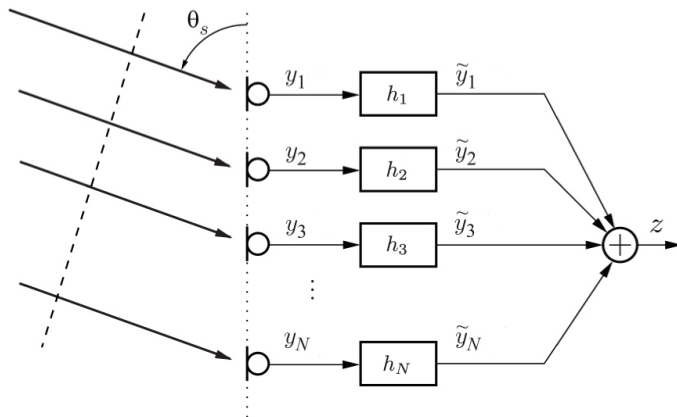


Figure : Block diagram of a filter-and-sum beamformer

# Fundamental Array Processing

## Beamforming - Filter and Sum Beamforming

- ▶ Time domain:

$$\begin{aligned} z(t) &= \sum_{n=1}^N h_n(t) * y_n(t). \\ &= \sum_{n=1}^N \int_0^{T'} h_n(t') y_n(t - t') dt'. \end{aligned}$$

- ▶ Short-time Fourier transform domain:

$$Z(m, k) = \sum_{n=1}^N H_n^*(k) Y_n(m, k) = \mathbf{h}^H(k) \mathbf{y}(m, k).$$



# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern

- ▶ The spatial response of a filter (i.e., **beam pattern**) is given by

$$\mathcal{S}(k) = \frac{\mathbf{h}^H(k) \mathbf{d}(k) X_1(m, k)}{X_1(m, k)} = \mathbf{h}^H(k) \mathbf{d}(k).$$

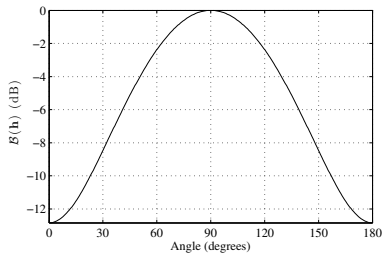
where  $\mathbf{d}(k)$  denotes the propagation vector.

- ▶ The **power pattern** is defined as the ratio of the variance of the beamformer output when the source impinges with a propagation vector  $\mathbf{d}(k)$  to the variance of the desired signal  $X_1(m, k)$ .
- ▶ From this definition we deduce the narrowband power patterns:

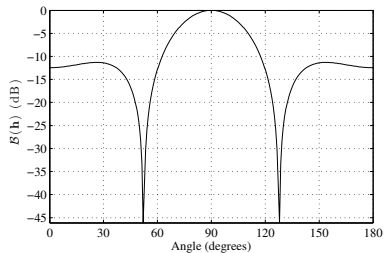
$$\begin{aligned} \mathcal{B}[\mathbf{d}(k)] &= \frac{E\{|\mathbf{h}^H(k) \mathbf{d}(k) X_1(m, k)|^2\}}{E\{|X_1(m, k)|^2\}} \\ &= |\mathbf{h}^H(k) \mathbf{d}(k)|^2. \end{aligned}$$

# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern



(a) 2 kHz

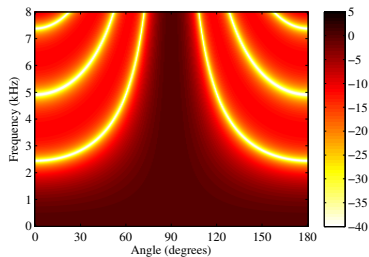


(b) 4 kHz

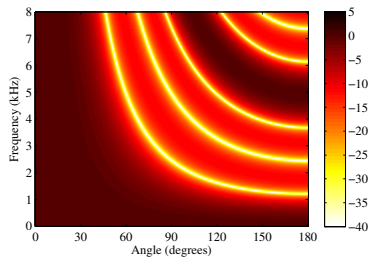
**Figure :** Power pattern of a ULA array with  $N = 4$  and  $d = 0.035$  m when using a delay-and-sum beamformer with  $\theta_s = 90^\circ$ .

# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern



(a) Broadside ( $\theta_s = 90^\circ$ )

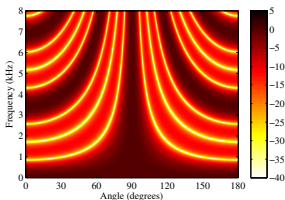


(b) Endfire ( $\theta_s = 0^\circ$ )

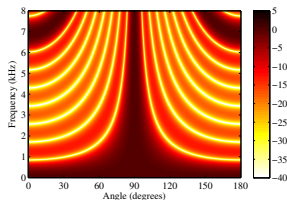
**Figure :** Power pattern of the delay-and-sum beamformer with  $N = 4$  microphones and  $d = 0.035$  m for the broadside (left) and endfire (right) orientation

# Fundamental Array Processing

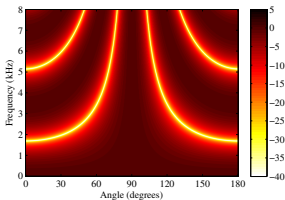
## Design Criteria - Beam Pattern and Power Pattern



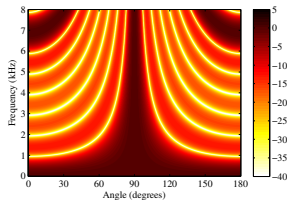
(a)  $N = 4$  and  $d = 10$  cm



(b)  $N = 8$  and  $d = 5$  cm



(c)  $N = 2$  and  $d = 10$  cm

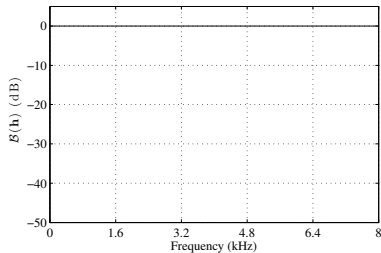


(d)  $N = 7$  and  $d = 5$  cm

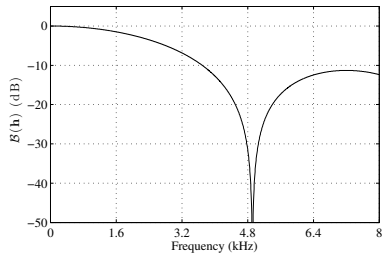
**Figure :** Power patterns of a delay-and-sum beamformer for different array configurations

# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern



(a)  $\theta = 90^\circ$



(b)  $\theta = 60^\circ$

**Figure :** Frequency response for two different directions for a ULA array with  $N = 4$  when  $\theta_s = 90^\circ$ ,  $d = 0.035$  m.

# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern

- ▶ **Main lobe** The angular region between two nulls which contains the angle where the array is steered towards and hence  $\mathcal{B}[\mathbf{d}(k)] = \mathcal{B}_{\max}$ .
- ▶ **Side lobes** The angular regions between two nulls which do not contain the angle where  $\mathcal{B}[\mathbf{d}(k)] = \mathcal{B}_{\max}$ .
- ▶ **Sidelobe level (SLL)** The power of the highest sidelobe relative to  $\mathcal{B}_{\max}$ .
- ▶ **Grating lobes** The angular region between two nulls which contains the angle where the array is **not** steered towards and  $\mathcal{B}[\mathbf{d}(k)] = \mathcal{B}_{\max}$ . Happens at and above the spatial aliasing frequency.
- ▶ **Nulls** Angle at which  $\mathcal{B}[\mathbf{d}(k)] = 0$ .
- ▶ **Half-power beamwidth (HPBW)** The angle spanned by the region for which  $\mathcal{B}_{\max}/2 \leq \mathcal{B} \leq \mathcal{B}_{\max}$ . The HPBW is often referred to as the 3 dB beamwidth.
- ▶ **First null beamwidth (FNBW)** The angle spanned by the main lobe. The FNBW is associated with the ability of a microphone array to reject an interference. For a ULA and delay-and-sum beamformer we obtain:

$$\text{FNBW} \approx \frac{2c}{fNd}$$

# Fundamental Array Processing

## Design Criteria - Beam Pattern and Power Pattern

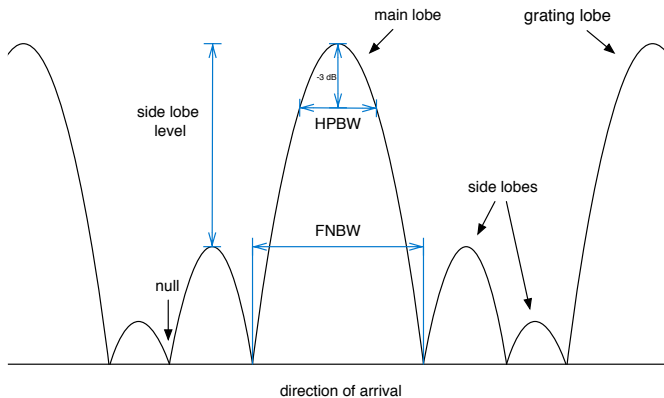


Figure : Illustration of a power pattern

# Fundamental Array Processing

## Design Criteria - Directivity Factor and Directivity Index

- ▶ The **directivity index (DI)** quantifies the ability of the beamformer to reduce spherically isotropic or diffuse noise.
- ▶ The **narrowband directivity factor** is given by

$$\begin{aligned}\mathcal{D}[\mathbf{h}(k)] &= \frac{\mathcal{B}[\mathbf{d}(k)]}{\frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi \mathcal{B}[\mathbf{d}(k, \phi, \theta)] \sin(\phi) \, d\phi \, d\theta} \\ &= \frac{\mathcal{B}[\mathbf{d}(k)]}{\mathbf{h}^H(k) \mathbf{\Gamma}(k) \mathbf{h}(k)} = \frac{|\mathbf{h}^H(k) \mathbf{d}(k)|^2}{\mathbf{h}^H(k) \mathbf{\Gamma}(k) \mathbf{h}(k)}\end{aligned}$$

where

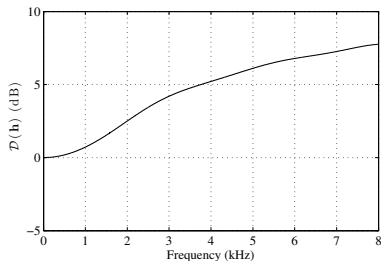
$$[\mathbf{\Gamma}(k)]_{nn'} = \text{sinc} \left( \frac{2\pi F_s k d_{nn'}}{c K} \right).$$

- ▶ The classical **directivity index** is given by  $\mathcal{DI}(k) = 10 \log_{10} \mathcal{D}(k)$ .

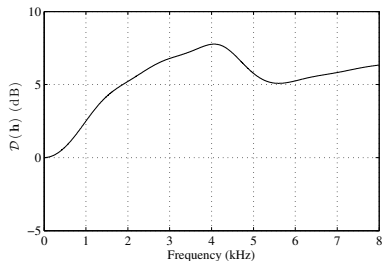


# Fundamental Array Processing

## Design Criteria - Directivity Factor and Directivity Index



(a) Broadside



(b) Endfire

**Figure :** Directivity index of the delay-and-sum beamformer with  $N = 4$  microphones and  $d = 0.035$  m for the broadside (left) and endfire (right) orientation

# Fundamental Array Processing

## Design Criteria - White Noise Gain and Sensitivity

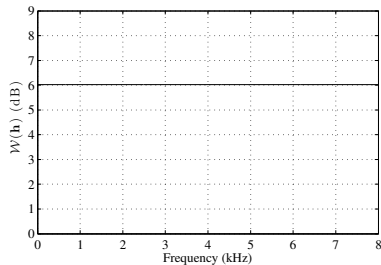
- ▶ The **white noise gain (WNG)** quantifies the ability of the beamformer to reduce spatially white noise.
- ▶ The WNG is also used as a measure for the robustness of the filters w.r.t. microphone gain and phase mismatches as well as microphone position errors.
- ▶ The **narrowband white noise gain** is given by

$$\mathcal{W}[\mathbf{h}(k)] = \frac{|\mathbf{h}^H(k)\mathbf{d}(k)|^2}{\mathbf{h}^H(k)\mathbf{h}(k)} = \frac{\mathcal{B}[\mathbf{d}(k)]}{\mathbf{h}^H(k)\mathbf{h}(k)}.$$

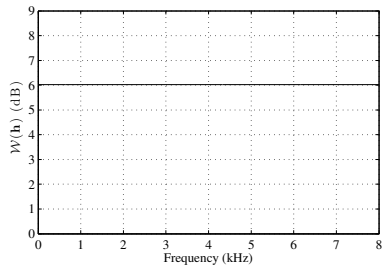
- ▶ The sensitivity is given by  $1/\mathcal{W}[\mathbf{h}(k)]$ . Hence, we can compare the sensitivity of different filter-and-sum beamformers by evaluating  $1/\mathcal{W}[\mathbf{h}(k)]$  for different spatial filters  $\mathbf{h}(k)$ .

# Fundamental Array Processing

## Design Criteria - White Noise Gain and Sensitivity



(a) Broadside



(b) Endfire

**Figure :** White noise gain of the delay-and-sum beamformer with  $N = 4$  microphones and  $d = 0.035$  m for the broadside (left) and endfire (right) orientation

# Outline

Introduction

Fundamental Acoustics

Signal Models

Fundamental Array Processing

**Data-Independent Beamforming**

- General overview

- Distortionless Response Beamformer

- Maximum White Noise Gain Beamformer

- Maximum Directivity Beamformer

- Robust Super-Directive Beamformer

- Jointly Optimizing Spatial and Frequency Response

- Post-filtering

- Application Example

Data-Dependent Beamforming

Data-Dependent Source Separation

Summary and Perspectives

# Data-Independent Beamforming

## General overview

- ▶ In general, data-independent beamformers are suitable for applications where the position of the source of interest is known in advance.
- ▶ When the source position is unknown a sound source localization / tracking system can be used to determine the **look direction**.
- ▶ Data-independent beamformers are commonly referred to as **fixed beamformers**.
- ▶ The beamformers are designed to obtain a spatial focus on the sound source of interest while minimizing sensor noise, reverberation, and sounds arriving from other locations.
- ▶ Compared with data-dependent beamformers, fixed beamformers require substantially lower computational complexity.

# Data-Independent Beamforming

## Distortionless Response Beamformer

- First we need to carefully define the desired signal!

1) A **distortionless response for the desired signal**  $S$  is obtained when

$$\begin{aligned}\mathbf{h}^H(k) \mathbf{a}(k) S(m, k) &= S(m, k) \\ \Rightarrow \mathbf{h}^H(k) \underbrace{\mathbf{a}(k)}_{\mathbf{d}(k)} &= 1\end{aligned}$$

2) A **distortionless response for the desired signal**  $X_1 = A_1 S$  is obtained when

$$\begin{aligned}\mathbf{h}^H(k) \mathbf{a}(k) S(m, k) &= A_1(k) S(m, k) \\ \Rightarrow \mathbf{h}^H(k) \mathbf{a}(k) &= A_1(k) \\ \Rightarrow \mathbf{h}^H(k) \underbrace{\frac{\mathbf{a}(k)}{A_1(k)}}_{\mathbf{d}(k)} &= 1\end{aligned}$$

- Hence, any spatial filter for which  $\mathbf{h}^H(k) \mathbf{d}(k) = 1$  provides a distortionless response for a source with propagation vector  $\mathbf{d}(k)$ .

# Data-Independent Beamforming

## Maximum White Noise Gain Beamformer

- ▶ Let us assume  $\Phi_{\mathbf{v}}(k) = \phi_V(k)\mathbf{I}$ , i.e., the noise is spatially white and identically distributed such that  $\phi_V = \phi_{V_1} = \phi_{V_2} = \dots = \phi_{V_N}$ .
- ▶ The **maximum white noise gain beamformer** is given by

$$\begin{aligned}\mathbf{h}_{\text{WNG}}(k) &= \underset{\mathbf{h}}{\operatorname{argmax}} \frac{|\mathbf{h}^H \mathbf{d}(k)|^2}{\mathbf{h}^H \mathbf{h}} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{d}(k) = 1 \\ &= \underset{\mathbf{h}}{\operatorname{argmin}} \mathbf{h}^H \mathbf{h} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{d}(k) = 1 \\ &= \frac{\mathbf{d}(k)}{\mathbf{d}^H(k) \mathbf{d}(k)}.\end{aligned}$$

- ▶ In free-field conditions we can write the propagation vector as:

$$\mathbf{d}(k) = \left[ 1, \exp\left(-j \frac{2\pi k f_s}{K} \tau_2\right), \dots, \exp\left(-j \frac{2\pi k f_s}{K} \tau_N\right) \right]^T.$$

such that

$$\mathbf{h}_{\text{WNG}}(k) = \frac{1}{N} \mathbf{d}(k),$$

which is equal to the delay-and-sum beamformer.

# Data-Independent Beamforming

## Maximum White Noise Gain Beamformer (free-field)

- Now the output signal can be written as

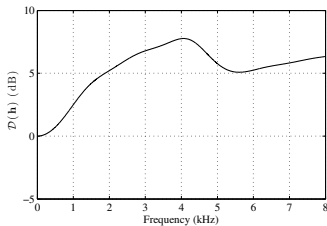
$$\begin{aligned} Z(m, k) &= \mathbf{h}^H(k) \mathbf{y}(m, k) \\ &= \frac{1}{N} \sum_{n=1}^N D_n^*(k) Y_n(m, k) \\ &= \frac{1}{N} \sum_{n=1}^N [D_n^*(k) D_n(k) X_1(m, k) + D_n^*(k) V_n(m, k)] \\ &= \frac{1}{N} \sum_{n=1}^N X_1(m, k) + \frac{1}{N} \sum_{n=1}^N D_n^*(k) V_n(m, k) \\ &= X_1(m, k) + \frac{1}{N} \sum_{n=1}^N D_n^*(k) V_n(m, k). \end{aligned}$$

- Note that the spectrum of the residual noise depends on the steering vector  $\mathbf{d}(k)$ !

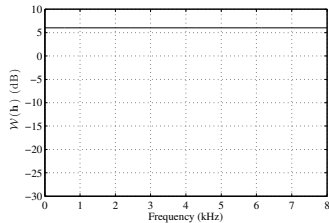


# Data-Independent Beamforming

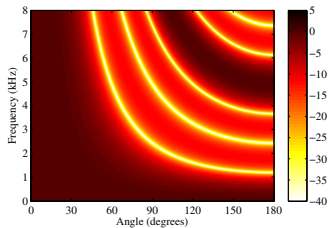
## Maximum White Noise Gain Beamformer (free-field)



(a) Directivity index



(b) WNG



(c) Beam pattern for  $\theta = 0^\circ$  and  $N = 4$

# Data-Independent Beamforming

## Maximum Directivity Beamformer

- ▶ The **maximum directivity beamformer** is given by

$$\begin{aligned}\mathbf{h}_{\text{SD}}(k) &= \underset{\mathbf{h}}{\operatorname{argmax}} \frac{|\mathbf{h}^H \mathbf{d}(k)|^2}{\mathbf{h}^H \mathbf{\Gamma} \mathbf{h}} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{d}(k) = 1 \\ &= \underset{\mathbf{h}}{\operatorname{argmin}} \mathbf{h}^H \mathbf{\Gamma} \mathbf{h} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{d}(k) = 1 \\ &= \frac{\mathbf{\Gamma}^{-1}(k) \mathbf{d}(k)}{\mathbf{d}^H(k) \mathbf{\Gamma}^{-1}(k) \mathbf{d}(k)}.\end{aligned}$$

- ▶ In a homogenous and spherically isotropic sound field the  $(n, n')$ -th element of  $\mathbf{\Gamma}(k)$  equals  $\operatorname{sinc}(2\pi k f_s d_{nn'} (K c)^{-1})$ , where  $d_{nn'}$  is the distance between the  $n$ -th and  $n'$ -th microphones.
- ▶ Using these weights the obtained directivity equals

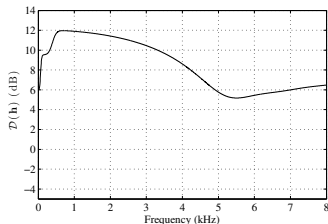
$$\text{DF}(k) = \mathbf{d}^H(k) \mathbf{\Gamma}^{-1}(k) \mathbf{d}(k).$$

It can be proven that **the largest attainable directivity equals  $N^2$** .

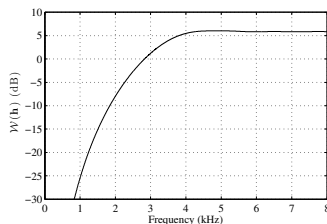
- ▶ This beamformer is also known as a super-directive beamformer.

# Data-Independent Beamforming

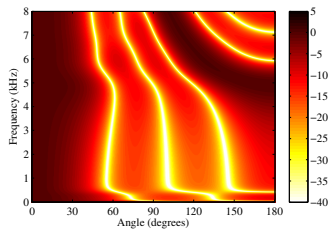
## Maximum Directivity Beamformer (free-field)



(d) Directivity Index



(e) WNG



(f) Beam pattern for  $\theta = 0^\circ$  and  $N = 4$

# Data-Independent Beamforming

## Robust Super-Directive Beamformer

- By enforcing a lower bound on the WNG we can find a more robust solution by adding an inequality constraint:

$$\mathbf{h}_{\text{RSD}}(k) = \underset{\mathbf{h}}{\operatorname{argmin}} \mathbf{h}^H \mathbf{\Gamma}(k) \mathbf{h}$$

$$\text{subject to } \mathbf{h}^H \mathbf{d}(k) = 1 \text{ and } \frac{1}{\mathbf{h}^H \mathbf{h}} \geq \text{const}(k).$$

- This optimization problem is non-convex. The solution has the following form:

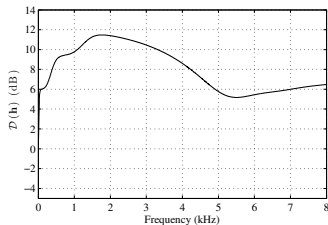
$$\mathbf{h}_{\text{RSD}}(k) = \frac{[\mathbf{\Gamma}(k) + c(k)\mathbf{I}]^{-1} \mathbf{d}(k)}{\mathbf{d}^H(k) [\mathbf{\Gamma}(k) + c(k)\mathbf{I}]^{-1} \mathbf{d}(k)},$$

- An iterative procedure can be used to find  $c(k)$  such that

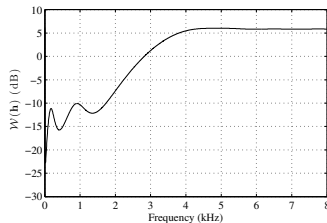
$$\frac{1}{\mathbf{h}_{\text{RSD}}^H(k) \mathbf{h}_{\text{RSD}}(k)} \geq \text{const}(k).$$

# Data-Independent Beamforming

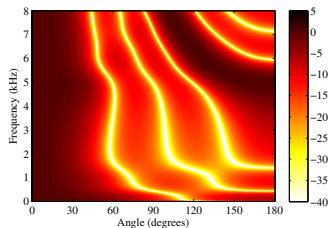
## Robust Super-Directive Beamformer (free-field)



(g) Directivity index



(h) WNG



(i) Beam pattern for  $\theta = 0^\circ$ ,  $N = 4$ , and  $\text{const}(k) = 0.0001$

# Data-Independent Beamforming

## Jointly Optimizing Spatial and Frequency Response

Design of a fixed beamformer with a desired beam pattern:

- ▶ The beam pattern (far-field model) of a filter is given by:

$$\mathcal{S}(\omega, \theta) = \sum_{n=1}^N H_n^*(\omega) e^{-j\omega \tau_{n1}(\theta)}$$

where  $\tau_{n1}(\theta)$  is the time difference of arrival for the  $n$ -th microphone w.r.t. the first microphone.

- ▶ We can express  $\mathcal{S}(\omega, \theta)$  as a function of the time-domain filter coefficients, i.e.,  $\mathcal{S}(\omega, \theta) = \mathbf{w}^T \mathbf{a}(\omega, \theta)$  where  $\mathbf{w} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_N^T]^T$  and  $\mathbf{a}(\omega, \theta) = [\mathbf{u}^T \exp(-j\omega \tau_{11}(\theta)), \dots, \mathbf{u}^T \exp(-j\omega \tau_{N1}(\theta))]^T$  with  $\mathbf{u} = [\exp(-j\omega 0), \dots, \exp(-j\omega (L-1))]^T$ .
- ▶ We can now define a weighted-LS approximation criterion:

$$J_{\text{LS}}(\mathbf{w}) = \int_{\Theta} \int_{\Omega} \nu(\omega, \theta) |\mathcal{S}(\omega, \theta) - \mathcal{S}_d(\omega, \theta)|^2 d\omega d\theta,$$

where  $\nu(\omega, \theta)$  is a weighting function to emphasize the importance of certain angles and frequencies and  $\mathcal{S}_d(\omega, \theta)$  is the desired beam pattern.

# Data-Independent Beamforming

## Jointly Optimizing Spatial and Frequency Response

- ▶ The weighted-LS approximation criterion can be written as a quadratic function

$$J_{\text{LS}}(\mathbf{w}) = \mathbf{w}^T \mathbf{Q} \mathbf{w} - 2 \mathbf{w}^T \mathbf{p} + \int_{\Theta} \int_{\Omega} \nu(\omega, \theta) |\mathcal{S}_d(\omega, \theta)|^2 d\omega d\theta$$

with

$$\mathbf{Q} = \int_{\Theta} \int_{\Omega} \nu(\omega, \theta) \text{Re}\{\mathbf{a}(\omega, \theta) \mathbf{a}^H(\omega, \theta)\} d\omega d\theta$$
$$\mathbf{p} = \int_{\Theta} \int_{\Omega} \nu(\omega, \theta) \text{Re}\{\mathcal{S}_d(\omega, \theta) \mathbf{a}^*(\omega, \theta)\} d\omega d\theta.$$

- ▶ Differentiating the cost function w.r.t.  $\mathbf{w}$  and equating the result to zero gives

$$\mathbf{w}_{\text{LS}} = \mathbf{Q}^{-1} \mathbf{p}.$$

- ▶ Numerical optimization methods can be used to compute the integrals.
- ▶ For further reading including adding constraints and robust design methods see (Doclo and Moonen 2003).

# Data-Independent Beamforming

## Jointly Optimizing Spatial and Frequency Response

Design specification:

- Passband  $(\omega_p, \theta_p) = (300 - 4000 \text{ Hz}, 70^\circ - 110^\circ)$
- Stopband  $(\omega_s, \theta_s) = (300 - 4000 \text{ Hz}, 0^\circ - 60^\circ + 120^\circ - 180^\circ)$

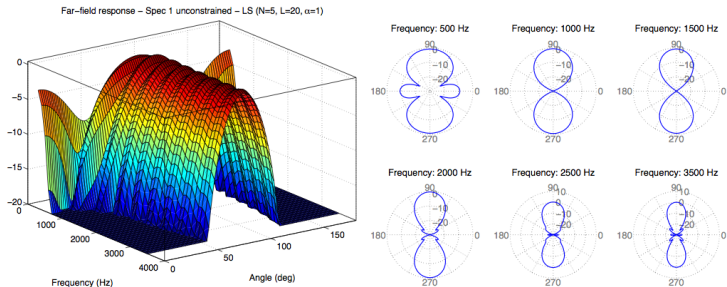


Figure : Weighted LS technique (no constraints,  $\nu = 1$ ,  $N = 5$ ,  $L = 20$ ) as shown in (Doclo and Moonen 2003).



# Data-Independent Beamforming

## Post-filtering

- ▶ Post-filters were initially developed because the noise reduction performance of data-independent beamformers was found to be insufficient.
- ▶ Hence, the objective of the post-filter is to reduce the noise at the output of the beamformer.
- ▶ In principle, any single-channel speech enhancement techniques can be employed.
- ▶ Under specific assumptions, the speech and noise PSDs at the output of the beamformer can be computed without the need for a voice activity detector.
- ▶ Many of the existing microphone array post-filters are derived under the following assumptions:
  1. The desired speech PSD at the sensors are equal:  
 $\phi_X = \phi_{X_1} = \phi_{X_2} = \dots = \phi_{X_N}$ .
  2. The noise-field is homogeneous :  $\phi_V = \phi_{V_1} = \phi_{V_2} = \dots = \phi_{V_N}$ .
  3. The desired speech and noise are uncorrelated.

# Data-Independent Beamforming

## Post-filtering - An Example

- Under these assumptions the microphone PSD matrix is given by

$$\Phi_y = \phi_X \mathbf{d}\mathbf{d}^H + \phi_V \Gamma_v \quad \text{with} \quad |D_1|^2 = |D_2|^2 = \dots = |D_N|^2 = 1.$$

- McCowan (2002) showed that  $\hat{\phi}_X$  can be estimated as

$$\hat{\phi}_X = \frac{2}{N(N-1)} \sum_{n=1}^{N-1} \sum_{n'=n+1}^N \frac{\text{Re}\{\phi_{Y_n Y_{n'}}\} - \frac{1}{2}(\phi_{Y_n} + \phi_{Y_{n'}})\text{Re}\{\Gamma_{V_n V_{n'}}\}}{1 - \text{Re}\{\Gamma_{V_n V_{n'}}\}}.$$

- Leukimmiatis et al. (2006) showed that  $\phi_V$  can be estimated as

$$\hat{\phi}_V = \frac{2}{N(N-1)} \sum_{n=1}^{N-1} \sum_{n'=n+1}^N \frac{\frac{1}{2}(\phi_{Y_n} + \phi_{Y_{n'}}) - \text{Re}\{\phi_{Y_n Y_{n'}}\}}{1 - \text{Re}\{\Gamma_{V_n V_{n'}}\}}.$$

- The noise PSD at the beamformer's output equals  $\hat{\phi}_V \mathbf{h}_{\text{MVDR}}^H \Gamma_v \mathbf{h}_{\text{MVDR}}$ .
- The single-channel Wiener filter is then given by:

$$H_W = \frac{\hat{\phi}_X}{\hat{\phi}_X + \hat{\phi}_V \mathbf{h}_{\text{MVDR}}^H \Gamma_v \mathbf{h}_{\text{MVDR}}}.$$

# Data-Independent Beamforming

## Application Example

- ▶ Setup
  - Array: Uniform linear array with  $N = 4$  omni-directional microphones
  - Source: Positioned at  $0^\circ$  (i.e., endfire)
  - Noise: Sensor noise (WGN) and babble speech
- ▶ We compare the reference microphone with
  1. Delay-and-sum beamformer
  2. Super-directive beamformer
  3. Super-directive beamformer with Leukimmiatis's post-filter

# Outline

Introduction

Fundamental Acoustics

Signal Models

Fundamental Array Processing

Data-Independent Beamforming

**Data-Dependent Beamforming**

- General overview

- Performance Measures

- Maximum SNR Beamformer

- Minimum Variance Distortionless Response Beamformer

- Linearly Constrained Minimum Variance Beamformer

- Generalized Sidelobe Canceller

- Multichannel Wiener Filter (Single Source)

- Parametric Multichannel Wiener Filter (Single Source)

- Detection and Estimation

- Parametric Spatial Filtering

- Application Examples

Data-Dependent Source Separation

Summary and Perspectives

# Data-Dependent Beamforming

## General overview

- ▶ Data-dependent beamformers are able to adjust to the acoustic situation at hand.
- ▶ For a given optimization criteria, the solution can be obtained in closed-form or using adaptive techniques.
- ▶ In the following we focus on closed-form solutions.
- ▶ These closed-form solutions also allow us to analyze and compare different beamformers.

# Data-Dependent Beamforming

## Performance Measures - Signal-to-Noise Ratio and Array Gain

- ▶ The narrowband input signal to noise ratio (SNR) is given by:

$$\text{iSNR}(m, k) = \frac{\phi_{X_1}(m, k)}{\phi_{V_1}(m, k)},$$

where  $\phi_{X_1}(m, k) = E\{|X_1(m, k)|^2\}$  and  $\phi_{V_1}(m, k) = E\{|V_1(m, k)|^2\}$ .

- ▶ The narrowband output SNR

$$\text{oSNR}[\mathbf{h}(m, k)] = \frac{\phi_{X_1}(m, k) |\mathbf{h}^H(m, k)\mathbf{d}(k)|^2}{\mathbf{h}^H(m, k)\mathbf{\Phi}_v(m, k)\mathbf{h}(m, k)}$$

where  $\mathbf{\Phi}_v(m, k)$  denotes the noise covariance matrix (a.k.a. PSD matrix).

- ▶ Finally, the narrowband array gain is given by

$$\mathcal{A}[\mathbf{h}(m, k)] = \frac{\text{oSNR}[\mathbf{h}(m, k)]}{\text{iSNR}(m, k)}.$$

# Data-Dependent Beamforming

## Performance Measures - Noise Reduction and Speech Distortion

- ▶ The narrowband noise reduction is given:

$$\xi_{\text{nr}}[\mathbf{h}(m, k)] = \frac{\phi_{V_1}(m, k)}{\mathbf{h}^H(m, k) \mathbf{\Phi}_{\mathbf{v}}(m, k) \mathbf{h}(m, k)}.$$

This value is expected to be lower bounded by 1; otherwise the filter amplifies the noise in sub-band  $k$ .

- ▶ The narrowband speech distortion is given by:

$$\xi_{\text{sd}}[\mathbf{h}(m, k)] = \frac{\phi_{X_1}(m, k)}{\mathbf{h}^H(m, k) \mathbf{\Phi}_{\mathbf{x}}(m, k) \mathbf{h}(m, k)} = \frac{1}{|\mathbf{h}^H(m, k) \mathbf{d}(k)|^2}.$$

This value is expected to be lower bounded by 1.

- ▶ We can now easily verify that we have the following fundamental relation:

$$\mathcal{A}[\mathbf{h}(m, k)] = \frac{\text{oSNR}[\mathbf{h}(m, k)]}{\text{iSNR}(m, k)} = \frac{\xi_{\text{nr}}[\mathbf{h}(m, k)]}{\xi_{\text{sd}}[\mathbf{h}(m, k)]}.$$

# Data-Dependent Beamforming

## Performance Measures - Speech Distortion Index

- ▶ Another commonly used measure is the narrowband **speech distortion index**:

$$\begin{aligned}\nu_{\text{sd}}[\mathbf{h}(m, k)] &= \frac{\mathcal{E}\{|\mathbf{h}^{\text{H}}(m, k)X_1(m, k) - X_1(m, k)|^2\}}{\mathcal{E}\{|X_1(m, k)|^2\}} \\ &= \left| \mathbf{h}^{\text{H}}(m, k)\mathbf{d}(k) - 1 \right|^2.\end{aligned}$$

- ▶ These performance measure are useful to derive and compare different beamformers.
- ▶ To evaluate the full-band performance, it is recommended to first transform the individual signals back to the time-domain and then to calculate segmental measures in the time-domain.



# Data-Dependent Beamforming

## Maximum SNR Beamformer

The oSNR can be written as:

$$\text{oSNR}[\mathbf{h}(m, k)] = \frac{\mathbf{h}^H(m, k) \mathbf{\Phi}_x(m, k) \mathbf{h}(m, k)}{\mathbf{h}^H(m, k) \mathbf{\Phi}_v(m, k) \mathbf{h}(m, k)}$$

where  $\mathbf{\Phi}_x(m, k) = \phi_{X_1}(m, k) \mathbf{d}(m, k) \mathbf{d}^H(m, k)$ .

The maximum SNR filter is given by

$$\begin{aligned} \mathbf{h}_{\max}(m, k) &= \underset{\mathbf{h}}{\operatorname{argmax}} \frac{\mathbf{h}^H \mathbf{\Phi}_x(m, k) \mathbf{h}}{\mathbf{h}^H \mathbf{\Phi}_v(m, k) \mathbf{h}} \quad \frac{\mathbf{\Phi}_x \mathbf{h} \mathbf{h}^H \mathbf{\Phi}_v \mathbf{h} - \mathbf{\Phi}_v \mathbf{h} \mathbf{h}^H \mathbf{\Phi}_x \mathbf{h}}{(\mathbf{h}^H \mathbf{\Phi}_v \mathbf{h})^2} = 0 \\ &= \rho(m, k) \mathbf{\Phi}_v^{-1}(m, k) \mathbf{d}(m, k) \end{aligned}$$

with  $\rho(m, k) \neq 0$ . The filter is equal to  $\rho$  times the eigenvector corresponding to the largest eigenvalue  $[\lambda_{\max}(m, k)]$  of the matrix  $\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)$ .

Since the rank of  $\mathbf{\Phi}_x(m, k)$  is one, the maximum output SNR is given by

$$\begin{aligned} \text{oSNR}[\mathbf{h}_{\max}(m, k)] &= \lambda_{\max}(m, k) = \operatorname{tr}\{\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)\} \\ &= \phi_{X_1}(m, k) \mathbf{d}^H(m, k) \mathbf{\Phi}_v^{-1}(m, k) \mathbf{d}(m, k). \end{aligned}$$

# Data-Dependent Beamforming

## Minimum Variance Distortionless Response Beamformer

We can minimize the residual noise  $\mathcal{E}\{|\mathbf{h}^H(m, k)\mathbf{v}(m, k)|^2\}$  with the constraint that the desired signal is not distorted. Mathematically, this is equivalent to

$$\mathbf{h}_{\text{MVDR}}(m, k) = \underset{\mathbf{h}}{\operatorname{argmin}} \mathbf{h}^H \Phi_{\mathbf{v}}(m, k) \mathbf{h} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{d}(k) = 1.$$

Using Lagrange multipliers we obtain the solution:

$$\mathbf{h}_{\text{MVDR}}(m, k) = \frac{\Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)}{\mathbf{d}^H(k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)}.$$

When the desired signal is  $X_1$ , this can be written as (Benesty, Chen, and Huang 2008)

$$\begin{aligned} \mathbf{h}_{\text{MVDR}}(m, k) &= \frac{\phi_{X_1}(m, k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)}{\operatorname{tr}\{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)\}} \\ &= \frac{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)}{\operatorname{tr}\{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)\}} \mathbf{i}_{N,1} \\ &= \frac{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{y}}(m, k) - \mathbf{I}_{N \times N}}{\operatorname{tr}\{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{y}}(m, k)\} - N} \mathbf{i}_{N,1}. \end{aligned}$$

# Data-Dependent Beamforming

## Linearly Constrained Minimum Variance Beamformer

- ▶ Let us adopt the multi-source model with  $J$  sources.
- ▶ The desired signal is given by

$$Z(m, k) = \sum_{j=1}^J Q_j^*(k) X_{1j}(m, k)$$

where  $Q_j^*(k)$  denotes the desired response for the  $j$ -th source.

- ▶ We can now minimize the residual noise at the output of the beamformer subject to the constraint

$$\mathbf{h}^H(m, k) \mathbf{D}(k) = \mathbf{q}^H(k)$$

with

$$\mathbf{D}(k) = [\mathbf{d}_1(k) \ \mathbf{d}_2(k) \ \cdots \ \mathbf{d}_J(k)] .$$

and

$$\mathbf{q}(k) = [Q_1(k) \ Q_2(k) \ \cdots \ Q_J(k)]^T .$$

# Data-Dependent Beamforming

## Linearly Constrained Minimum Variance Beamformer

- ▶ We can minimize the residual noise  $\mathcal{E}\{|\mathbf{h}^H(m, k)\mathbf{v}(m, k)|^2\}$  with the constraint that the desired signal is not distorted.
- ▶ Mathematically, this is equivalent to (Er and Cantoni 1983)

$$\mathbf{h}_{\text{LCMV}}(m, k) = \underset{\mathbf{h}}{\operatorname{argmin}} \mathbf{h}^H \Phi_{\mathbf{v}}(m, k) \mathbf{h} \quad \text{subject to} \quad \mathbf{h}^H \mathbf{D}(k) = \mathbf{q}^H(k).$$

The solution is given by

$$\mathbf{h}_{\text{LCMV}}(m, k) = \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{D}(k) \left[ \mathbf{D}^H(k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{D}(k) \right]^{-1} \mathbf{q}(k),$$

- ▶ The LCMV beamformer can be interpreted as a two stage spatial processor that first computes  $J$  signals given by  $\mathbf{D}^H(k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{y}(m, k)$ . Finally, these signals are combined using  $\mathbf{q}^H(k) \left[ \mathbf{D}^H(k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{D}(k) \right]^{-1}$  to compute the output of the LCMV beamformer  $Z(m, k)$ .

# Data-Dependent Beamforming

## Generalized Sidelobe Canceller

- ▶ The weights of the MVDR beamformer span an  $N$  dimensional space.
- ▶ This space can be divided into two orthogonal subspaces, i.e., a constraint subspace and an orthogonal subspace.
- ▶ The constraint subspace is defined by the column space of  $\mathbf{d}$  (with rank 1), and the orthogonal subspace is defined by left null space of  $\mathbf{d}$  (with rank  $N - 1$ ).
- ▶ Following this decomposition we can represent the MVDR filter as

$$\mathbf{h}_{\text{MVDR}}(m, k) = \mathbf{h}_c(m, k) - \mathbf{B}(k)\mathbf{h}_{\text{nc}}(m, k),$$

where  $\mathbf{h}_c(m, k)$  lies in the constraint subspace and  $-\mathbf{B}(k)\mathbf{h}_{\text{nc}}(m, k)$  lies in the orthogonal subspace. The matrix  $\mathbf{B}(k)$  is referred to as the blocking matrix and  $\mathbf{h}_{\text{nc}}(m, k)$  is referred to as the noise cancellation filter.

- ▶ The blocking matrix is chosen such that

$$\mathbf{d}^H \mathbf{B} = \mathbf{0}_{1 \times N-1}.$$

Consequently, any vector that lies in the column space of  $\mathbf{B}$  (and hence null space of  $\mathbf{d}^H$ ) lies in the orthogonal subspace.

# Data-Dependent Beamforming

## Generalized Sidelobe Canceller

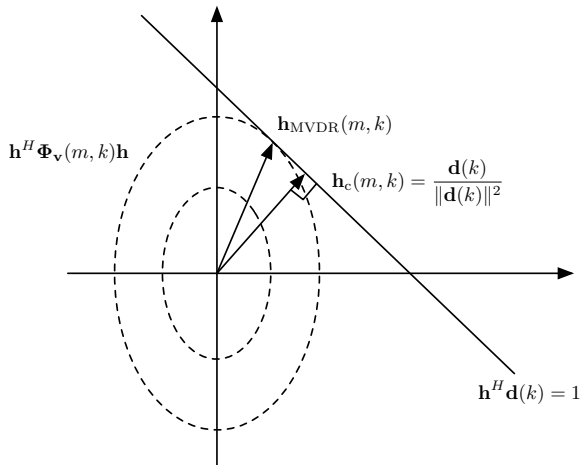


Figure : Constrained minimization

# Data-Dependent Beamforming

## Generalized Sidelobe Canceller

- ▶ The filter  $\mathbf{h}_c(m, k)$  can be obtained by projecting the MVDR filter on the constraint subspace:

$$\mathbf{h}_c(m, k) = \mathbf{d}(k) \left[ \mathbf{d}^H(k) \mathbf{d}(k) \right]^{-1} \mathbf{d}^H(k) \mathbf{h}_{\text{MVDR}}(m, k) = \frac{\mathbf{d}(k)}{\|\mathbf{d}(k)\|^2},$$

- ▶ An example of the blocking matrix, known as a sparse blocking matrix, is

$$\mathbf{B}(k) = \begin{pmatrix} -\frac{G_2^*(k)}{G_1^*(k)} & -\frac{G_3^*(k)}{G_1^*(k)} & \cdots & -\frac{G_N^*(k)}{G_1^*(k)} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- ▶ A closed-form solution for the noise cancellation filter of size  $(N - 1) \times 1$  is given by

$$\mathbf{h}_{\text{nc}}(m, k) = [\mathbf{B}^H(k) \mathbf{\Phi}_v(m, k) \mathbf{B}(k)]^{-1} \mathbf{B}^H(k) \mathbf{\Phi}_v(m, k) \mathbf{h}_c(m, k).$$

# Data-Dependent Beamforming

## Generalized Sidelobe Canceller

- ▶ This structure is called a generalized sidelobe canceller (GSC).
- ▶ It is often preferred for adaptive implementations as it allows us to write a constrained optimization problem as an unconstrained optimization problem.

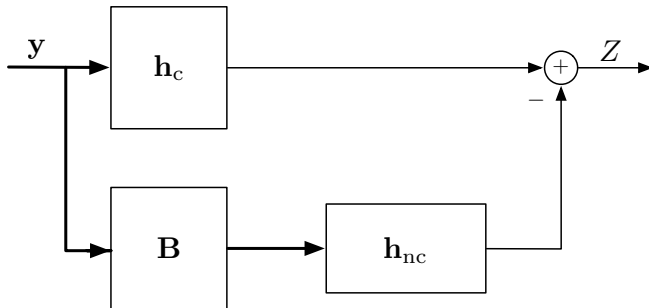


Figure : Generalized sidelobe structure.



# Data-Dependent Beamforming

## Multichannel Wiener Filter (Single Source)

The Wiener filter provides the smallest mean squared error (MSE):

$$\begin{aligned} J[\mathbf{h}(m, k)] &= \mathcal{E}\{|\mathbf{h}^H(m, k)\mathbf{y}(m, k) - X_1(m, k)|^2\} \\ &= \phi_{X_1}(m, k) + \mathbf{h}^H(m, k)\mathbf{\Phi}_{\mathbf{y}}(m, k)\mathbf{h}(m, k) \\ &\quad - \phi_{X_1}(m, k) \left( \mathbf{h}^H(m, k)\mathbf{d}(k) + \mathbf{d}^H(k)\mathbf{h}(m, k) \right). \end{aligned}$$

We can now compute the Wiener filter by taking the derivative w.r.t.  $\mathbf{h}^H(m, k)$  and equate the expression to zero:

$$\mathbf{h}_W(m, k) = \phi_{X_1}(m, k)\mathbf{\Phi}_{\mathbf{y}}^{-1}(m, k)\mathbf{d}(k).$$

We can also express the Wiener filter as:

$$\begin{aligned} \mathbf{h}_W(m, k) &= \mathbf{\Phi}_{\mathbf{y}}^{-1}(m, k)\mathcal{E}\{\mathbf{x}(m, k)X_1^*(m, k)\} \\ &= \mathbf{\Phi}_{\mathbf{y}}^{-1}(m, k)\mathbf{\Phi}_{\mathbf{x}}(m, k)\mathbf{i}_{N,1} \\ &= [\mathbf{I}_{N \times N} - \mathbf{\Phi}_{\mathbf{y}}^{-1}(m, k)\mathbf{\Phi}_{\mathbf{v}}(m, k)]\mathbf{i}_{N,1}, \end{aligned}$$

where  $\mathbf{I}_{N \times N}$  is the identity matrix of size  $N \times N$ .

# Data-Dependent Beamforming

## Multichannel Wiener Filter (Single Source)

- ▶ We can decompose the multichannel Wiener filter into a MVDR filter followed by a single-channel Wiener filter.
- ▶ We can write the Wiener filter as:

$$\mathbf{h}_W(m, k) = \frac{\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)}{1 + \text{tr}\{\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)\}} \mathbf{i}_{N,1}$$

- ▶ It can then easily be verified that

$$\mathbf{h}_W(m, k) = \mathbf{h}_{\text{MVDR}}(m, k) H_W(m, k)$$

with

$$\begin{aligned} H_W(m, k) &= \frac{\text{tr}\{\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)\}}{1 + \text{tr}\{\mathbf{\Phi}_v^{-1}(m, k) \mathbf{\Phi}_x(m, k)\}} \\ &= \frac{\text{oSNR}[\mathbf{h}_{\text{MVDR}}(m, k)]}{1 + \text{oSNR}[\mathbf{h}_{\text{MVDR}}(m, k)]} \\ &= \frac{\mathbf{h}_{\text{MVDR}}^H(m, k) \mathbf{\Phi}_x(m, k) \mathbf{h}_{\text{MVDR}}(m, k)}{\mathbf{h}_{\text{MVDR}}^H(m, k) [\mathbf{\Phi}_x(m, k) + \mathbf{\Phi}_v(m, k)] \mathbf{h}_{\text{MVDR}}(m, k)}. \end{aligned}$$

# Data-Dependent Beamforming

## Parametric Multichannel Wiener Filter (Single Source)

In order to provide the ability to control the amount of noise reduction and speech distortion we consider the following optimization problem:

$$J[\mathbf{h}(m, k)] = \mathcal{E}\{|\mathbf{h}^H(m, k)\mathbf{x}(m, k) - X_1(m, k)|^2\} + \mu \mathcal{E}\{|\mathbf{h}^H(m, k)\mathbf{v}(m, k)|^2\}$$

where  $\mu$  is the trade-off parameter. The first term is related to the speech distortion and the second term is related to the residual noise.

The solution for the  $k$ th sub-band is given by

$$\begin{aligned}\mathbf{h}_{\text{PW},\mu}(m, k) &= \phi_{X_1}(m, k) [\Phi_{\mathbf{x}}(m, k) + \mu \Phi_{\mathbf{v}}(m, k)]^{-1} \mathbf{d}(k) \\ &= \frac{\phi_{X_1}(m, k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)}{\mu + \phi_{X_1}(m, k) \mathbf{d}^H(k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)} \\ &= \frac{\phi_{X_1}(m, k) \Phi_{\mathbf{v}}^{-1}(m, k) \mathbf{d}(k)}{\mu + \text{tr}\{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)\}} \\ &= \frac{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)}{\mu + \text{tr}\{\Phi_{\mathbf{v}}^{-1}(m, k) \Phi_{\mathbf{x}}(m, k)\}} \mathbf{i}_{N,1}.\end{aligned}$$

# Data-Dependent Beamforming

## Parametric Multichannel Wiener Filter (Single Source)

We consider the following special cases:

- ▶  $\mu = 0$ :  $\mathbf{h}_{\text{PW},0} = \mathbf{h}_{\text{MVDR}}$ , which is the MVDR filter.
- ▶  $\mu = 1$ :  $\mathbf{h}_{\text{PW},1} = \mathbf{h}_{\text{W}}$ , which is the Wiener filter.
- ▶  $\mu > 1$ : Results in a filter producing **low residual noise** (compared to the Wiener filter) at the expense of **high speech distortion**.
- ▶  $\mu < 1$ : Results in a filter producing **high residual noise** (compared to the Wiener filter) and **low speech distortion**.

It can then easily be verified that

$$\mathbf{h}_{\text{PW},\mu}(m, k) = H_{\text{PW},\mu}(m, k) \mathbf{h}_{\text{MVDR}}(m, k)$$

with

$$\begin{aligned} H_{\text{PW},\mu}(m, k) &= \frac{\text{tr}\{\mathbf{\Phi}_{\mathbf{v}}^{-1}(m, k)\mathbf{\Phi}_{\mathbf{x}}(m, k)\}}{\mu(k) + \text{tr}\{\mathbf{\Phi}_{\mathbf{v}}^{-1}(m, k)\mathbf{\Phi}_{\mathbf{x}}(m, k)\}} \\ &= \frac{\mathbf{h}_{\text{MVDR}}^{\text{H}}(m, k)\mathbf{\Phi}_{\mathbf{x}}(m, k)\mathbf{h}_{\text{MVDR}}(m, k)}{\mathbf{h}_{\text{MVDR}}^{\text{H}}(m, k)[\mathbf{\Phi}_{\mathbf{x}}(m, k) + \mu\mathbf{\Phi}_{\mathbf{v}}(m, k)]\mathbf{h}_{\text{MVDR}}(m, k)}. \end{aligned}$$

# Data-Dependent Beamforming

## Detection and Estimation

- ▶ To compute the spatial filters we need to estimate the second-order statistics (SOS) of the desired signal(s) and the undesired signal.
- ▶ One possibility is to use a detection/classification and estimation approach.
- ▶ The detector/classifier can exploit spatial, spectral, and/or temporal features and will tell us, for example, whether the desired-plus-undesired or undesired sounds are active.
- ▶ For example, if the hypothesis is true that the undesired signal is active, the SOS of the undesired signal can be updated.
- ▶ In some cases, additional prior information can be incorporated such as a (pre-defined) region of interest.

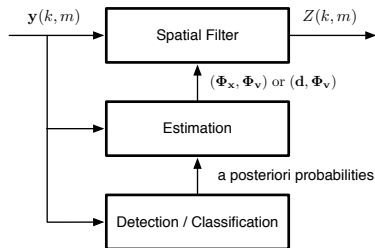


Figure : General structure

# Data-Dependent Beamforming

## PSD Estimation

- ▶ Let us define two hypotheses:

- ▶  $\mathcal{H}_0 : \mathbf{y}(m, k) = \mathbf{v}(m, k)$  (speech absence)

- ▶  $\mathcal{H}_1 : \mathbf{y}(m, k) = \mathbf{x}(m, k) + \mathbf{v}(m, k)$  (speech presence)

- ▶ The PSD matrix of the observed noisy signals can be estimated using:

$$\hat{\Phi}_{\mathbf{y}}(m, k) = \alpha_y(m, k) \hat{\Phi}_{\mathbf{y}}(m-1, k) + [1 - \alpha_y(m, k)] \mathbf{y}(m, k) \mathbf{y}^H(m, k).$$

- ▶ The PSD matrix of the noise signals can be estimated using:

$$\hat{\Phi}_{\mathbf{v}}(m, k) = \alpha_v(m, k) \hat{\Phi}_{\mathbf{v}}(m-1, k) + [1 - \alpha_v(m, k)] \mathbf{y}(m, k) \mathbf{y}^H(m, k).$$

- ▶ The smoothing parameters  $\alpha_y(m, k)$  and  $\alpha_v(m, k)$  need to be controlled, for example, based on the posterior probabilities,  $p[\mathcal{H}_1|\mathbf{y}]$  and  $p[\mathcal{H}_0|\mathbf{y}]$ .

# Data-Dependent Beamforming

## RTF Estimation

- ▶ We can obtain an estimate of the RTF w.r.t.  $X_1$  in the MMSE sense by solving:

$$\hat{\mathbf{d}}(m, k) = \underset{\mathbf{d}}{\operatorname{argmin}} \mathcal{E}\{|\mathbf{x}(m, k) - \mathbf{d}X_1(m, k)|^2\}.$$

The solution is given by

$$\begin{aligned}\hat{\mathbf{d}}(m, k) &= \frac{\mathcal{E}\{\mathbf{x}(m, k)X_1^*(m, k)\}}{\mathcal{E}\{|X_1(m, k)|^2\}} \\ &= \frac{\mathbf{\Phi}_{\mathbf{x}}(m, k)\mathbf{i}_{N,1}}{\mathbf{i}_{N,1}^T \mathbf{\Phi}_{\mathbf{x}}(m, k)\mathbf{i}_{N,1}} \quad \text{with } \mathbf{i}_{N,1} = [1 \ 0 \ \dots \ 0]^T.\end{aligned}$$

- ▶ By using the fact that  $\mathbf{x} = \mathbf{a}S$ , we can verify that

$$\hat{\mathbf{d}}(k) = \frac{\mathbf{a}(k)}{A_1(k)}.$$

- ▶ Using the fact that  $\mathbf{\Phi}_{\mathbf{y}}(m, k) = \mathbf{\Phi}_{\mathbf{x}}(m, k) + \mathbf{\Phi}_{\mathbf{v}}(m, k)$ :

$$\hat{\mathbf{d}}(m, k) = \frac{[\mathbf{\Phi}_{\mathbf{y}}(m, k) - \mathbf{\Phi}_{\mathbf{v}}(m, k)]\mathbf{i}_{N,1}}{\mathbf{i}_{N,1}^T [\mathbf{\Phi}_{\mathbf{y}}(m, k) - \mathbf{\Phi}_{\mathbf{v}}(m, k)]\mathbf{i}_{N,1}}.$$

# Data-Dependent Beamforming

## RTF Estimation using Non-Stationarity

- ▶ The desired speech is non-stationary and the statistics of the noise very slowly compared to the statistics of the speech (Gannot, Burshtein, and Weinstein 2001).
- ▶ The microphone signal can be expressed as

$$Y_n(m, k) = D_n(k)Y_1(m, k) + U_n(m, k)$$

where

$$U_n(m, k) = V_n(m, k) - D_n(k)V_1(m, k) \quad \text{and} \quad D_n(k) = \frac{A_n(k)}{A_1(k)}.$$

- ▶ Multiplying both sides with  $Y_1^*(m, k)$  and taking the expectation yields:

$$\hat{\phi}_{Y_n Y_1}(m, k) = D_n(k)\hat{\phi}_{Y_1 Y_1}(m, k) + \phi_{U_n Y_1}(m, k) + \epsilon_n(m, k)$$

where  $\epsilon_n(m, k) = \hat{\phi}_{U_n Y_1}(m, k) - \phi_{U_n Y_1}(m, k)$ .

- ▶ With a short time period of  $L$  frames we can assume the noise is stationary such that  $\phi_{U_n Y_1}(m, k) = \phi_{U_n Y_1}(k)$ .



# Data-Dependent Beamforming

## RTF Estimation using Non-Stationarity

- We can collect estimates for  $L$  frames and construct the following overdetermined set of equations:

$$\begin{bmatrix} \hat{\phi}_{Y_n Y_1}(m, k) \\ \hat{\phi}_{Y_n Y_1}(m-1, k) \\ \vdots \\ \hat{\phi}_{Y_n Y_1}(m-L+1, k) \end{bmatrix} = \begin{bmatrix} \hat{\phi}_{Y_1 Y_1}(m, k) & 1 \\ \hat{\phi}_{Y_1 Y_1}(m-1, k) & 1 \\ \vdots & \\ \hat{\phi}_{Y_1 Y_1}(m-L+1, k) & 1 \end{bmatrix} \begin{bmatrix} D_n(k) \\ \phi_{U_n Y_1}(k) \end{bmatrix} + \begin{bmatrix} \epsilon_n(m, k) \\ \epsilon_n(m-1, k) \\ \vdots \\ \epsilon_n(m-L+1, k) \end{bmatrix}$$

for  $n \in \{2, 3, \dots, N\}$ .

# Data-Dependent Beamforming

## RTF Estimation using Non-Stationarity

- An unbiased estimate of  $D_n(k)$  is now given by

$$\hat{D}_n(m, k) = \frac{\langle \hat{\phi}_{Y_1 Y_1}(m, k) \hat{\phi}_{Y_n Y_1}(m, k) \rangle - \langle \hat{\phi}_{Y_1 Y_1}(m, k) \rangle \langle \hat{\phi}_{Y_n Y_1}(m, k) \rangle}{\langle \hat{\phi}_{Y_1 Y_1}^2(m, k) \rangle - \langle \hat{\phi}_{Y_1 Y_1}(m, k) \rangle^2}$$

with

$$\langle A(m, k) \rangle \triangleq \frac{1}{L} \sum_{m'=0}^{L-1} A(m - m', k).$$

## Parametric Spatial Filtering

- ▶ Parametric spatial filtering incorporate nearly instantaneous information about the acoustic scene in the design of the (spatial) filter.
- ▶ We can differentiate between two types of parametric spatial filters:
  1. A single-channel filter that is computed based on spatial parameters and SOS.
  2. A multi-channel filter that is computed based on spatial parameters and SOS.
- ▶ Commonly used spatial parameters are DOA, signal-to-diffuse ratio, interaural phase differences, interaural level differences, etc.

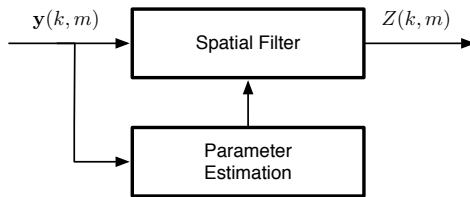
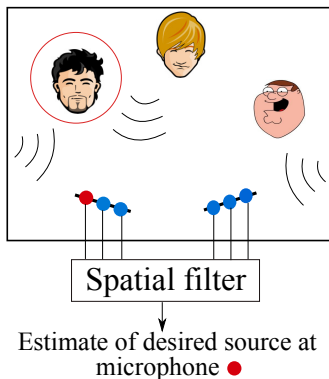


Figure : General parametric spatial filtering structure

# Application Examples

## Source Extraction: Problem Formulation



## Scenario

- ▶ Multiple talkers
- ▶ Additive background noise
- ▶ Distributed sensor arrays

## Applications

- ▶ Teleconferencing systems
- ▶ Automatic speech recognition
- ▶ Spatial sound reproduction

▶ **Signal model:**  $\mathbf{y}(m, k) = \mathbf{x}_{j^{\text{target}}}(m, k) + \sum_{j \neq j^{\text{target}}} \mathbf{x}_j(m, k) + \mathbf{v}(m, k).$

- ▶ **Aim:** Obtain an MMSE estimate of  $X_{1_{j^{\text{target}}}}(m, k).$

# Application Examples

Source Extraction: Proposed Solution (Taseska and Habets 2014)

- Hypotheses:

$$\mathcal{H}_{\mathbf{v}} : \mathbf{y}(m, k) = \mathbf{v}(m, k) \rightarrow \text{speech absent}$$

$$\mathcal{H}_{\mathbf{x}} : \mathbf{y}(m, k) = \sum_j \mathbf{x}_j(m, k) + \mathbf{v}(m, k) \rightarrow \text{speech present}$$

$$\mathcal{H}_{\mathbf{x}_j} : \mathbf{y}(m, k) = \mathbf{x}_j(m, k) + \underbrace{\sum_{j' \neq j}^J \mathbf{x}_{j'}(m, k)}_{\approx 0} + \mathbf{v}(m, k) \quad j = 1, 2, \dots, J$$

- Recursive estimation of the PSD matrices:

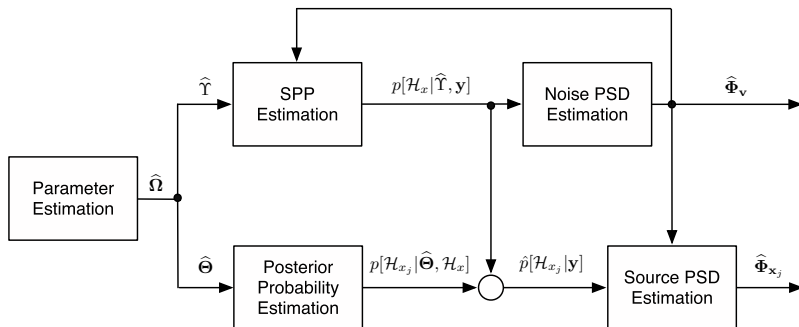
$$\begin{aligned} \hat{\Phi}_{\mathbf{x}_j + \mathbf{v}}(m) = p[\mathcal{H}_{\mathbf{x}_j} | \mathbf{y}] & \left( \alpha_x \hat{\Phi}_{\mathbf{x}_j + \mathbf{v}}(m-1) + (1 - \alpha_x) \mathbf{y} \mathbf{y}^H \right) \\ & + (1 - p[\mathcal{H}_{\mathbf{x}_j} | \mathbf{y}]) \hat{\Phi}_{\mathbf{x}_j + \mathbf{v}}(m-1) \end{aligned}$$

- Signal-to-diffuse ratio ( $\Upsilon$ ) and position ( $\Theta$ ) based posterior probabilities:

$$p[\mathcal{H}_{\mathbf{x}_j} | \mathbf{y}] = p[\mathcal{H}_{\mathbf{x}_j} | \mathbf{y}, \mathcal{H}_{\mathbf{x}}] \cdot p[\mathcal{H}_{\mathbf{x}} | \mathbf{y}] \approx p[\mathcal{H}_{\mathbf{x}_j} | \Theta, \mathcal{H}_{\mathbf{x}}] \cdot p[\mathcal{H}_{\mathbf{x}} | \Upsilon, \mathbf{y}]$$

# Application Examples

## Source Extraction: Parameter-based PSD Matrix Estimation



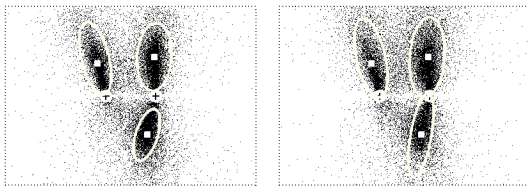
- ▶ The distribution  $p[\hat{\Theta} | \mathcal{H}_x]$  is modelled as a Gaussian mixture (GM).
- ▶ GM parameters estimated by the Expectation-Maximization algorithm.
- ▶ For more details see (Taseska and Habets 2014).

# Application Examples

## Source Extraction: Results (1)

### Setup:

- ▶ Three reverberant sources with approximately equal power, diffuse babble speech ( $\text{SNR}=22$  dB), and uncorrelated sensor noise ( $\text{SNR}=50$  dB). The reverberation time was  $T60 = 250$  ms.
- ▶ Two uniform circular arrays were used with three omnidirectional microphones, a diameter 2.5 cm and an inter-array spacing of 1.5 m.



(a) Training during single-talk      (b) Training during triple-talk

**Figure :** Output of the EM algorithm (3 iterations) and 4.5 s of noisy speech data. The actual source positions are denoted by white squares. The array location is marked by a plus symbol. The interior of each ellipse contains 85% probability mass of the respective Gaussian.

# Application Examples

## Source Extraction: Results (2)

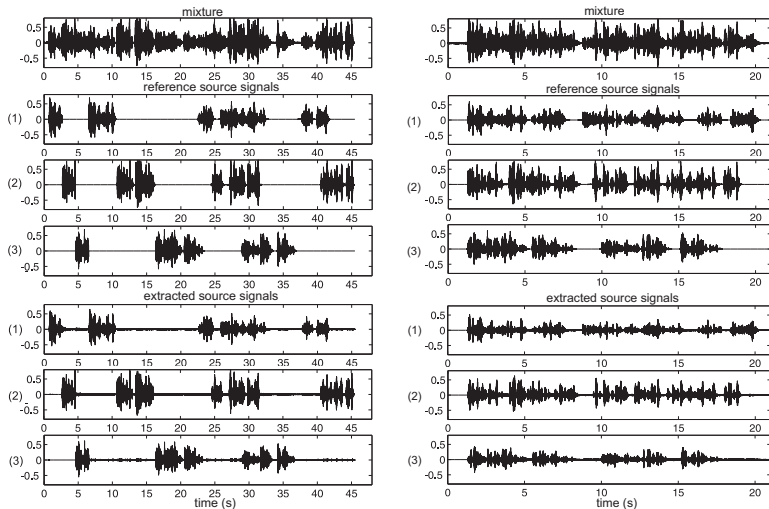


Figure : Left: constant single-talk scenario. Right: mainly triple-talk scenario. (S,M)



# Application Examples

## Directional Filtering

- ▶ Flexible sound acquisition in noisy and reverberant environments with rapidly changing acoustic scenes is a common problem in modern communication systems.
- ▶ A spatial filter is proposed that provides an **arbitrary spatial response** for  $J$  sources being simultaneously active per time and frequency.
- ▶ The spatial filter provides an **optimal tradeoff** between the white noise gain (WNG) and the directivity index.
- ▶ The filter is controlled by nearly instantaneous information (i.e., narrowband DOAs and diffuse-to-noise ratio) to respond quickly to changes in the acoustic scene.

# Application Examples

## Directional Filtering: Problem Formulation

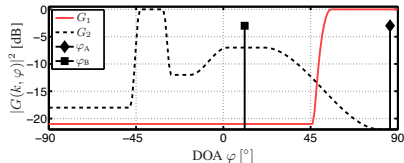
- **Signal model:** Based on a multi-wave sound field model, the  $N$  microphone signals can be expressed as:

$$\mathbf{y}(m, k) = \underbrace{\sum_{j=1}^J \mathbf{x}_j(m, k)}_{J \text{ plane waves}} + \underbrace{\mathbf{x}_r(m, k)}_{\text{diffuse sound}} + \underbrace{\mathbf{v}(m, k)}_{\text{sensor noise } (\Phi_{\mathbf{v}} = \phi_V \mathbf{I})}$$

- **Aim:** Capturing  $J$  plane waves ( $J \leq N$ ) with desired arbitrary gain while attenuating the sensor noise and reverberation.

The desired signal is given by:

$$Z(m, k) = \sum_{j=1}^J G(k, \varphi_j) X_{1j}(m, k).$$



- The desired signal is estimated using an informed LCMV filter:

$$\hat{Z}(m, k) = \mathbf{h}_{\text{iLCMV}}^H(m, k) \mathbf{y}(m, k).$$

# Application Examples

## Directional Filtering: Proposed Solution (1)

- ▶ The proposed informed LCMV filter is given by:

$$\begin{aligned} \mathbf{h}_{\text{iLCMV}} = \underset{\mathbf{h}}{\operatorname{argmin}} \quad & \mathbf{h}^H [\Phi_{\mathbf{x}_r}(m, k) + \Phi_{\mathbf{v}}(m, k)] \mathbf{h} \\ \text{s. t.} \quad & \mathbf{h}^H(m, k) \mathbf{a}_d(k, \varphi_j) = G(k, \varphi_j), \quad j \in \{1, 2, \dots, J\} \end{aligned}$$

where  $\mathbf{a}_d(k, \varphi_j)$  denotes the steering vector for the  $j$ th plane wave at time  $m$  and frequency  $k$ .

For the assumed signal model, we can alternatively minimize

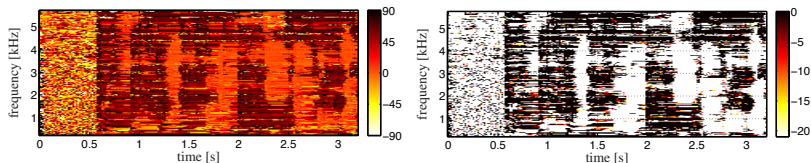
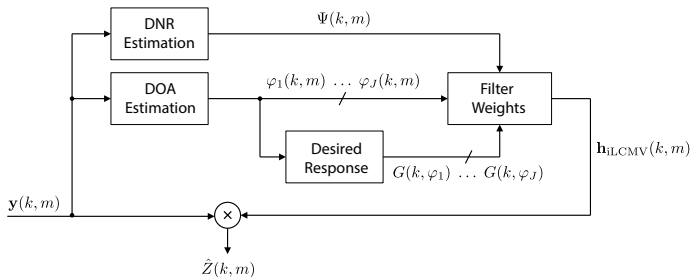
$$\mathbf{h}^H [\Psi(m, k) \Gamma_d(k) + \mathbf{I}] \mathbf{h},$$

where  $\Psi(m, k) = \phi_r(m, k)/\phi_v(m, k)$  denotes the diffuse-to-noise ratio (DNR) and  $\Gamma(k)$  denotes the spatial coherence matrix of the diffuse sound field.

- ▶ The filter is updated for each time and frequency given the parametric information (i.e., DOAs and DNR).
- ▶ The filter requires knowledge of the DNR, which can be estimated using an auxiliary spatial filter (c.f. (Thiergart, Taseska, and Habets 2014)).

# Application Examples

## Directional Filtering: Proposed Solution (2)



**Figure :** Left: DOA  $\varphi_1(m, k)$  as a function of time and frequency. Right: Desired response  $|G(k, \varphi_1)|^2$  in dB for DOA  $\varphi_1(m, k)$  as a function of time and frequency.

# Application Examples

## Directional Filtering: Results (1)

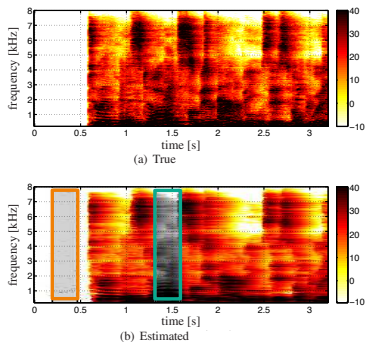


Figure : Top: True DNR in dB. Bottom: Estimated DNR in dB.

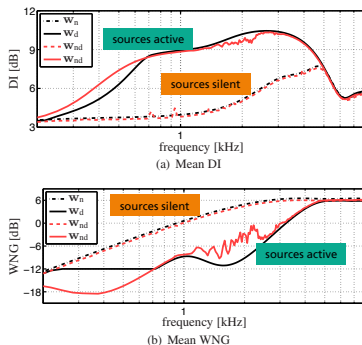


Figure : Top: Directivity index (DI) in dB. Bottom: White noise gain (WNG) in dB.  $w_n$  minimizes the noise power,  $w_d$  minimizes the diffuse power,  $w_{nd}$  is the proposed iLCMV filter that minimizes the diffuse plus noise power [shown when the sources are active (red solid line) and silent (red dashed line)].

# Application Examples

## Directional Filtering: Results (2)

- ▶ The proposed spatial filter provides a high DI when the sound field is diffuse and a high WNG when the sensor noise is dominant.
- ▶ Interfering sound can be strongly attenuated if desired.
- ▶ The proposed DNR estimator provides a sufficiently high accuracy and temporal resolution to allow signal enhancement under adverse conditions even in changing acoustic scenes.

	SegSIR [dB]		SegSRR [dB]		SegSNR [dB]		PESQ	
*	11	(11)	-7	(-7)	26	(26)	1.5	(1.5)
$\mathbf{w}_n$	21	(32)	-2	(-3)	<b>33</b>	<b>(31)</b>	2.0	(1.7)
$\mathbf{w}_d$	<b>26</b>	<b>(35)</b>	0	(-1)	22	(24)	<b>2.1</b>	<b>(2.0)</b>
$\mathbf{w}_{nd}$	25	<b>(35)</b>	<b>1</b>	(-1)	28	(26)	<b>2.1</b>	<b>(2.0)</b>

**Table :** Performance of all spatial filters [\* unprocessed, first sub-column using true DOAs (of the sources), second sub-column using estimated DOAs (of the plane waves)]. Audio Examples

# Outline

Introduction

Fundamental Acoustics

Signal Models

Fundamental Array Processing

Data-Independent Beamforming

Data-Dependent Beamforming

**Data-Dependent Source Separation**

- General Overview

- Notations

- Nongaussianity

- Hard/Soft Clustering

- Independent Component Analysis and Sparse Component Analysis

- Initialization and Constraints (Nongaussian Models)

- Nonstationarity

- Multichannel Wiener Filter (Revised)

- Spatial Image EM (SIEM)

- Alternative Flavors of EM

- Initialization and Constraints (Nonstationary Models)

- Joint Spatial-Spectral Estimation

Summary and Perspectives

# Data-Dependent Source Separation

## General Overview

Rather than using a detection/classification technique to estimate the signal statistics and subsequently deriving the spatial filter, we seek to estimate them jointly.

In a probabilistic framework:

- ▶ build a full generative model of the mixture signal,
- ▶ infer the parameters and the source signals in some probabilistic sense.



# Data-Dependent Source Separation

## General Overview

Theorem (Darmois): two stationary Gaussian sources are not separable from each other.

Two interchangeable modeling paradigms:

- ▶ **nongaussianity** (Comon and Jutten 2010; Makino, Lee, and Sawada 2007; O'Grady, Pearlmutter, and Rickard 2005; Pedersen et al. 2008),
- ▶ **nonstationarity** (Févotte and Cardoso 2005; Pham, Servière, and Boumaraf 2003; Vincent, Bertin, et al. 2014; Vincent, Jafari, et al. 2010).

# Data-Dependent Source Separation

## Notations

Reminder: in the case of  $J$  directional sources

$$Y_n(m, k) = \sum_{j=1}^J X_{nj}(m, k) + V_n(m, k)$$

$Y_n(m, k)$ : recorded at the  $n$ -th microphone  
 $X_{nj}(m, k)$ :  $j$ -th source as received by the  $n$ -th mic  
 $V_n(m, k)$ : diffuse noise

We call  $X_{nj}(m, k)$  the **spatial image** of the  $j$ -th source at the  $n$ -th microphone.

Assuming low reverberation,  $X_{nj}(m, k)$  can be modeled as

$$X_{nj}(m, k) = A_{nj}(k) S_j(m, k)$$

$A_{nj}(k)$ : Fourier transform of the RIR  
 $S_j(m, k)$ : anechoic source signal

# Data-Dependent Source Separation

## Notations

This can also be written in vector form (one entry per microphone):

$$\mathbf{y}(m, k) = \sum_{j=1}^J \mathbf{x}_j(m, k) + \mathbf{v}(m, k)$$

$$\mathbf{x}_j(m, k) = \mathbf{a}_j(k) S_j(m, k)$$

or in matrix form (one entry per microphone and per source):

$$\mathbf{y}(m, k) = \mathbf{A}(k) \mathbf{s}(m, k).$$

We call  $\mathbf{a}_j(k)$  the mixing vectors and  $\mathbf{A}(k)$  the **mixing matrix**.

# Data-Dependent Source Separation

## Nongaussianity

In the time-frequency domain, the distribution of the source STFT coefficients  $S_j(m, k)$  is nongaussian.

More specifically, it is **sparse**: at each frequency, a few coefficients are large and most are close to zero.

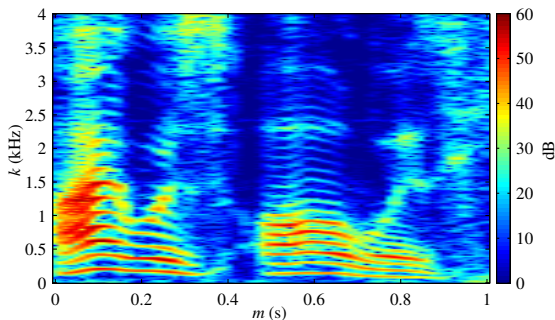


Figure : STFT of a speech source.

# Data-Dependent Source Separation

## Nongaussianity

This property can be modeled in two ways:

- ▶ **binary activation model**

- hard/soft clustering

- ▶ **sparse i.i.d. model**

- independent component analysis and sparse component analysis

# Data-Dependent Source Separation

## Hard/Soft Clustering

Binary activation model:

- ▶ in each time-frequency bin, only one source  $j^{\text{act}}(m, k)$  is active and the other sources are equal to zero,
- ▶  $j^{\text{act}}(m, k)$  is uniformly distributed in  $\{1, \dots, J\}$ ,
- ▶ the noise  $\mathbf{v}(m, k)$  is Gaussian with covariance matrix  $\Phi_{\mathbf{v}}(k)$ .

Note: the source STFT coefficients  $S_j^{\text{act}}(m, k)$  are considered as deterministic parameters.

Goal: jointly estimate the hidden data  $j^{\text{act}}(m, k)$  and the model parameters  $\theta = \{\mathbf{a}_j(k), \Phi_{\mathbf{v}}(k)\}$  from the observed data by maximizing the log-likelihood:

$$\begin{aligned} \mathcal{L} = & \sum_{m,k} -\log \det(\pi \Phi_{\mathbf{v}}(k)) \\ & - (\mathbf{y}(m, k) - \mathbf{a}_j^{\text{act}}(k) S_j^{\text{act}}(m, k))^{\text{H}} \Phi_{\mathbf{v}}^{-1}(k) (\mathbf{y}(m, k) - \mathbf{a}_j^{\text{act}}(k) S_j^{\text{act}}(m, k)) \end{aligned}$$

# Data-Dependent Source Separation

## Hard/Soft Clustering

If  $j = j^{\text{act}}(m, k)$ , then the ML estimate of  $S_j(m, k)$  is given by the MVDR beamformer

$$\check{S}_j(m, k) = \mathbf{h}_{\text{MVDR},j}^{\text{H}}(k) \mathbf{y}(m, k) \quad \text{with} \quad \mathbf{h}_{\text{MVDR},j}(k) = \frac{\Phi_{\mathbf{v}}^{-1}(k) \mathbf{a}_j(k)}{\mathbf{a}_j^{\text{H}}(k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{a}_j(k)}.$$

Note that  $\mathbf{h}_{\text{MVDR},j}(k)$  is computed from  $\mathbf{a}_j(k)$ , not from  $\mathbf{a}_{\text{d},j}(k)$ .

The log-likelihood simplifies to

$$\begin{aligned} \mathcal{L} = \sum_{m,k} & -\log \det(\pi \Phi_{\mathbf{v}}(k)) - \mathbf{y}^{\text{H}}(m, k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{y}(m, k) \\ & + \frac{|\mathbf{a}_j^{\text{act H}}(k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{y}(m, k)|^2}{\mathbf{a}_j^{\text{act H}}(k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{a}_j^{\text{act}}(k)} \end{aligned}$$

# Data-Dependent Source Separation

## Hard/Soft Clustering

Expectation-maximization (EM) algorithm:

► E-step:

$$\begin{aligned}\gamma_j(m, k) &\triangleq p(j^{\text{act}}(m, k) = j | \mathbf{y}, \theta) \\ &\propto e^{\frac{|\mathbf{a}_j^H(k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{y}(m, k)|^2}{\mathbf{a}_j^H(k) \Phi_{\mathbf{v}}^{-1}(k) \mathbf{a}_j(k)}}\end{aligned}$$

► M-step:

$$\begin{aligned}\mathbf{a}_j(k) &= \frac{\sum_{j,m} \gamma_j(m, k) \check{S}_j^*(m, k) \mathbf{y}(m, k)}{\sum_{j,m} \gamma_j(m, k) |\check{S}_j(m, k)|^2} \\ \Phi_{\mathbf{v}}(k) &= \frac{1}{M} \sum_{j,m} \gamma_j(m, k) (\mathbf{y}(m, k) - \mathbf{a}_j(k) \check{S}_j(m, k)) (\mathbf{y}(m, k) - \mathbf{a}_j(k) \check{S}_j(m, k))^H\end{aligned}$$

with  $\check{S}_j(m, k)$  updated as above



# Data-Dependent Source Separation

## Hard/Soft Clustering

After convergence, estimate  $S_j(m, k)$ :

- ▶ either in the ML sense (**hard time-frequency masking**)

$$\hat{S}_j(m, k) = \begin{cases} \check{S}_j(m, k) & \text{if } j = j^{\text{act}}(m, k) \\ 0 & \text{otherwise} \end{cases}$$

- ▶ or in the MMSE sense (**soft time-frequency masking**)

$$\hat{S}_j(m, k) = \gamma_j(m, k) \check{S}_j(m, k)$$

Popular heuristic alternatives: k-means clustering of

- ▶ interchannel phase and intensity differences,
- ▶ phase- and amplitude-normalized  $\mathbf{y}(m, k)$ .

# Data-Dependent Source Separation

## Hard/Soft Clustering

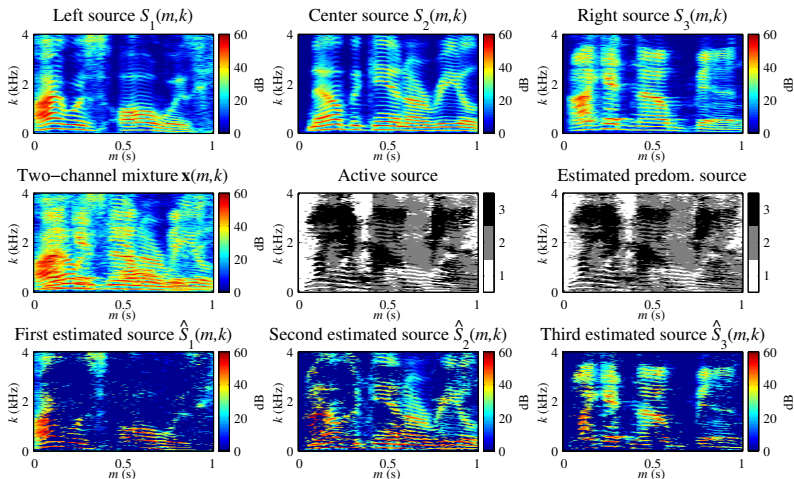


Figure : Hard time-frequency masking ( $\mathbf{a}_j(k)$  known).

# Data-Dependent Source Separation

## Hard/Soft Clustering

Hard vs soft clustering makes little difference in practice.

Both do not fully exploit the benefit of multichannel processing: the beamformer enhances the target but not does attenuate interfering speakers.

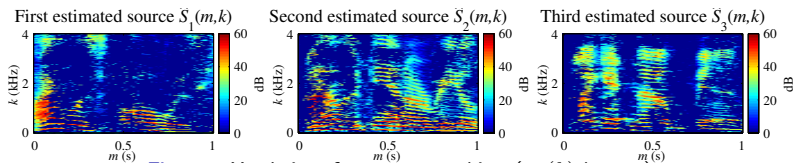


Figure : Hard time-frequency masking ( $\mathbf{a}_j(k)$  known).

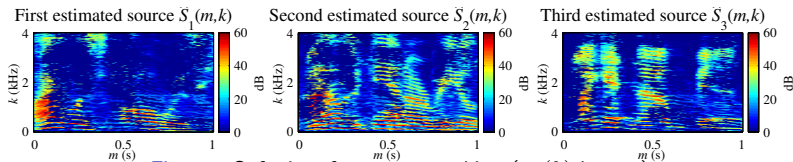


Figure : Soft time-frequency masking ( $\mathbf{a}_j(k)$  known).

# Data-Dependent Source Separation

## Independent Component Analysis and Sparse Component Analysis

To reduce distortion, model  $S_j(m, k)$  as independent and identically distributed (i.i.d.) according to a continuous (circular complex) distribution.

Example model: generalized exponential distribution

$$P(|S_j(m, k)|) = \frac{p}{\beta(k)\Gamma(1/p)} e^{-\left|\frac{S_j(m, k)}{\beta(k)}\right|^p}$$

$p$ : shape parameter  
 $\beta(k)$ : scale parameter

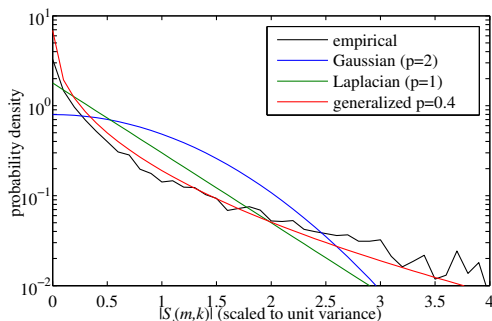


Figure : Distribution of the magnitude STFT coefficients of a speech source.

# Data-Dependent Source Separation

## Independent Component Analysis and Sparse Component Analysis

When there are  $J = N$  directional sources and no noise (determined mixture), the sources are obtained by inverting the mixing matrix:

$$\check{\mathbf{s}}(m, k) = \mathbf{A}^{-1}(k) \mathbf{y}(m, k).$$

Goal: estimate  $\mathbf{A}(k)$  from the observed data by maximizing the log-likelihood:

$$\mathcal{L} = \sum_{j,m,k} \log P(\check{S}_j(m, k))$$

with  $\check{S}_j(m, k)$  depending on  $\mathbf{A}(k)$  and  $\mathbf{y}(m, k)$  as above.

This is equivalent to minimizing the mutual information between the sources.

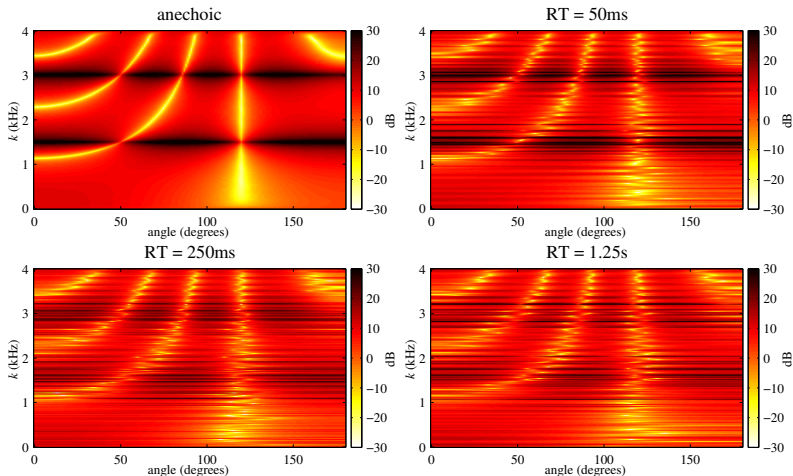
This is called **frequency-domain independent component analysis** (FDICA).

Optimization performed using nonlinear optimization techniques, e.g., gradient ascent. See V. Zarzoso and A. Yeredor's lecture for more details.

# Data-Dependent Source Separation

## Independent Component Analysis and Sparse Component Analysis

FDICA was said to perform **blind beamforming** because it automatically adapts to the deviations of  $\mathbf{a}_j(k)$  from  $\mathbf{a}_{d,j}(k)$  due to echoes and reverberation.



**Figure :** Beam patterns obtained by FDICA as a function of the direct path angle ( $\theta_{\text{target}} = 50^\circ$ ,  $\theta_{\text{interf}} = 120^\circ$ ,  $N = 2$ ,  $d = 30$  cm).

# Data-Dependent Source Separation

## Independent Component Analysis and Sparse Component Analysis

When there are  $J > N$  directional sources or noise (under-determined mixture), joint estimation of  $\mathbf{A}(k)$  and  $\mathbf{s}(m, k)$  in the ML sense is difficult.

Dictionary learning techniques have little been applied so far.

Popular heuristics:

- ▶ first estimate  $\mathbf{A}(k)$  using some clustering technique (all columns) or ICA ( $N$  columns at once),
- ▶ then estimate  $\mathbf{s}(m, k)$  in the ML sense.

This is called **sparse component analysis** (SCA).

For typical values of  $p$ , the resulting  $\mathbf{s}(m, k)$  are nonzero for up to  $N$  sources.

# Data-Dependent Source Separation

## Independent Component Analysis and Sparse Component Analysis

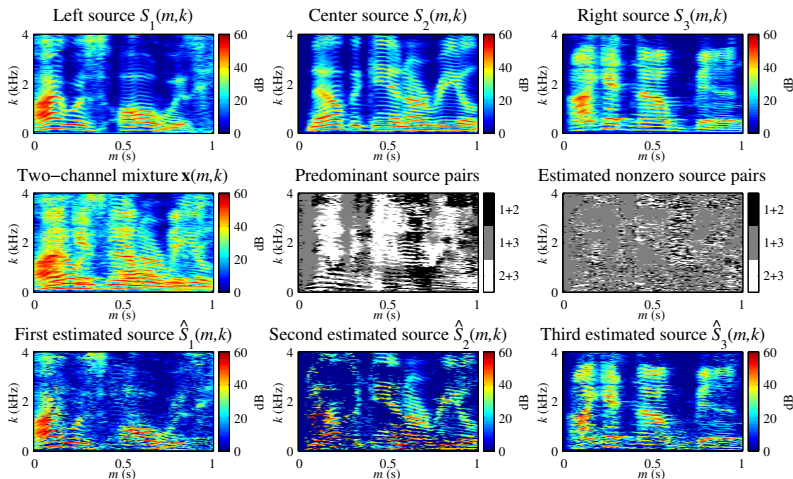


Figure : Sparse component analysis ( $\mathbf{a}_j(k)$  known).



# Data-Dependent Source Separation

## Initialization and Constraints (Nongaussian Models)

Up to now, the mixing vectors  $\mathbf{a}_j(k)$  at different frequencies  $k$  are unrelated with each other.

Consequence: the sources  $S_j(m, k)$  are estimated up to a **permutation indeterminacy**.

Heuristic approaches:

- ▶ initialize  $\mathbf{a}_j(k)$  with the steering vector  $\mathbf{a}_{d_j}(k)$ ,
- ▶ align the permutations by minimizing, e.g.,  $\|\mathbf{a}_j(k) - \mathbf{a}_{d_j}(k)\|$ ,
- ▶ modify the estimation algorithm so as to account for a linear constraint, e.g.,  $\mathbf{a}_{d_j}^H(k)\mathbf{a}_j(k) = 1$ , or a penalty term, e.g.,  $\|\mathbf{a}_j(k) - \mathbf{a}_{d_j}(k)\|^2$ .
- ▶ Note that  $\mathbf{a}_j(k)$  needs to be normalized in the same way as  $\mathbf{a}_{d_j}(k)$ .

# Data-Dependent Source Separation

## Initialization and Constraints (Nongaussian Models)

Note: none of these approaches matches the actual distribution of  $\mathbf{a}_j(k)$  in a reverberant environment. . .

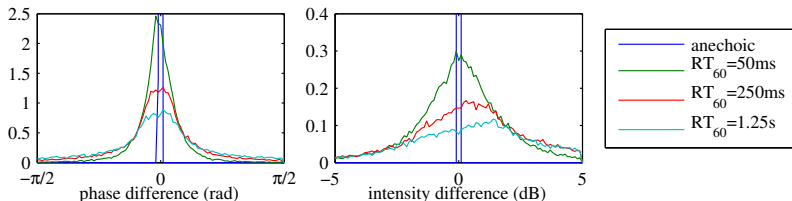


Figure : Distribution of  $A_{2j}(k)/A_{1j}(k)$  over  $k$  for one source  $j$ .

. . . but they work nevertheless to a certain extent!

# Data-Dependent Source Separation

## Initialization and Constraints (Nongaussian Models)

Results ( $N = 2$ ,  $J = 3$ ,  $d = 1$  m, soft time-frequency masking)

Mixture with RT = 130 ms 

Estimated sources   

Mixture with RT = 250 ms 

Estimated sources   

# Data-Dependent Source Separation

## Nonstationarity

Despite their success, FDICA and SCA have two fundamental limitations:

- ▶ model valid for directional sources with low reverberation only,
- ▶ joint estimation difficult when  $J > N$ .

Idea 1: instead of considering the signals emitted by the sources, consider their spatial images  $\mathbf{x}_j(m, k)$ :

$$\mathbf{y}(m, k) = \sum_{j=1}^J \mathbf{x}_j(m, k)$$

Becomes valid for reverberated and diffuse sources.

No need for a specific noise term: noise is just a source (or several sources) as the others.

# Data-Dependent Source Separation

## Nonstationarity

Idea 2: rather than considering a sparse i.i.d. model use a simpler (circular complex) model but with time-varying parameters.

For a wide class of distributions, time-varying parameters result in sparse data.

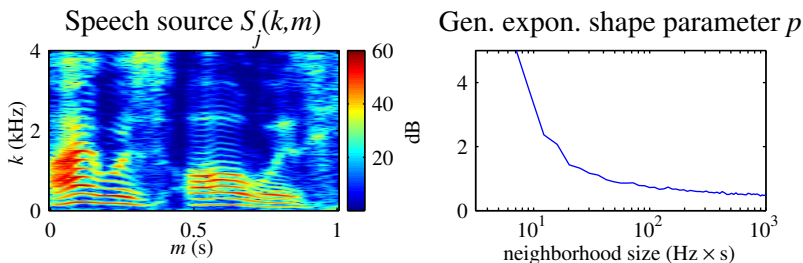
Better matches the physical production process of speech and other sounds.

Will make it easy to exploit spectral models in addition to spatial models.

# Data-Dependent Source Separation

## Nonstationarity

The non-sparsity of source STFT coefficients over small time-frequency regions suggests the use of a non-sparse distribution.



**Figure :** Empirical distribution of STFT coefficients over time-frequency regions of increasing size.

# Data-Dependent Source Separation

## Nonstationarity

Besides the generalized exponential, several other distributions have been proposed for the source magnitude/power STFT coefficients that do not easily generalize to multichannel data.

These distributions are generally equivalently expressed as **divergences**:

- ▶ Poisson  $\leftrightarrow$  Kullback-Leibler divergence aka I-divergence
- ▶ nonzero-mean tied-variance Gaussian  $\leftrightarrow$  squared Euclidean distance

# Data-Dependent Source Separation

## Multichannel Wiener Filter (Revised)

**Local Gaussian model (LGM):**

$$\mathbf{x}_j(m, k) \sim \mathcal{N}(\mathbf{0}, \Phi_{\mathbf{x}_j}(m, k))$$

$$\Rightarrow \mathbf{y}(m, k) \sim \mathcal{N}(\mathbf{0}, \Phi_{\mathbf{y}}(m, k)) \quad \text{with} \quad \Phi_{\mathbf{y}}(m, k) = \sum_{j=1}^J \Phi_{\mathbf{x}_j}(m, k)$$

Factorization into a **time-varying power spectrum**  $\phi_{S_j}(m, k)$  and a **spatial covariance matrix**  $\Phi_{\mathbf{a}_j}(k)$

$$\Phi_{\mathbf{x}_j}(m, k) = \phi_{S_j}(m, k) \Phi_{\mathbf{a}_j}(k)$$

Goal: jointly estimate the model parameters  $\theta = \{\phi_{S_j}(m, k), \Phi_{\mathbf{a}_j}(k)\}$  from the observed data by maximizing the log-likelihood:

$$\mathcal{L} = \sum_{m, k} -\log \det(\pi \Phi_{\mathbf{y}}(m, k)) - \mathbf{y}^H(m, k) \Phi_{\mathbf{y}}^{-1}(m, k) \mathbf{y}(m, k)$$

Note:  $S_j(m, k)$  are now considered as random variables and do not appear in  $\mathcal{L}$  anymore.



# Data-Dependent Source Separation

## Multichannel Wiener Filter (Revised)

Generalization: replace  $\mathbf{y}(m, k)$  by the **empirical mixture covariance matrix**

$$\hat{\Phi}_{\mathbf{y}}(m, k) = \mathbb{E}\{\mathbf{y}(m, k)\mathbf{y}^H(m, k)\}:$$

$$\begin{aligned}\hat{\mathcal{L}} &\triangleq \sum_{m,k} -\log \det(\pi \Phi_{\mathbf{y}}(m, k)) - \mathbb{E}\{\mathbf{y}^H(m, k) \Phi_{\mathbf{y}}^{-1}(m, k) \mathbf{y}(m, k)\} \\ &= \sum_{m,k} -\log \det(\pi \Phi_{\mathbf{y}}(m, k)) - \text{tr}(\mathbb{E}\{\mathbf{y}^H(m, k) \Phi_{\mathbf{y}}^{-1}(m, k) \mathbf{y}(m, k)\}) \\ &= \sum_{m,k} -\log \det(\pi \Phi_{\mathbf{y}}(m, k)) - \text{tr}(\mathbb{E}\{\Phi_{\mathbf{y}}^{-1}(m, k) \mathbf{y}(m, k) \mathbf{y}^H(m, k)\}) \\ &= \sum_{m,k} -\log \det(\pi \Phi_{\mathbf{y}}(m, k)) - \text{tr}(\Phi_{\mathbf{y}}^{-1}(m, k) \hat{\Phi}_{\mathbf{y}}(m, k)).\end{aligned}$$

Computed by averaging of  $\mathbf{y}(m, k)\mathbf{y}^H(m, k)$  locally over time and/or frequency.

Besides the observed phase and intensity differences,  $\hat{\Phi}_{\mathbf{y}}(m, k)$  also accounts for the correlation aka **coherence** between microphones.

# Data-Dependent Source Separation

## Multichannel Wiener Filter (Revised)

Coherence reduces indeterminacies and helps recovering up to  $N^2$  sources.

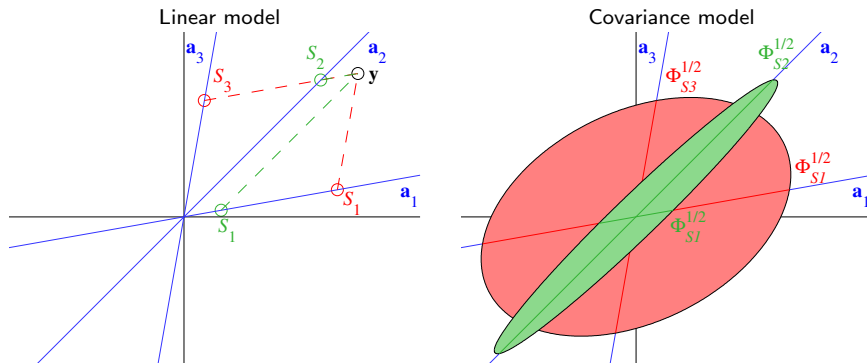


Figure : Use of  $y(m, k)$  vs  $\hat{\Phi}_y(m, k)$  as the input representation.

# Data-Dependent Source Separation

## Multichannel Wiener Filter (Revised)

Given the model parameters, estimate the source spatial images in the MMSE sense using the **multichannel Wiener filter**

$$\hat{\mathbf{x}}_j(m, k) = \mathbf{H}_j^H(m, k) \mathbf{y}(m, k).$$

Taking the derivative of  $E\{\|\mathbf{H}_j^H(m, k) \mathbf{y}(m, k) - \mathbf{x}_j(m, k)\|^2\}$  w.r.t.  $\mathbf{H}_j^H(m, k)$  and equating it to zero yields

$$\begin{aligned} \mathbf{H}_{Wj}(m, k) &= \Phi_{\mathbf{y}}^{-1}(m, k) \Phi_{\mathbf{x}_j}(m, k) \\ &= \left( \sum_{j'=1}^J \phi_{S_{j'}}(m, k) \Phi_{\mathbf{a}_{j'}}(k) \right)^{-1} \phi_{S_j}(m, k) \Phi_{\mathbf{a}_j}(k). \end{aligned}$$

The tradeoff between interference and noise reduction and target distortion can be controlled similarly to above.

# Data-Dependent Source Separation

## Spatial Image EM (SIEM)

Expectation-maximization (EM) algorithm with  $\{\mathbf{x}_j(m, k)\}$  as hidden data:

- ▶ E-step: compute  $p(\mathbf{x}_j(m, k)|\mathbf{y}, \theta)$

$$\mathbf{H}_{Wj}(m, k) = \Phi_{\mathbf{y}}^{-1}(m, k) \Phi_{\mathbf{x}_j}(m, k)$$

$$p(\mathbf{x}_j(m, k)|\mathbf{y}, \theta) = \mathcal{N}(\underbrace{\mathbf{H}_{Wj}^H(m, k) \mathbf{y}(m, k)}_{\text{posterior mean}}, \underbrace{(\mathbf{I} - \mathbf{H}_{Wj}^H(m, k)) \Phi_{\mathbf{x}_j}(m, k)}_{\text{posterior covariance}})$$

with  $\mathbf{I}$  the identity matrix of size  $N \times N$

$$\Rightarrow \hat{\Phi}_{\mathbf{x}_j}(m, k) = \underbrace{\mathbf{H}_{Wj}^H(m, k) \hat{\Phi}_{\mathbf{y}}(m, k) \mathbf{H}_{Wj}(m, k) + (\mathbf{I} - \mathbf{H}_{Wj}^H(m, k)) \Phi_{\mathbf{x}_j}(m, k)}_{\text{posterior second order moment}}$$

- ▶ M-step:  $\max_{\theta} E_{\mathbf{x}}\{\log p(\mathbf{y}, \mathbf{x})\} \Leftrightarrow$   
 $\max_{\theta} \sum_{j,m,k} -\log \det(\pi \Phi_{\mathbf{x}_j}(m, k)) - \text{tr}(\Phi_{\mathbf{x}_j}^{-1}(m, k) \hat{\Phi}_{\mathbf{x}_j}(m, k))$

$$\phi_{S_j}(m, k) = \frac{1}{N} \text{tr}(\Phi_{\mathbf{a}_j}^{-1}(k) \hat{\Phi}_{\mathbf{x}_j}(m, k))$$

$$\Phi_{\mathbf{a}_j}(k) = \frac{1}{M} \sum_{m=1}^M \frac{\hat{\Phi}_{\mathbf{x}_j}(m, k)}{\phi_{S_j}(m, k)}$$

# Data-Dependent Source Separation

## Spatial Image EM (SIEM)

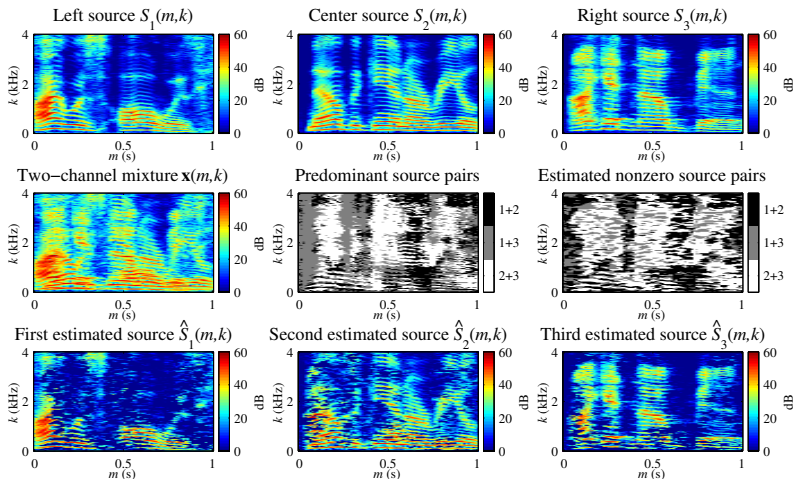


Figure : Spatial image EM ( $\mathbf{a}_j(k)$  known).

# Data-Dependent Source Separation

## Alternative Flavors of EM

Alternative flavors of EM depending on the choice of the hidden data:

- ▶ “subsources”,
- ▶ binary activations.

Also, more general techniques beyond EM not covered here:

- ▶ other auxiliary function for the log-likelihood  
→ minorization-maximization (MM)
- ▶ free energy,  
→ variational Bayes (VB), see A. Deleforge's lecture for more details

# Data-Dependent Source Separation

## Alternative Flavors of EM

Subsource EM (SSEM): decompose (non-uniquely)  $\Phi_{\mathbf{a}_j}(k) = \mathbf{A}_j(k)\mathbf{A}_j^H(k)$  and  $\mathbf{s}_j(k) \triangleq \mathbf{A}_j^{-1}(k)\mathbf{x}_j(m, k)$  with  $\mathbf{A}_j(k) \in \mathbb{C}^{N \times R_j}$  and  $\mathbf{s}_j(k) \in \mathbb{C}^{R_j}$ , stack into  $\mathbf{A}(k) \in \mathbb{C}^{N \times R}$  and  $\mathbf{s}(k) \in \mathbb{C}^R$ , and consider  $\mathbf{s}(k) \in \mathbb{C}^R$  as hidden data.

- E-step:

$$\Phi_{\mathbf{s}}(m, k) = \text{diag}(\overbrace{\phi_{S_j}(m, k)}^{R_j \text{ times}})$$

$$\Phi_{\mathbf{y}}(m, k) = \mathbf{A}(k)\Phi_{\mathbf{s}}(m, k)\mathbf{A}^H(k) + \Phi_{\mathbf{v}}(k)$$

$$\mathbf{H}_W(m, k) = \Phi_{\mathbf{y}}^{-1}(m, k)\mathbf{A}(k)\Phi_{\mathbf{s}}(m, k)$$

$$\hat{\Phi}_{\mathbf{s}}(m, k) = \mathbf{H}_W^H(m, k)\hat{\Phi}_{\mathbf{y}}(m, k)\mathbf{H}_W(m, k) + (\mathbf{I} - \mathbf{H}_W^H(m, k)\mathbf{A}(k))\Phi_{\mathbf{s}}(m, k)$$

with  $\mathbf{I}$  the identity matrix of size  $R \times R$ .

- M-step:

$$\phi_{S_j}(m, k) = \frac{1}{R_j} \text{tr}(\hat{\Phi}_{\mathbf{s}_j}(m, k))$$

$$\mathbf{A}(k) = \left( \sum_{m=1}^M \hat{\Phi}_{\mathbf{y}}(m, k)\mathbf{H}(m, k) \right) \left( \sum_{m=1}^M \hat{\Phi}_{\mathbf{s}}(m, k) \right)^{-1}$$

# Data-Dependent Source Separation

## Alternative Flavors of EM

Binary activation EM (BAEM): assume a single active source  $j^{\text{act}}(m, k)$  uniformly distributed in  $\{1, \dots, J\}$  and consider its index as hidden data.

► E-step:

$$\begin{aligned}\gamma_j(m, k) &\triangleq p(j^{\text{act}}(m, k) = j | \mathbf{y}, \theta) \\ &\propto \frac{e^{-\text{tr}(\mathbf{\Phi}_{\mathbf{x}_j}^{-1}(m, k) \hat{\mathbf{\Phi}}_{\mathbf{y}}(m, k))}}{\det(\pi \mathbf{\Phi}_{\mathbf{x}_j}(m, k))}\end{aligned}$$

► M-step:

$$\begin{aligned}\phi_{S_j}(m, k) &= \frac{1}{N} \text{tr}(\mathbf{\Phi}_{\mathbf{a}_j}^{-1}(k) \hat{\mathbf{\Phi}}_{\mathbf{x}_j}(m, k)) \\ \mathbf{\Phi}_{\mathbf{a}_j}(k) &= \frac{\sum_{m=1}^M \gamma_j(m, k) \hat{\mathbf{\Phi}}_{\mathbf{y}}(m, k) / \phi_{S_j}(m, k)}{\sum_{m=1}^M \gamma_j(m, k)}\end{aligned}$$



# Data-Dependent Source Separation

## Initialization and Constraints (Nonstationary Models)

Up to now, the spatial covariance matrices  $\Phi_{\mathbf{a}_j}(k)$  at different frequencies  $k$  are unrelated with each other.

Consequence: permutation indeterminacy again.

Reminder: on average over all absolute positions in the room

$$\mathbb{E}\{\Phi_{\mathbf{a}_j}(k)\} = \mathbf{a}_{dj}(k)\mathbf{a}_{dj}^H(k) + \phi_r(k)\mathbf{\Gamma}(k)$$

with

$$[\mathbf{\Gamma}(k)]_{nn'} = \text{sinc}\left(\frac{2\pi F_s k d_{nn'}}{c K}\right).$$

Heuristic approaches:

- ▶ initialize  $\Phi_{\mathbf{a}_j}(k)$  with its average  $\mathbb{E}\{\Phi_{\mathbf{a}_j}(k)\}$ ,
- ▶ align the permutations by minimizing, e.g.,  $\|\Phi_{\mathbf{a}_j}(k) - \mathbb{E}\{\Phi_{\mathbf{a}_j}(k)\}\|$

# Data-Dependent Source Separation

## Initialization and Constraints (Nonstationary Models)

More principled approach: modify the estimation algorithm so as to account for

- ▶ either a set of deterministic rank-1 constraints

$$\Phi_{\mathbf{a}_j}(k) = \sum_{\theta} \phi_{\theta_j} \mathbf{a}_d(k, \theta) \mathbf{a}_d^H(k, \theta)$$

- ▶ or a probabilistic inverse-Wishart prior

$$\Phi_{\mathbf{a}_j}(k) \sim \mathcal{IW}(\Psi_j(k), f)$$

where  $\Psi_j(k) = (f - N)E\{\Phi_{\mathbf{a}_j}(k)\}$  and the number of degrees of freedom  $f$  is learned from data

→ M-step modified for maximum a posteriori (MAP) estimation as

$$\Phi_{\mathbf{a}_j}(k) = \frac{1}{\gamma(f + N) + M} \left( \gamma \Psi_j(k) + \sum_{m=1}^M \frac{\hat{\Phi}_{\mathbf{x}_j}(m, k)}{\phi_{S_j}(m, k)} \right)$$

with  $\gamma$  a tradeoff hyper-parameter determining the strength of the prior.

# Data-Dependent Source Separation

## Initialization and Constraints (Nonstationary Models)

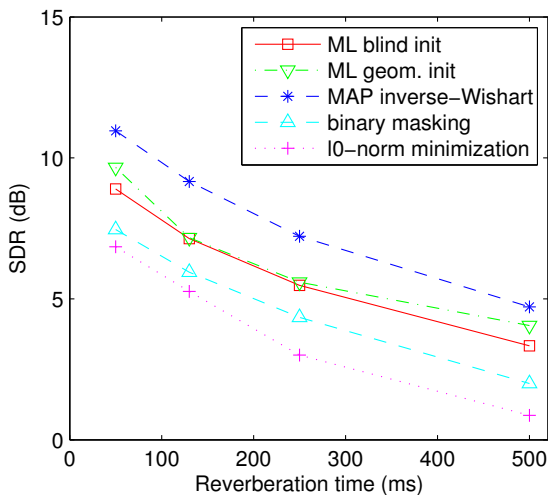


Figure : Separation of 3 speech sources from 2 mics spaced by 5 cm

# Data-Dependent Source Separation

## Joint Spatial-Spectral Estimation

The nonstationary Gaussian model makes it easy to exploit spectral models:

- ▶ nonnegative matrix factorization (NMF) and its constrained variants, e.g., harmonic NMF,
- ▶ spectral/temporal continuity models (HPSS, KAM),
- ▶ deep neural networks (DNN) ...

These models provide deterministic constraints on  $\phi_{S_j}(m, k)$ .

The M-step is simply modified by

- ▶ first estimating  $\phi_{S_j}(m, k)$  as  $1/N \times \text{tr}(\Phi_{\mathbf{a}_j}^{-1}(k) \hat{\Phi}_{\mathbf{x}_j}(m, k))$ ,
- ▶ subsequently projecting it to the constrained space.





It can be shown that this approach does maximize the log-likelihood under the constraint.

# Data-Dependent Source Separation

## Joint Spatial-Spectral Estimation

Results ( $N = 2$ ,  $J = 4$ ,  $d = 1$  m, RT = 250 ms)

Mixture 

Estimated sources using rank-1 spatial covariance    

full-rank spatial covariance    

rank-1 + harmonic NMF    

full-rank + harmonic NMF    

# Outline

Introduction

Fundamental Acoustics

Signal Models

Fundamental Array Processing

Data-Independent Beamforming

Data-Dependent Beamforming

Data-Dependent Source Separation

**Summary and Perspectives**

Wrap-up

Resources

Current Challenges and Opportunities

Acknowledgment

# Summary and Perspectives

## Wrap-up

We have seen many speech enhancement and source separation techniques.

Historically: targeted different use cases in terms of

- ▶ number of microphones,
- ▶ number of speech sources,
- ▶ number of directional and diffuse noise sources.

Today:

- ▶ use cases have merged,
- ▶ rely on the same fundamental principles of acoustics and array processing,
- ▶ share the same signal models,
- ▶ share some estimation criteria.

# Summary and Perspectives

## Wrap-up

The only remaining differences are perhaps

- ▶ whether sensor noise is generally considered or not,
- ▶ whether spatial and spectral models are exploited successively or jointly,
- ▶ whether independence is at the core of the estimation criteria or not.
- ▶ whether the signal statistics and the spatial filter are estimated successively or alternatively,
- ▶ whether estimation is generally done online or not.

But even these differences tend to disappear!



# Summary and Perspectives

## Resources

For development, simulate data by mixing speech with RIRs and noise

- ▶ recorded in a real room

Table : Some impulse response datasets.

Name	# RIRs	N	# rooms	# array pos.	J	moving	real noise
RWCP <sup>1</sup>	364	84	7	1	9	no	no
SiSEC <sup>2</sup>	~50	2	5	1	~20	no	no
AIR <sup>3</sup>	214	2	8	1	13	no	no
CAMIL <sup>4</sup>	32400	2	1	16200	1	yes	no
CHiME2 <sup>5</sup>	242	2	1	1	121	yes	yes

- ▶ or simulated by software<sup>6789</sup>.

<sup>1</sup><http://research.nii.ac.jp/src/en/RWCP-SSD.html>

<sup>2</sup><https://sisec.inria.fr/>

<sup>3</sup><http://www.ind.rwth-aachen.de/de/forschung/tools-downloads/aachen-impulse-response-database/>

<sup>4</sup><https://team.inria.fr/perception/category/data/>

<sup>5</sup>[http://spandh.dcs.shef.ac.uk/chime\\_challenge/chime2013/](http://spandh.dcs.shef.ac.uk/chime_challenge/chime2013/)

<sup>6</sup><http://sourceforge.net/projects/roomsim/>

<sup>7</sup><http://www.audiolabs-erlangen.de/fau/professor/habets/software/{rir-generator,smir-generator}>

<sup>8</sup><http://www.loria.fr/~evincent/Roomsimove.zip>

<sup>9</sup><http://www.audiolabs-erlangen.de/fau/professor/habets/software/noise-generators>

# Summary and Perspectives

## Resources

For test, use real data with a reference (close-talk microphone).

**Table :** Some real multichannel audio datasets with a reference (Le Roux et al. 2015).

Name	applic.	# hours	N	# envs.	speak. pos.	speak. overl.
Aurora-3 <sup>10</sup>	car	~20	4	1	static	no
AMI <sup>11</sup>	meeting	100	16	3	static	yes
DICIT <sup>12</sup>	TV order	6	16	1	moving	no
COSINE <sup>13</sup>	discuss.	38	20	8	moving	yes
SWC <sup>14</sup>	game	7	92	1	moving	yes
CHiME3 <sup>15</sup>	tablet	19	6	4	moving	little

For more datasets, see wiki of ISCA Robust Speech Processing SIG<sup>16</sup>.

<sup>10</sup>[http://catalog.elra.info/index.php?cPath=37\\_40](http://catalog.elra.info/index.php?cPath=37_40)

<sup>11</sup><http://groups.inf.ed.ac.uk/ami/>

<sup>12</sup><http://shine.fbk.eu/resources/dicit-acoustic-woz-data>

<sup>13</sup><http://melodi.ee.washington.edu/cosine/>

<sup>14</sup><http://mini.dcs.shef.ac.uk/data-2/>

<sup>15</sup>[http://spandh.dcs.shef.ac.uk/chime\\_challenge/](http://spandh.dcs.shef.ac.uk/chime_challenge/)

<sup>16</sup><https://wiki.inria.fr/rosp/>

# Summary and Perspectives

## Resources

Get inspiration from or compare with existing software. . .

**Table :** Some multichannel speech enhancement and separation software.

Name	Implemented techniques
BeamformIt <sup>17</sup>	DS beamformer
DSRtk <sup>18</sup>	maximum negentropy beamformer post-filter
HARK <sup>19</sup>	DS or LCMV beamformer FDICA with linear constraint post-filter
MESSL <sup>20</sup>	soft clustering
ManyEars <sup>21</sup>	LGM with linear constraint post-filter
FASST <sup>22</sup>	multichannel NMF and constrained variants

For more software, see wiki of ISCA RoSP SIG or LVA Central<sup>23</sup>.

---

<sup>17</sup><http://www.xavieranguera.com/beamformit/>

<sup>18</sup><http://distantpeechrecognition.sourceforge.net/>

<sup>19</sup><http://www.hark.jp/>

<sup>20</sup><https://github.com/mim/messl>

<sup>21</sup><http://sourceforge.net/projects/manyyears/>

<sup>22</sup><http://bass-db.gforge.inria.fr/fasst/>

<sup>23</sup><http://lvacentral.inria.fr/>

Evaluate the results using

- ▶ subjective listening tests (e.g., MUSHRA or ABX)
- ▶ objective quality metrics

Table : Evaluation software.

Name	Implemented metrics
PESQ <sup>24</sup>	perceptual speech quality (PESQ)
PEMO-Q <sup>25</sup>	perceptual similarity metric (PSM)
Loizou's <sup>26</sup>	segmental SNR log-likelihood ratio cepstrum distance composite measure. . .
BSS Eval <sup>27</sup>	signal-to-distortion ratio (SDR) signal-to-interference ratio (SIR) signal-to-artifacts ratio (SAR)
PEASS <sup>28</sup>	overall perceptual score (OPS) target-related perceptual score (TPS) interference-related perceptual score (IPS) artifacts-related perceptual score (APS)

<sup>24</sup><http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=5374>

<sup>25</sup>[http://www.hoertech.de/web\\_en/produkte/pemo-q.shtml](http://www.hoertech.de/web_en/produkte/pemo-q.shtml)

<sup>26</sup><http://www.crcpress.com/product/isbn/9781466504219>

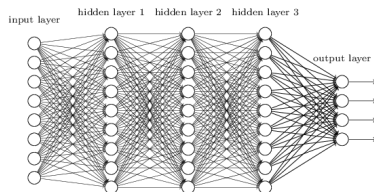
<sup>27</sup>[http://bass-db.gforge.inria.fr/bss\\_eval/](http://bass-db.gforge.inria.fr/bss_eval/)

<sup>28</sup><http://bass-db.gforge.inria.fr/peass/>

# Summary and Perspectives

## Current Challenges and Opportunities

Improved signal models.



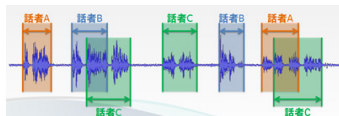
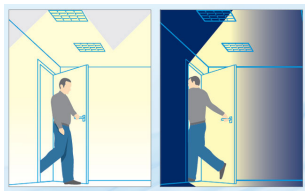
Challenges:

- ▶ account for inter-frame and inter-frequency characteristics,
- ▶ model the phase of the source signals and use this model in a multichannel scenario,
- ▶ leverage expertise in signal processing to exploit recent advances in machine learning, e.g., deep learning, optimally.

# Summary and Perspectives

## Current Challenges and Opportunities

Time-varying acoustic scenes.



Challenges:

- ▶ number of sources changing over time,
- ▶ find which sources appeared/disappeared,
- ▶ sources not continuously active.

# Summary and Perspectives

## Current Challenges and Opportunities

Source and microphone movements.



Challenges:

- ▶ track the sources while moving,
- ▶ jointly estimate the source locations and the signal model parameters,
- ▶ whenever possible, control the movement.

Opportunities:

- ▶ avoid location indeterminacies for linear/planar arrays,
- ▶ increase SNR by moving closer to the speakers.

# Summary and Perspectives

## Current Challenges and Opportunities

Learning of the manifold of acoustic responses specific to a given room.



### Challenges:

- ▶ estimation of some acoustic responses in the first place,
- ▶ dimension reduction,
- ▶ robustness to change of temperature, position of furniture and people. . .
- ▶ new approaches for source separation as a model selection problem

Opportunity: account for all possibly available spatial information: direct path, delays and amplitudes of early echoes, shape of reverberation.



# Summary and Perspectives

## Current Challenges and Opportunities

Ad-hoc arrays built from separate devices available at a given time.



### Challenges:

- ▶ time-varying delay, sampling frequency mismatch, microphone mismatch,
- ▶ computational constraints,
- ▶ distributed estimation.

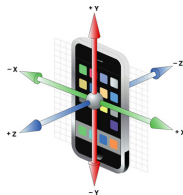
### Opportunities:

- ▶ use all available microphones,
- ▶ wider spatial coverage.

# Summary and Perspectives

## Current Challenges and Opportunities

Multimodal integration.



Challenges:

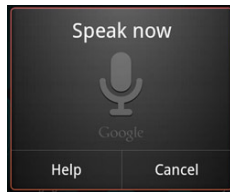
- ▶ integrate with cameras, accelerometers, lasers. . .
- ▶ heterogeneous data with different sampling rates.

Opportunity: exploit each modality for what it works best (e.g., vision for source localization).

# Summary and Perspectives

## Current Challenges and Opportunities

Integration with subsequent applications.



Challenges:

- ▶ use for remixing or automatic speech recognition,
- ▶ shape residual noise and speech distortion so that they are as little disturbing as possible for the considered task,
- ▶ characterize the uncertainty in the estimated source signals and propagate it to the considered task.

Opportunity: better integration will improve performance for the considered task.

# Summary and Perspectives

## Acknowledgment

Part of this lecture follows the article “Multi-microphone speech enhancement and source separation” co-authored with Sharon Gannot, Shmulik Markovich-Golan, and Alexey Ozerov.

Many thanks to them!

## References I

- Benesty, J., J. Chen, and Y. Huang (2008). **Microphone Array Signal Processing**. Berlin, Germany: Springer-Verlag.
- Comon, P. and C. Jutten, eds. (2010). **Handbook of Blind Source Separation, Independent Component Analysis and Applications**. Academic Press.
- Crochiere, R. E. and L. R. Rabiner (1983). **Multirate Digital Signal Processing**. Englewood Cliffs, New Jersey, USA: Prentice-Hall.
- Doclo, S. and M. Moonen (Dec. 2003). "Design of far-field and near-field broadband beamformers using eigenfilters". In: **Signal Processing** 83.12, pp. 2641–2673.
- Er, Meng and A. Cantoni (Dec. 1983). "Derivative constraints for broad-band element space antenna array processors". In: **IEEE Trans. Acoust., Speech, Signal Process.** 31.6, pp. 1378–1393. ISSN: 0096–3518.
- Févoite, C. and J.-F. Cardoso (2005). "Maximum likelihood approach for blind audio source separation using time-frequency Gaussian models". In: **Proc. 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)**, pp. 78–81.
- Gannot, S., D. Burshtein, and E. Weinstein (Aug. 2001). "Signal enhancement using beamforming and nonstationarity with applications to speech". In: **IEEE Trans. Signal Process.** 49.8, pp. 1614–1626.

## References II

- Le Roux, J. et al. (2015). “MICbots: collecting large realistic datasets for speech and audio research using mobile robots”. In: **Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 5635–5639.
- Makino, S., T.-W. Lee, and H. Sawada (2007). **Blind Speech Separation**. Springer.
- O’Grady, P., B. Pearlmutter, and S. T. Rickard (2005). “Survey of Sparse and Non-Sparse Methods in Source Separation”. In: **Int. J. Imaging Syst. Tech.** 15.1, pp. 18–33.
- Pedersen, M.S. et al. (2008). “Convolutional Blind Source Separation Methods”. In: **Springer Handbook of Speech Processing**. Springer, pp. 1065–1094.
- Pham, D.-T., C. Servière, and H. Boumaraf (2003). “Blind separation of speech mixtures based on nonstationarity”. In: **Proc. 7th Intl. Symp. on Signal Processing and its Applications (ISSPA)**, pp. 73–76.
- Taseska, M. and E.A.P. Habets (July 2014). “Informed spatial filtering for sound extraction using distributed microphone arrays”. In: **IEEE/ACM Trans. Acoust., Speech, Signal Process.** 22.7.
- Thiergart, O., M. Taseska, and E.A.P. Habets (Dec. 2014). “An informed parametric spatial filter based on instantaneous direction-of-arrival estimate”. In: **IEEE/ACM Trans. Acoust., Speech, Signal Process.** 22.12.

## References III

- Vincent, E., N. Bertin, et al. (2014). “From blind to guided audio source separation: How models and side information can improve the separation of sound”. In: **IEEE Signal Process. Mag.** 31.3, pp. 107–115.
- Vincent, E., M.G. Jafari, et al. (2010). “Probabilistic modeling paradigms for audio source separation”. In: **Machine Audition: Principles, Algorithms and Systems**. IGI Global, pp. 162–185.
- Wexler, J. and S. Raz (Nov. 1990). “Discrete Gabor expansions”. In: **Signal Processing** 21.3, pp. 207–220.