Digital Audio Signal Processing

DASP

Chapter-4:
Adaptive Beamforming & Multi-Channel Noise Reduction

Marc Moonen
Dept. E.E./ESAT-STADIUS, KU Leuven
marc.moonen@esat.kuleuven.be
homes.esat.kuleuven.be/~moonen/

1

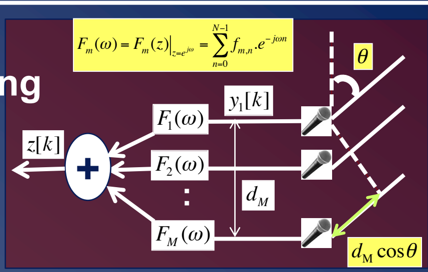
Overview

- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction** (/speech enhancement)
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=spectral+spatial filtering)

2

Recap 1/4

Filter-and-sum Beamforming



Classification:

Fixed beamforming:

Data-independent, fixed filters F_m = Previous chapter

e.g. matched filtering, superdirective, …

Adaptive beamforming:

Data-dependent filters F_m = This chapter

e.g. LCMV-beamformer, generalized sidelobe canceler

Recap 2/4

Data model: source signal

- Microphone signals are filtered versions of source signal ***S(ω)*** at angle θ

$$Y_m(\omega,\theta) = \overbrace{H_m(\omega,\theta)}^{\text{dir. pattern}}.\overbrace{e^{-j\omega\tau_m(\theta)}}^{\text{pos.-dep. phase shift}}.S(\omega)$$

- Stack all microphone signals (m=1..M) in a vector

$$\mathbf{Y}(\omega,\theta) = \mathbf{d}(\omega,\theta).S(\omega)$$

$$\mathbf{d}(\omega,\theta) = \left[H_1(\omega,\theta).e^{-j\omega\tau_1(\theta)} \quad \dots \quad H_M(\omega,\theta).e^{-j\omega\tau_M(\theta)}\right]^T$$

d is **`steering vector'**

Recap 3/4

Data model: source signal + noise

- Microphone signals are also corrupted by additive noise

$$\mathbf{Y}(\omega,\theta) = \mathbf{d}(\omega,\theta).S(\omega) + \mathbf{N}(\omega)$$

$$\mathbf{N}(\omega) = \left[N_1(\omega) \quad N_2(\omega) \quad \ldots \quad N_M(\omega)\right]^T$$

Noise correlation matrix is

$$\mathbf{\Phi}_{noise}(\omega) = E\{\mathbf{N}(\omega).\mathbf{N}(\omega)^H\}$$

- Output signal after `filter-and-sum' is

$$Z(\omega,\theta) = \sum_{m=1}^{M} F_m^*(\omega).Y_m(\omega,\theta) = \mathbf{F}^H(\omega).\mathbf{Y}(\omega,\theta) = \{\mathbf{F}^H(\omega).\mathbf{d}(\omega,\theta)\}.S(\omega) + \mathbf{F}^H(\omega).\mathbf{N}(\omega,\theta)$$

Recap 4/4

Definitions:

- Array directivity pattern = `transfer function' for source signal at θ

$$H(\omega,\theta) = \mathbf{F}^H(\omega).\mathbf{d}(\omega,\theta)$$

- Array Gain = improvement in SNR for source signal at θ

$$G(\omega,\theta) = \frac{SNR_{output}}{SNR_{input}} = \frac{\left|\mathbf{F}^H(\omega).\mathbf{d}(\omega,\theta)\right|^2}{\mathbf{F}^H(\omega).\mathbf{\Gamma}_{noise}(\omega).\mathbf{F}(\omega)}$$

White Noise Gain

=array gain for **spatially uncorrelated** noise $\Gamma_{noise}^{white} = I$

Directivity

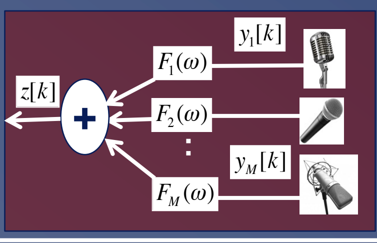
=array gain for **diffuse** noise $\Gamma_{ij}^{diffuse}(\omega) = \text{sinc}\left(\frac{\omega f_s (d_j - d_i)}{c}\right)$

Overview

- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction** (/speech enhancement)
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=spectral+spatial filtering)

Adaptive beamforming Goal & Set-up

- **Adaptive filter-and-sum structure:**
  - Aim is to minimize noise output power, while maintaining a fixed response for a given angle **ψ** (=assumed source signal angle) (plus possibly other linear constraints, see below)
  - This is similar to the operation of a matched filter beamformer (in white noise) or superdirective beamformer (in diffuse noise) see Chapter-3 (**) p.26 & 36
    but now noise field is unknown & adapted to



Adaptive beamforming Goal & Set-up

- **Adaptive filter-and-sum structure:**
  - Will use (adaptive) FIR filters :

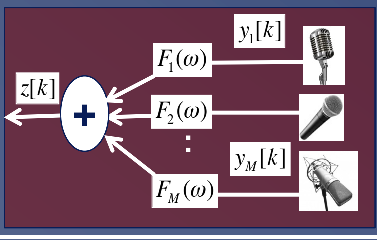
$$z[k] = \mathbf{f}^T \mathbf{y}[k] = \sum_{m=1}^{M} \mathbf{f}_m^T \mathbf{y}_m[k]$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1^T & \mathbf{f}_2^T & \dots & \mathbf{f}_M^T \end{bmatrix}^T$$

$$\mathbf{f}_m = \begin{bmatrix} f_{m,0} & f_{m,1} & \dots & f_{m,N-1} \end{bmatrix}^T$$

$$\mathbf{y}[k] = \begin{bmatrix} \mathbf{y}_1^T[k] & \mathbf{y}_2^T[k] & \dots & \mathbf{y}_M^T[k] \end{bmatrix}^T$$

$$\mathbf{y}_m[k] = \begin{bmatrix} y_m[k] & y_m[k-1] & \dots & y_m[k-N+1] \end{bmatrix}^T$$



Overview

- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction** (/speech enhancement)
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=spectral+spatial filtering)

LCMV beamformer

LCMV = Linearly Constrained Minimum Variance

- Minimize (source signal+noise) output power (='variance') $z[k]$ (see 'Note') :

$$\min_{\mathbf{f}} E\{z^2[k]\} = \min_{\mathbf{f}} \mathbf{f}^T \cdot \mathbf{R}_{yy}[k] \cdot \mathbf{f} \qquad \mathbf{R}_{yy}[k] = E\{\mathbf{y}[k].\mathbf{y}[k]^T\}$$

- Subject to (J) linear constraints to avoid source signal cancellation

$$\mathbf{C}^T.\mathbf{f} = \mathbf{b} \qquad \mathbf{f} \in \Re^{MN}, \mathbf{C} \in \Re^{MN \times J}, \mathbf{b} \in \Re^J$$

Example: fix array directivity pattern for angle ψ at J sample freqs ω_i

Lecture-3 p.22

$$\mathbf{F}^H(\omega_i).\mathbf{d}(\omega_i,\psi) \quad = \quad \underline{\underline{\mathbf{d}}}^T(\omega_i,\psi).\mathbf{f} = 1 \qquad i = 1..J$$

Note: For J→∞, constrained output power minimization corresponds to constrained noise output power minimization (if source signal angle is equal to ψ). (why?)
Hence, this LCMV formulation is appropriate for cases where R_{yy} can be estimated, while R_{nn} can not be estimated separately.
When the source signal is a speech signal, where R_{nn} is observable during speech pauses, then R_{nn} can be used instead of R_{yy} in the LCMV formulation (see p.21).

LCMV beamformer

LCMV = Linearly Constrained Minimum Variance

- Minimize output power $z[k]$:

$$\min_{\mathbf{f}} E\{z^2[k]\} = \min_{\mathbf{f}} \mathbf{f}^T \cdot \mathbf{R}_{yy}[k] \cdot \mathbf{f} \qquad \mathbf{R}_{yy}[k] = E\{\mathbf{y}[k].\mathbf{y}[k]^T\}$$

- Subject to (J) linear constraints to avoid source signal cancellation

$$\mathbf{C}^T.\mathbf{f} = \mathbf{b} \qquad \mathbf{f} \in \Re^{MN}, \mathbf{C} \in \Re^{MN \times J}, \mathbf{b} \in \Re^J$$

- Solution is (obtained using Lagrange-multipliers, etc..):

$$\mathbf{f}_{opt} = \mathbf{R}_{yy}^{-1}[k] \cdot \mathbf{C} \cdot \left(\mathbf{C}^T \cdot \mathbf{R}_{yy}^{-1}[k] \cdot \mathbf{C}\right)^{-1} \mathbf{b}$$

This can be computed for each time k: first estimate $R_{yy}^{-1}[k]$, etc..
But would rather have a truly recursive algorithm, see next slides

- PS: Compare to Chapter-3, p.26 & 36 (frequency domain & J=1, R_{nn}←→R_{yy})

Overview

- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction** **(/speech enhancement)**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=<u>spectral+spatial</u> filtering)

Generalized sidelobe canceler (GSC)

<u>GSC</u> = Adaptive filter solution for LCMV-problem

Constrained optimisation is reformulated as a *constraint pre-processing*, followed by an *unconstrained* optimisation, as follows:

- **<u>LCMV-problem</u>** is

$$\min_{\mathbf{f}} \mathbf{f}^T \cdot \mathbf{R}_{yy}[k] \cdot \mathbf{f}, \quad \mathbf{C}^T \cdot \mathbf{f} = \mathbf{b} \qquad \mathbf{f} \in \Re^{MN}, \mathbf{C} \in \Re^{MN \times J}, \mathbf{b} \in \Re^J$$

- Define `blocking matrix' $\mathbf{C}_a$, columns spanning the null-space of $\mathbf{C}$

$$\mathbf{C}^T.\mathbf{C}_a = \mathbf{0} \qquad \mathbf{C}_a \in \Re^{MN \times (MN-J)}$$

- Define 'quiescent response vector' $\mathbf{f}_q$ satisfying constraints

$$\mathbf{f}_q = \mathbf{C}.(\mathbf{C}^T.\mathbf{C})^{-1}\mathbf{b}$$

- Parametrize all **f**'s that satisfy constraints (verify!)

$$\mathbf{f} = \mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a \qquad \mathbf{f}_a \in \Re^{(MN-J)}$$

i.e. filter **f** can be decomposed in a *fixed* part $\mathbf{f}_q$ and a *variable* part $\mathbf{C}_a.\mathbf{f}_a$

Generalized sidelobe canceler (GSC)

GSC = Adaptive filter formulation of the LCMV-problem

Constrained optimisation is reformulated as a *constraint pre-processing*, followed by an *unconstrained* optimisation, as follows:

- **LCMV-problem** is

$$\min_{\mathbf{f}} \mathbf{f}^T \cdot \mathbf{R}_{yy}[k] \cdot \mathbf{f}, \quad \mathbf{C}^T \cdot \mathbf{f} = \mathbf{b} \qquad \mathbf{f} \in \Re^{MN}, \mathbf{C} \in \Re^{MN \times J}, \mathbf{b} \in \Re^{J}$$

- **Unconstrained optimization of $\mathbf{f}_a$:** (MN-J coefficients)

$$\mathbf{f} = \mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a$$

$$\min_{\mathbf{f}_a} (\mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a)^T.\mathbf{R}_{yy}[k].(\mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a) \qquad \mathbf{f}_a \in \Re^{(MN-J)}$$

Generalized sidelobe canceler

GSC (continued)

$$\min_{\mathbf{f}_a} (\mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a)^T.\overbrace{\mathbf{R}_{yy}[k]}^{\mathbf{R}_{yy}[k] = E\{\mathbf{y}[k].\mathbf{y}[k]^T\}}.(\mathbf{f}_q - \mathbf{C}_a.\mathbf{f}_a) = \ldots = \min_{\mathbf{f}_a} E\{\left|\overbrace{(\mathbf{y}[k]^T.\mathbf{f}_q)}^{\overset{\Delta}{=} d[k]} - \overbrace{(\mathbf{y}[k]^T.\mathbf{C}_a)}^{\overset{\Delta}{=} \tilde{\mathbf{y}}^T[\mathbf{k}]}.\mathbf{f}_a\right|^2\}$$

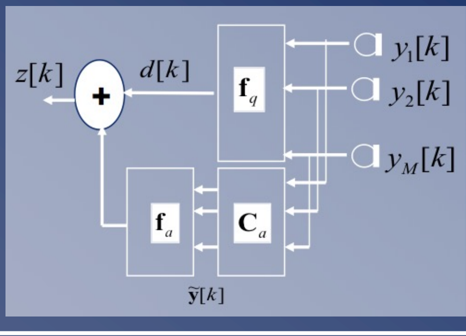
- Hence unconstrained optimization of $\mathbf{f}_a$ can be implemented as an adaptive filter (adaptive linear combiner), with filter inputs (='left-hand sides') equal to $\tilde{\mathbf{y}}[k] = \mathbf{y}[k]^T.\mathbf{C}$ and desired filter output (='right-hand side') equal to $d[k] = \mathbf{y}[k]^T.\mathbf{f}_q$
- **Example: LMS algorithm**

$$\mathbf{f}_a[k+1] = \mathbf{f}_a[k] + \mu.\underbrace{\mathbf{C}_a^T.\mathbf{y}[k]}_{\tilde{\mathbf{y}}[k]}.(\underbrace{\mathbf{f}_q^T.\mathbf{y}[k]}_{d[k]} - \underbrace{\mathbf{y}[k]^T.\mathbf{C}_a}_{\tilde{\mathbf{y}}[k]^T}.\mathbf{f}_a[k])$$

Generalized sidelobe canceler

GSC then consists of three parts:

- **Fixed beamformer** (cfr. f_q), satisfying constraints ('LC') but not yet minimum variance ('MV')), creating `source signal reference' $d[k]$
- **Blocking matrix (cfr.** C_a**),** beamformers with null response for angle ψ (at sampling frequencies) (cfr. $C'.C_a=0$), creating MN-J `noise references' $\tilde{\mathbf{y}}[k]$
- **Multi-channel adaptive filter** (linear combiner) your favourite one, e.g. LMS

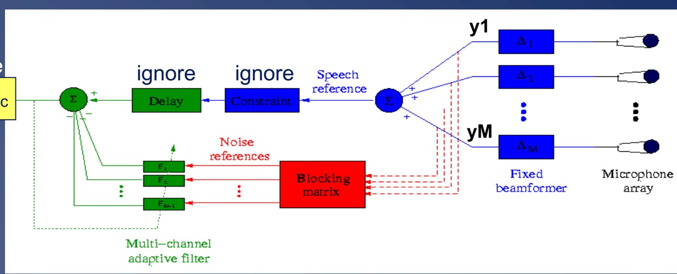


PS: $\mathbf{C}_a \in \Re^{MN \times (MN-J)}$ = large matrix complexity O(MN.(MN-J)) even with LMS for f_a

17

Generalized sidelobe canceler

A popular & cheaper GSC realization is as follows



Note that some reorganization has been done:

The blocking matrix now generates (typically) M-1 (instead of MN-J) noise references.

The multichannel adaptive filter performs FIR-filtering on each noise reference (instead of merely scaling in the linear combiner) (=complexity O((M-1)N) for LMS)

Philosophy is the same, mathematics are different (details on next slide).

18

Generalized sidelobe canceler

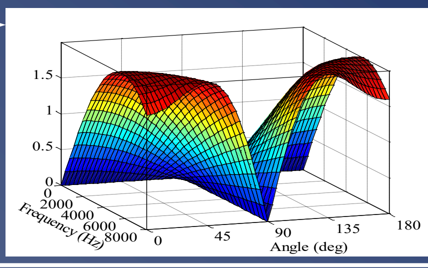
A popular & cheaper GSC realization is as follows (continued)

- Blocking matrix
  - Creating (M-1) independent noise references by means of (M-1) beamformers with null response for angle ψ
  - Different possibilities, e.g.

$\mathbf{f}_q^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$ (ψ=90 , i.e. 'broadside steering')

$\tilde{\mathbf{C}}_a^T = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}$ Griffiths-Jim

$\tilde{\mathbf{C}}_a^T = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$ Walsh



Generalized sidelobe canceler

- Math details: (ψ=90 , i.e. 'broadside steering')

$$\tilde{\mathbf{y}}[k] = \mathbf{C}_a^T.\mathbf{y}[k] = \mathbf{C}_{a,\text{permuted}}^T.\mathbf{y}_{\text{permuted}}[k]$$

$$\mathbf{y}_{\text{permuted}}[k] = \begin{bmatrix} \mathbf{y}_{1:M}^T[k] & \mathbf{y}_{1:M}^T[k-1] & \dots & \mathbf{y}_{1:M}^T[k-L+1] \end{bmatrix}^T$$

$$\mathbf{y}_{1:M}[k] = \begin{bmatrix} y_1[k] & y_2[k] & \dots & y_M[k] \end{bmatrix}^T$$

select `sparse' blocking matrix such that :

$$\mathbf{C}_{a,\text{permuted}}^T = \begin{bmatrix} \tilde{\mathbf{C}}_a^T & 0 & \dots & 0 \\ 0 & \tilde{\mathbf{C}}_a^T & \dots & 0 \\ : & : & & : \\ 0 & 0 & \dots & \tilde{\mathbf{C}}_a^T \end{bmatrix}$$

$$\tilde{\mathbf{y}}[k] = \begin{bmatrix} \tilde{\mathbf{y}}_{1:M}^T[k] & \tilde{\mathbf{y}}_{1:M}^T[k-1] & \dots & \tilde{\mathbf{y}}_{1:M}^T[k-L+1] \end{bmatrix}^T$$

$$\tilde{\mathbf{y}}_{1:M}[k] = \tilde{\mathbf{C}}_a^T.\mathbf{y}_{1:M}[k]$$

=input to multi-channel adaptive filter

=use this as blocking matrix now

Generalized sidelobe canceler

- Problems of GSC:
  - Impossible to reduce noise from angle **ψ** (cfr. constraint)
  - If steering vectors $\mathbf{d}(\omega_i,\psi)$ (see p.11) are not accurately known (e.g. due to inaccurate source position or reverberation), will have source signal 'leakage' into noise references, which can lead to source signal cancellation by the adaptive filter.

    Therefore, to avoid source signal cancellation, adaptive filter should only be updated when no source signal is present (i.e. in **noise-only periods**).

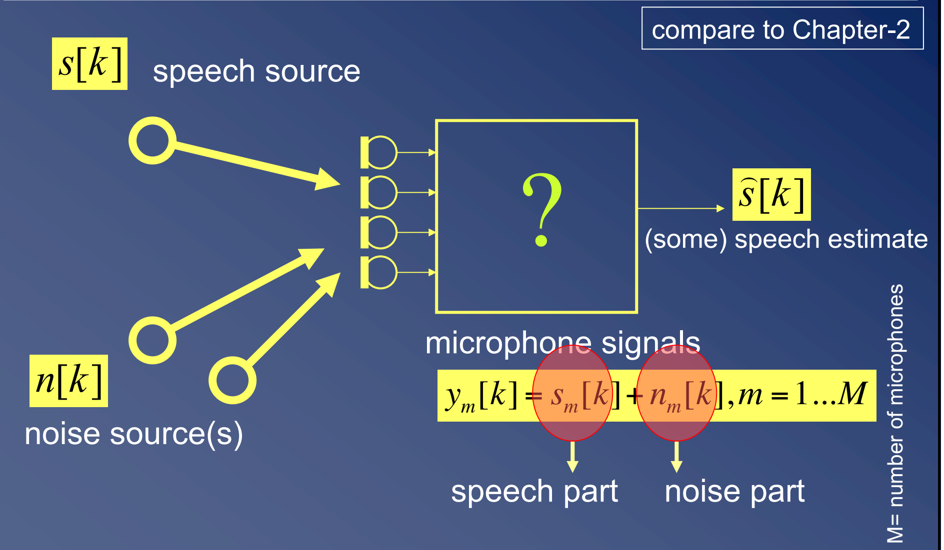
    In speech applications, this is done with a **voice activity detection** (VAD).

    Effectively, this corresponds to replacing R_{yy} by R_{nn} in p.11 (and following slides)

Overview

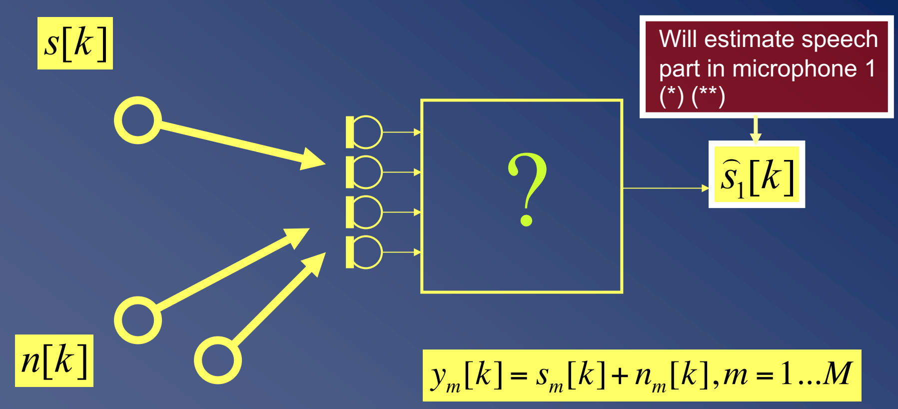
- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction (/speech enhancement)**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=spectral+spatial filtering)

Multi-Microphone Noise Reduction Problem



23

Multi-Microphone Noise Reduction Problem



(*) Estimating s[k] is more difficult, would include dereverberation

(**) This is similar to single-microphone model (Chapter-2), where additional microphones (m=2..M) help to get a better estimate

24

Multi-Microphone Noise Reduction Problem

- Data model:
  (frequency domain)

$$\mathbf{Y}(\omega) = \mathbf{S}(\omega) + \mathbf{N}(\omega)$$
$$= \mathbf{d}(\omega).S(\omega) + \mathbf{N}(\omega)$$

$$\begin{bmatrix} Y_1(\omega) \\ Y_2(\omega) \\ \vdots \\ Y_M(\omega) \end{bmatrix} = \begin{bmatrix} H_1(\omega) \\ H_2(\omega) \\ \vdots \\ H_M(\omega) \end{bmatrix} .S(\omega) + \begin{bmatrix} N_1(\omega) \\ N_2(\omega) \\ \vdots \\ N_M(\omega) \end{bmatrix}$$

See Chapter-3 on multi-path propagation (with **q** left out for conciseness)

$H_m(\omega)$ is complete transfer function from speech source position to *m*-the microphone

No prior knowledge assumed here!

Overview

- **Recap**
- **Adaptive Beamforming**
  - Adaptive beamforming goal & set-up
  - LCMV beamformer
  - Generalized sidelobe canceler
- **Multi-channel Wiener filter for multi-microphone noise reduction** **(/speech enhancement)**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=<u>spectral+spatial</u> filtering)

Multi-Channel Wiener Filter (MWF)

- Data model:

$$\mathbf{Y}(\omega) = \mathbf{d}(\omega).S(\omega) + \mathbf{N}(\omega)$$

- Will use <u>linear filters</u> to obtain speech estimate

$$\hat{S}_1(\omega) = \sum_{m=1}^{M} F_m^*(\omega).Y_m(\omega) = \mathbf{F}^H(\omega).\mathbf{Y}(\omega)$$

- <u>Wiener filter</u> (=**linear MMSE** approach)

$$\min_{\mathbf{F}(\omega)} E\{\left|S_1(\omega) - \mathbf{F}^H(\omega).\mathbf{Y}(\omega)\right|^2\}$$

Note that (unlike in DSP-CIS) `desired response' signal $S_1(w)$ is **unknown** here (!)

Multi-Channel Wiener Filter (MWF)

- Wiener filter solution is (see DSP-CIS)

$$\mathbf{F}(\omega) = \underbrace{E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}}_{\text{autocorrelation}}^{-1} \underbrace{E\{\mathbf{Y}(\omega).S_1^*(\omega)\}}_{\text{crosscorrelation}}.$$

$$= \ldots \qquad (\text{with } E\{\mathbf{S}(\omega).N_1^*(\omega)\} = 0)$$

$$= E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\underbrace{\begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T}_{\mathbf{e}_1}$$

compute during speech+noise periods

compute during noise-only periods

- All quantities can be estimated !
- Special case of this is single-channel Wiener filter formula (Chapter-2)
- In practice, use alternative to 'subtraction' operation (see slide 32)

Multi-Channel Wiener Filter (MWF)

- MWF combines spatial filtering with single-channel spectral filtering (as in Chapter-2) :

if

$$\underbrace{\begin{bmatrix} Y_1(\omega) \\ Y_2(\omega) \\ \vdots \\ Y_M(\omega) \end{bmatrix}}_{\mathbf{Y}(\omega)} = \underbrace{\mathbf{d}(\omega)}_{\text{steering vector}} .S(\omega) + \underbrace{\mathbf{N}(\omega)}_{\text{noise}}$$

$$E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\} = \mathbf{\Phi}_{noise}(\omega)$$

then…

Multi-Channel Wiener Filter (MWF)

…then it can be shown that

$$\mathbf{F}(\omega) = \underbrace{\alpha(\omega)}_{\text{scalar}} . \underbrace{\mathbf{\Phi}^{-1}_{noise}(\omega).\mathbf{d}(\omega)}_{\overline{\mathbf{F}}(\omega)}$$

① $\overline{\mathbf{F}}(\omega) = \mathbf{\Phi}^{-1}_{noise}(\omega).\mathbf{d}(\omega)$ **represents a spatial filtering (*)**

Compare to superdirective & matched filter beamforming (Chapter-3)

- Matched filter beamf. maximizes array gain in white noise field
- Superdirective beamf. maximizes array gain in diffuse noise field
- MWF maximizes array gain in unknown (!) noise field.

MWF is operated without invoking any prior knowledge (steering vector/noise field) ! (the secret is in the voice activity detection… (explain))

(*) Note that spatial filtering can improve SNR, spectral filtering never improves SNR (at one frequency)

Multi-Channel Wiener Filter (MWF)

…then it can be shown that

$$\mathbf{F}(\omega) = \underbrace{\alpha(\omega)}_{\text{scalar}} . \underbrace{\Phi^{-1}_{noise}(\omega).\mathbf{d}(\omega)}_{\bar{\mathbf{F}}(\omega)}$$

① $\bar{\mathbf{F}}(\omega) = \Phi^{-1}_{noise}(\omega).\mathbf{d}(\omega)$ **represents a <u>spatial filtering</u> (*)**

② $\alpha(\omega)$ **represents an additional <u>spectral 'post-filter'</u>**

i.e. single-channel Wiener estimate of $S_1(\omega)$ (Chapter-2 p.9)
applied to output signal of spatial filter (...prove it!)

$$\alpha(\omega) = \frac{\mathrm{E}\left\{S_1^*(\omega).\bar{\mathbf{F}}^H(\omega).\mathbf{Y}(\omega)\right\}}{\mathrm{E}\left\{\left|\bar{\mathbf{F}}^H(\omega).\mathbf{Y}(\omega)\right|^2\right\}} = \ldots = \frac{\left|S(\omega)\right|^2 .H_1^*(\omega)}{\bar{\sigma}^2_{noise}.\left|S(\omega)\right|^2 + 1}$$

$$\text{with } \mathrm{E}\left\{\left|\bar{\mathbf{F}}^H(\omega).\mathbf{Y}(\omega)\right|^2\right\} = \left(\mathbf{d}^H(\omega).\Phi^{-1}_{noise}(\omega).\mathbf{d}(\omega)\right)^2.\left|S(\omega)\right|^2 + \underbrace{\left(\mathbf{d}^H(\omega).\Phi^{-1}_{noise}(\omega).\mathbf{d}(\omega)\right)}_{\bar{\sigma}^2_{noise}}$$

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- Correlation matrices

$$E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} \text{ and } E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}$$

are estimated in each frame by averaging over a number of previous frames, possibly with exponential weighting, i.e.

$$\overbrace{\hat{\Phi}_{\text{s\&n}}}^{\text{frame-i}} \Leftarrow \lambda^2.\overbrace{\hat{\Phi}_{\text{s\&n}}}^{\text{frame-(i-1)}} + (1-\lambda^2).\overbrace{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)}^{\text{frame-i}}$$ Update in speech+noise frames

$$\hat{\Phi}_{\text{noise}} \Leftarrow \lambda^2.\hat{\Phi}_{\text{noise}} + (1-\lambda^2).\mathbf{N}(\omega).\mathbf{N}^H(\omega)$$ Update in noise-only frames

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- Note that (in the above formula)

$$E\{\mathbf{Y}(\omega)\cdot\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega)\cdot\mathbf{N}^H(\omega)\} = E\{\mathbf{S}(\omega)\cdot\mathbf{S}^H(\omega)\} = \mathbf{d}(\omega)\cdot E\{S(\omega)\cdot S^H(\omega)\}\cdot\mathbf{d}^H(\omega)$$

is a positive-definite <u>rank-1</u> matrix, whereas (with estimated matrices)

$$\hat{\Phi}_{\text{s\&n}} - \hat{\Phi}_{\text{noise}}$$

is generally not rank-1, and often not positive-definite

Hence

$$\hat{\mathbf{F}}(\omega) = \hat{\Phi}_{\text{s\&n}}^{-1}.\left(\hat{\Phi}_{\text{s\&n}} - \hat{\Phi}_{\text{noise}}\right).\mathbf{e}_1$$

generally provides a poor filter estimate..

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- A better procedure (v1.0) could be as follows

$$\hat{\mathbf{F}}(\omega) = \hat{\Phi}_{\text{s\&n}}^{-1}.\left(\hat{\Phi}_{\text{speech}}\right).\mathbf{e}_1$$

where ϕ_{speech} is a <u>rank-1</u> matrix estimated (in each frame) as

$$\hat{\Phi}_{\text{speech}} = \operatorname{argmin}_{\substack{\Phi \\ \text{rank}(\Phi)=1}} \left\| (\hat{\Phi}_{\text{s\&n}} - \hat{\Phi}_{\text{noise}}) - \Phi \right\|_F^2$$

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- An even better procedure (v2.0) is as follows

$$\hat{\mathbf{F}}(\omega) = \hat{\Phi}_{\text{s\&n}}^{-1}.\left(\hat{\Phi}_{\text{speech}}\right).\mathbf{e}_1$$

where ϕ_{speech} is a rank-1 matrix estimated (in each frame) as

$$\hat{\Phi}_{\text{speech}} = \operatorname{argmin}_{\substack{\Phi \\ \text{rank}(\Phi)=1}} \left\| \hat{\Phi}_{\text{noise}}^{-1/2}.((\hat{\Phi}_{\text{s\&n}} - \hat{\Phi}_{\text{noise}}) - \Phi).\hat{\Phi}_{\text{noise}}^{-H/2} \right\|_F^2 \qquad \hat{\Phi}_{\text{noise}} = \hat{\Phi}_{\text{noise}}^{1/2}\,\hat{\Phi}_{\text{noise}}^{H/2}$$

which now also includes a 'noise whitening' operation

(making the estimation also immune to, for instance, scalings of the microphone signals)

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- Solution is (for v2.0) is based on

Generalized Eigenvalue Decomposition

of matrix pair {ϕ^$_{\text{s\&n}}$, ϕ^$_{\text{noise}}$} :

$$\hat{\Phi}_{\text{s\&n}} = Q \cdot \Sigma_{\text{s\&n}} \cdot Q^H = Q \cdot diag\{\sigma_{\text{s\&n},1}, \sigma_{\text{s\&n},2}, \ldots, \sigma_{\text{s\&n},M}\} \cdot Q^H$$

$$\hat{\Phi}_{\text{noise}} = Q \cdot \Sigma_{\text{noise}} \cdot Q^H = Q \cdot diag\{\sigma_{\text{noise},1}, \sigma_{\text{noise},2}, \ldots, \sigma_{\text{noise},M}\} \cdot Q^H$$

where Q is an invertible matrix

a.k.a 'joint diagonalization'

This correspondes to an eigenvalue decomposition of $\hat{\Phi}_{\text{noise}}^{-1}\hat{\Phi}_{\text{s\&n}}$:

$$\hat{\Phi}_{\text{noise}}^{-1}\hat{\Phi}_{\text{s\&n}} = Q^{-H} \cdot \Sigma_{\text{noise}}^{-1}\Sigma_{\text{s\&n}} \cdot Q^H = Q^{-H} \cdot diag\{\frac{\sigma_{\text{s\&n},1}}{\sigma_{\text{noise},1}}, \frac{\sigma_{\text{s\&n},2}}{\sigma_{\text{noise},2}}, \ldots, \frac{\sigma_{\text{s\&n},M}}{\sigma_{\text{noise},M}}\} \cdot Q^H$$

where eigenvalues can be sorted such that $\frac{\sigma_{\text{s\&n},1}}{\sigma_{\text{noise},1}} \geq \frac{\sigma_{\text{s\&n},2}}{\sigma_{\text{noise},2}} \geq \ldots \geq \frac{\sigma_{\text{s\&n},M}}{\sigma_{\text{noise},M}}$

Multi-Channel Wiener Filter (MWF)

$$\mathbf{F}(\omega) = E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\} - E\{\mathbf{N}(\omega).\mathbf{N}^H(\omega)\}\right).\mathbf{e}_1$$

- With this ϕ_{speech} (in v2.0) becomes (proof omitted)

$$\hat{\Phi}_{speech} = Q \cdot diag\{\sigma_{s\&n,1} - \sigma_{noise,1}, 0, \ldots, 0\} \cdot Q^H = \underbrace{(Q \cdot \mathbf{e}_1)}_{\substack{\text{estimate of steering vector} \\ \text{(up to scalar)}}} \cdot (\sigma_{s\&n,1} - \sigma_{noise,1}) \cdot (Q \cdot \mathbf{e}_1)^H$$

and then

$$\hat{\mathbf{F}}(\omega) = \hat{\Phi}^{-1}_{s\&n} \cdot \left(\hat{\Phi}_{speech}\right) \cdot \mathbf{e}_1$$

$$= Q^{-H} \cdot diag\{\frac{\sigma_{s\&n,1} - \sigma_{noise,1}}{\sigma_{s\&n,1}}, 0, \ldots, 0\} \cdot Q^H \cdot \mathbf{e}_1$$

$$= \underbrace{(\frac{\sigma_{s\&n,1} - \sigma_{noise,1}}{\sigma_{s\&n,1}} \cdot [Q^H]_{11})}_{\text{scalar}} \cdot \underbrace{Q^{-H} \cdot \mathbf{e}_1}_{\substack{\text{principal eigenvector} \\ \text{of } \Phi^{-1}_{noise} \cdot \Phi_{s\&n}}}$$

PS: If k[th] microphone is reference microphone (instead of 1[st] microphone, see page 24) then first line has $\mathbf{e}_k$, last line has $[Q^H]_{1k}$ and $\mathbf{e}_1$

37

Multi-Channel Wiener Filter (MWF)

$$\hat{\mathbf{F}}(\omega) = \underbrace{(\frac{\sigma_{s\&n,1} - \sigma_{noise,1}}{\sigma_{s\&n,1}} \cdot [Q^H]_{11})}_{\text{scalar}} \cdot \underbrace{Q^{-H} \cdot \mathbf{e}_1}_{\substack{\text{principal eigenvector} \\ \text{of } \Phi^{-1}_{noise} \cdot \Phi_{s\&n}}}$$

- Scalar is representative of **spectral filtering**:
  Compare with spectral subtraction formulas…
  Eigenvalues ($\sigma_{s\&n}/\sigma_{noise}$) represent signal-to-noise ratios
- Eigenvector is representative of **spatial filtering:**

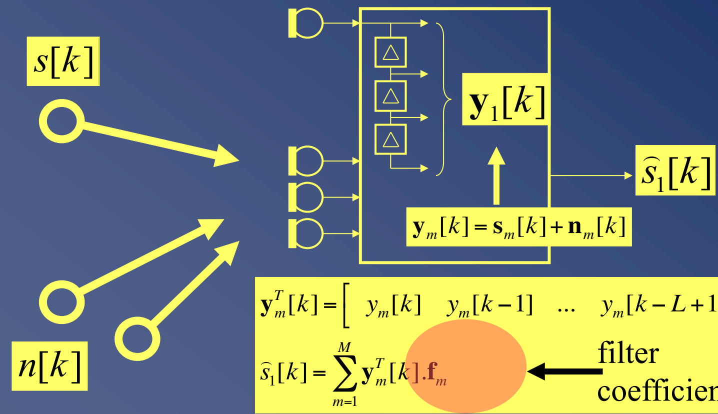
$$Q^{-H} \cdot \mathbf{e}_1 \overset{\hat{\Phi}^{-1}_{noise} = Q^{-H} \cdot \Sigma^{-1}_{noise} \cdot Q^{-1}}{=} (\sigma_{noise,1}) \cdot \left[\hat{\Phi}^{-1}_{noise} \cdot \underbrace{(Q \cdot \mathbf{e}_1)}_{\substack{\text{estimate of steering vector} \\ \text{(up to scalar)}}}\right]$$

Compare to p.30-31

38

Multi-Channel Wiener Filter: Implementation

- Implementation with short-time Fourier transform (see Chapter-2)
- Implementation with time-domain **FIR filtering**:



$$\mathbf{y}_m^T[k] = \begin{bmatrix} y_m[k] & y_m[k-1] & \dots & y_m[k-L+1] \end{bmatrix}$$

$$\hat{s}_1[k] = \sum_{m=1}^{M} \mathbf{y}_m^T[k].\mathbf{f}_m$$

39

Multi-Channel Wiener Filter: Implementation

- Implementation with time-domain **linear filtering**:

$$\min_{\underline{\mathbf{f}}} E\{|s_1[k] - \underline{\underline{\mathbf{y}}}^T[k].\underline{\mathbf{f}}|^2\}$$

$$\underline{\mathbf{f}} = [\mathbf{f}_1^T \quad \mathbf{f}_2^T \quad \dots \quad \mathbf{f}_M^T]^T$$

$$\underline{\underline{\mathbf{y}}}[k] = [\mathbf{y}_1^T[k] \quad \mathbf{y}_2^T[k] \quad \dots \quad \mathbf{y}_M^T[k]]^T$$

Solution is…

$$\underline{\mathbf{f}} = \left[E\{\underline{\mathbf{y}}[k].\underline{\mathbf{y}}[k]^T\}\right]^{-1}.E\{\underline{\mathbf{y}}[k].s_1[k]\}$$

$$= \left[E\{\underline{\mathbf{y}}[k].\underline{\mathbf{y}}[k]^T\}\right]^{-1}.\left[E\{\underline{\mathbf{y}}[k].y_1[k]\} - E\{\underline{\mathbf{n}}[k].n_1[k]\}\right]$$

compute during speech+noise periods

compute during noise-only periods

40