Digital Audio Signal Processing

Lecture-3
Noise Reduction

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

Overview

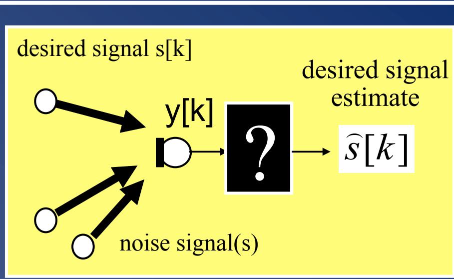
- **Spectral subtraction for single-micr. noise reduction**
  - Single-microphone noise reduction problem
  - Spectral subtraction basics (=spectral filtering)
  - Features: gain functions, implementation, musical noise,…
  - Iterative Wiener filter based on speech signal model
- **Multi-channel Wiener filter for multi-micr. noise red.**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=spectral+spatial filtering)
- **Kalman filter based noise reduction**
  - Kalman filters
  - Kalman filters for noise reduction

Single-Microphone Noise Reduction Problem

- **Microphone signal** is

$$y[k] = s[k] + n[k]$$

$s[k]$: desired signal contribution
$n[k]$: noise contribution



- **Goal:** Estimate s[k] based on y[k]
- **Applications:**
  Speech enhancement in conferencing, handsfree telephony, hearing aids, …
  Digital audio restoration
- **Will consider <u>speech applications</u>:** s[k] = speech signal

Spectral Subtraction Methods: Basics

$$\boxed{y[k] = s[k] + n[k]}$$

- Signal chopped into `<u>frames</u>' (e.g. 10..20msec), for each frame a frequency domain representation is

$$Y_i(\omega) = S_i(\omega) + N_i(\omega) \quad \textit{(i-th frame)}$$

- However, speech signal is an on/off signal, hence some frames have **<u>speech +noise</u>**, i.e.

$$Y_i(\omega) = S_i(\omega) + N_i(\omega) \qquad \text{frame}_i \in \{\text{`speech + noise' frames}\}$$

some frames have **<u>noise only</u>**, i.e.

$$Y_i(\omega) = \quad 0 \quad + N_i(\omega) \qquad \text{frame}_i \in \{\text{`noise-only' frames}\}$$

- A **<u>speech detection algorithm</u>** is needed to distinguish between these 2 types of frames (based on energy/dynamic range/statistical properties,…)

Spectral Subtraction Methods: Basics

- Definition: $\mu(\omega)$ = average amplitude of noise spectrum

  $$\mu(\omega) = E\{|N_i(\omega)|\}$$

- Assumption: noise characteristics change slowly, hence estimate $\mu(\omega)$ by (long-time) averaging over (M) noise-only frames

  $$\hat{\mu}(\omega) = \frac{1}{M} \sum_{M \text{ noise-only frames}} |Y_i(\omega)|$$

- Estimate clean speech spectrum $Si(\omega)$ *(for each frame)*, using corrupted speech spectrum $Yi(\omega)$ *(for each frame, i.e. short-time estimate)* + estimated $\mu(\omega)$:

  $$\hat{S}_i(\omega) = G_i(\omega) Y_i(\omega)$$

  based on `gain function' $$G_i(\omega) = f(Y_i(\omega), \hat{\mu}(\omega))$$

Spectral Subtraction: Gain Functions

Magnitude Subtraction	$G_i(\omega) = \left[1 - \frac{\hat{\mu}(\omega)}{\lvert Y_i(\omega)\rvert}\right]$
Spectral Subtraction	$G_i(\omega) = \sqrt{1 - \rho \frac{\hat{\mu}^2(\omega)}{\lvert Y_i(\omega)\rvert^2}}$
Wiener Estimation	$G_i(\omega) = 1 - \frac{\hat{\mu}^2(\omega)}{\lvert Y_i(\omega)\rvert^2}$
Maximum Likelihood	$G_i(\omega) = \frac{1}{2}\left[1 + \sqrt{1 - \frac{\hat{\mu}^2(\omega)}{\lvert Y_i(\omega)\rvert^2}}\right]$
Non-linear Estimation	$G_i(\omega) = f(\hat{\mu}(\omega), \lvert Y_i(\omega)\rvert)$
Ephraim-Malah = most frequently used in practice	$G_i(\omega) = f(\text{SNR}_{\text{post}}, \text{SNR}_{\text{prio}})$ see next slide

Spectral Subtraction: Gain Functions

skip formulas

- **Example 1: Ephraim-Malah Suppression Rule (EMSR)**

$$G_i(\omega)=\frac{\sqrt{\pi}}{2}\sqrt{\left(\frac{1}{\mathrm{SNR}_{\mathrm{post}}}\right)\left(\frac{\mathrm{SNR}_{\mathrm{prio}}}{1+\mathrm{SNR}_{\mathrm{prio}}}\right)}.M\left[\mathrm{SNR}_{\mathrm{post}}\left(\frac{\mathrm{SNR}_{\mathrm{prio}}}{1+\mathrm{SNR}_{\mathrm{prio}}}\right)\right]$$

with:

$$\begin{aligned} M[\theta] &= e^{-\frac{\theta}{2}}\left[(1+\theta)I_0(\frac{\theta}{2})+\theta I_1(\frac{\theta}{2})\right] \\ \mathrm{SNR}_{\mathrm{post}}(\omega) &= \frac{|Y_i(\omega)|^2}{\hat{\mu}(\omega)^2} \\ \mathrm{SNR}_{\mathrm{prio}}(\omega) &= (1\text{-}\alpha)\max(\mathrm{SNR}_{\mathrm{post}}\text{-}1,0)+\alpha\frac{|G_{i-1}(\omega)Y_{i-1}(\omega)|^2}{\hat{\mu}(\omega)^2} \end{aligned}$$

modified Bessel functions

- This corresponds to a **MMSE** (*) estimation of the speech spectral amplitude $|S_i(\omega)|$ based on observation $Y_i(\omega)$ (estimate equal to **E{** $|S_i(\omega)|$ **|** $Y_i(\omega)$ **}**) assuming Gaussian a priori distributions for $S_i(\omega)$ and $N_i(\omega)$ [Ephraim & Malah 1984].
- Similar formula for MMSE log-spectral amplitude estimation [Ephraim & Malah 1985].

(*) minimum mean squared error

Spectral Subtraction: Gain Functions

- **Example 2: Magnitude Subtraction**
  - Signal model:

$$\begin{aligned} Y_i(\omega) &= S_i(\omega)+N_i(\omega) \\ &= |Y_i(\omega)|e^{j\theta_{y,i}(\omega)} \end{aligned}$$

  - Estimation of clean speech spectrum:

$$\begin{aligned} \hat{S}_i(\omega) &= \left[|Y_i(\omega)|-\hat{\mu}(\omega)\right]e^{j\theta_{y,i}(\omega)} \\ &= \underbrace{\left[1-\frac{\hat{\mu}(\omega)}{|Y_i(\omega)|}\right]}_{G_i(\omega)}Y_i(\omega) \end{aligned}$$

  - PS: half-wave rectification

$$G_i(\omega)\Leftarrow\max(0,G_i(\omega))$$

Spectral Subtraction: Gain Functions

- **Example 3: Wiener Estimation**
  - **<u>Linear MMSE</u> estimation**:
    find linear filter *Gi(ω)* to minimize MSE

$$= E\left\{ \left| S_i(\omega) - \overbrace{G_i(\omega).Y_i(\omega)}^{\hat{S}_i(\omega)} \right|^2 \right\}$$

  - Solution:

$$G_i(\omega) = \frac{E\{S_i(\omega).Y_i(\omega)\}}{E\{Y_i(\omega).Y_i(\omega)\}} = \frac{P_{sy,i}(\omega)}{P_{yy,i}(\omega)}$$

<- cross-correlation in i-th frame

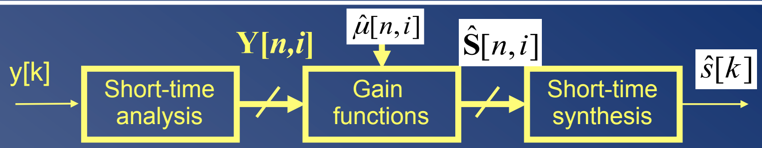
<- auto-correlation in i-th frame

Assume speech *s[k]* and noise *n[k]* are uncorrelated, then...

$$G_i(\omega) = \frac{P_{ss,i}(\omega)}{P_{yy,i}(\omega)} = \frac{P_{yy,i}(\omega) - P_{nn,i}(\omega)}{P_{yy,i}(\omega)} = \frac{|Y_i(\omega)|^2 - \hat{\mu}(\omega)^2}{|Y_i(\omega)|^2} = 1 - \frac{\hat{\mu}(\omega)^2}{|Y_i(\omega)|^2}$$

  - PS: half-wave rectification

Spectral Subtraction: Implementation



→ <u>Short-time Fourier Transform</u> (=uniform DFT-modulated analysis filter bank)

$$Y[n,i] = \sum_{k=0}^{K-1} h[k] y[i.D-k] e^{-j2\pi kn/N}$$

= estimate for $Y(\omega_n)$ at time *i* (*i*-th frame)

N=number of frequency bins (channels) n=0..N-1
D=downsampling factor
K=frame length h[k] = length-K analysis window (=prototype filter)

→ frames with 50%...66% overlap (i.e. 2-, 3-fold oversampling, N=2D..3D)
→ subband processing: $\hat{S}[n,i] = G[n,i].Y[n,i]$

→ synthesis bank: matched to analysis bank (see DSP-CIS)

Spectral Subtraction: Musical Noise

- Audio demo: car noise

$y[k]$ → magnitude subtraction → $\hat{s}[k]$

- Artifact: **musical noise**

What?

Short-time estimates of |*Yi(ω)*| fluctuate randomly in noise-only frames, resulting in random gains *Gi(ω)*

→ *statistical analysis shows that broadband noise is transformed into signal composed of short-lived tones with randomly distributed frequencies (=musical noise)*

Spectral Subtraction: Musical Noise

Solutions?

- Magnitude averaging: replace *Yi(ω)* in calculation of *Gi(ω)* by a local average over frames

$$\hat{S}_i(\omega) = \underbrace{G(\omega)}_{\text{average}}\underbrace{Y_i(\omega)}_{\text{instantaneous}}$$

- EMSR (p7)
- augment *Gi(ω)* with soft-decision VAD:

$$Gi(\omega) \rightarrow \underbrace{P(H_1 \mid Yi(\omega))}_{\text{probability that speech is present, given observation}} .\ Gi(\omega)$$

…

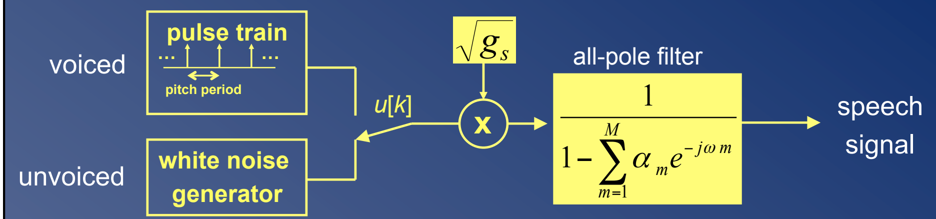
Spectral Subtraction: Iterative Wiener Filter

Example of signal model-based spectral subtraction…

- **Basic**:
  Wiener filtering based spectral subtraction (p.9), with (improved) spectra estimation based on parametric models

- **Procedure**:
  1. Estimate parameters of a speech model from noisy signal *y[k]*
  2. Using estimated speech parameters, perform noise reduction (e.g. Wiener estimation, p. 9)
  3. Re-estimate parameters of speech model from the speech signal estimate
  4. Iterate 2 & 3

Spectral Subtraction: Iterative Wiener Filter



$$\sqrt{g_s}$$

$$\frac{1}{1-\sum_{m=1}^{M}\alpha_m e^{-j\omega m}}$$

frequency domain:

$$S(\omega)=\frac{\sqrt{g_s}}{1-\sum_{m=1}^{M}\alpha_m e^{-j\omega m}}U(\omega)$$

time domain:

$$s[k]=\sum_{m=1}^{M}\alpha_m s[k-m]+\sqrt{g_s}\,u[k]$$

$\boldsymbol{\alpha}=\begin{bmatrix}\alpha_1 & \cdots & \alpha_M\end{bmatrix}^T$ = linear prediction parameters

Spectral Subtraction: Iterative Wiener Filter

For each frame (vector) **y**[*m*] (i=iteration nr.)

1. Estimate $g_{s,i}$ and $\boldsymbol{\alpha}_i = [\alpha_{1,i} \ \cdots \ \alpha_{M,i}]^T$
2. Construct Wiener Filter (p.9)

$$G(\omega) = .. = \frac{P_{ss}(\omega)}{P_{ss}(\omega) + P_{nn}(\omega)}$$

with:

- $P_{nn}(\omega)$ estimated during noise-only periods
- $$P_{ss}(\omega) \approx \frac{g_{s,i}}{\left|1 - \sum_{m=1}^{M} \alpha_{m,i} e^{-j\omega m}\right|^2}$$

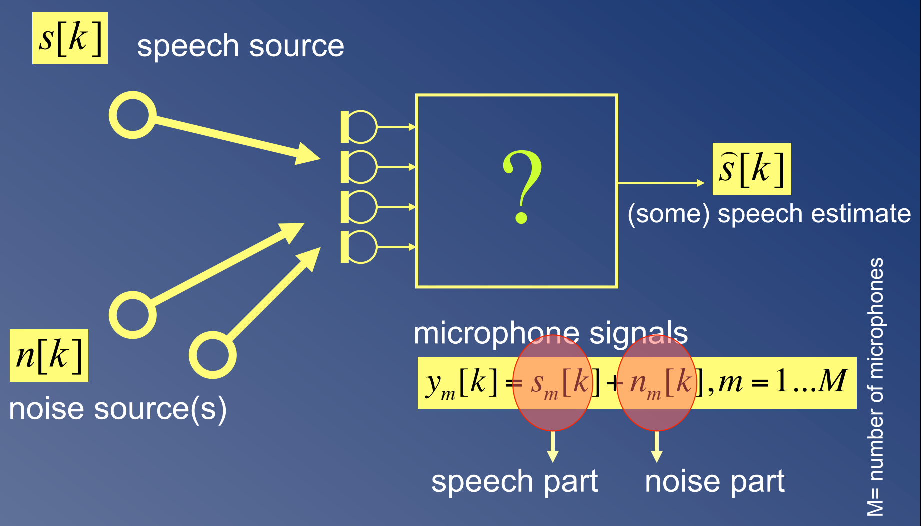
3. Filter speech frame **y**[*m*] → $\hat{\mathbf{s}}_i[m]$

Repeat until some error criterion is satisfied

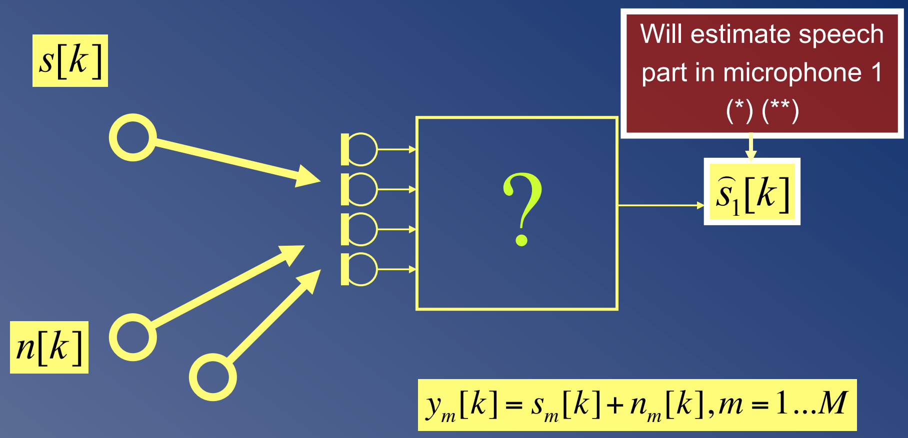
Overview

- **Spectral subtraction for <u>single</u>-micr. noise reduction**
  - Single-microphone noise reduction problem
  - Spectral subtraction basics (=<u>spectral filtering</u>)
  - Features: gain functions, implementation, musical noise,…
  - Iterative Wiener filter based on speech signal model
- **Multi-channel Wiener filter for <u>multi</u>-micr. noise red.**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=<u>spectral+spatial</u> filtering)
- **Kalman filter based noise reduction**
  - Kalman filters
  - Kalman filters for noise reduction

Multi-Microphone Noise Reduction Problem



Multi-Microphone Noise Reduction Problem



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

(**) This is similar to single-microphone model (p.3), where additional microphones (m=2..M) help to get a better estimate

Multi-Microphone Noise Reduction Problem

- Data model:

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

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

See Lecture-2 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

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 (as in Lecture-2)

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

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

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

Note that (unlike in DSP-CIS) `desired response' signal $S_1$(w) is **unknown** here (!), hence solution will be `unusual' …

Multi-Channel Wiener Filter (MWF)

- Wiener filter solution is (see DSP-CIS)

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

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

$$= E\{\mathbf{Y}(\omega).\mathbf{Y}^H(\omega)\}^{-1}.\left(E\{\mathbf{Y}(\omega).Y_1^*(\omega)\} - E\{\mathbf{N}(\omega).N_1^*(\omega)\}\right)$$

compute during speech+noise periods

compute during noise-only periods

  - All quantities can be computed !
  - Special case of this is single-channel Wiener filter formula on p.9
  - In practice, use alternative to 'subtraction' operation (see literature)

Multi-Channel Wiener Filter (MWF)

- **Note that…**

  MWF combines spatial filtering (as in Lecture-2) with single-channel spectral filtering (as in single-channel noise reduction)

if

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

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

then…

Multi-Channel Wiener Filter (MWF)

…then it can be shown that

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

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

Compare to superdirective & delay-and-sum beamforming (Lecture-2)

– Delay-and-sum 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 <u>without invoking any prior knowledge</u> (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

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

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

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

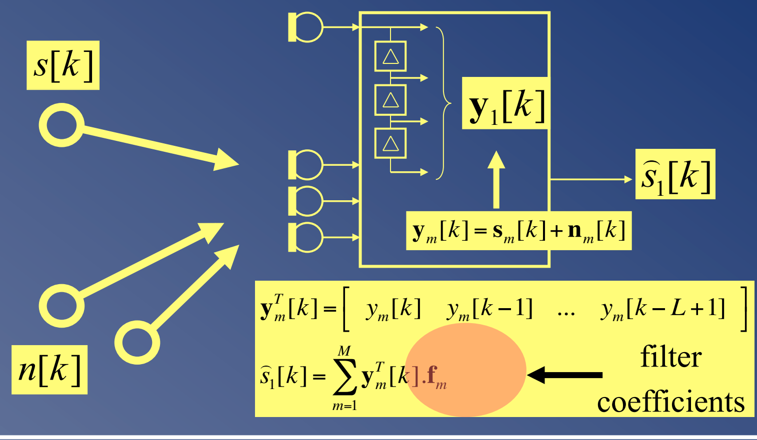
i.e. single-channel Wiener estimate (p.9), applied to output signal of spatial filter

$$\alpha(\omega) = \ldots = \frac{|S(\omega)|^2 . H_1^*(\omega)}{\left(\mathbf{d}^H(\omega).\mathbf{\Phi}_{NN}^{-1}(\omega).\mathbf{d}(\omega)\right).|S(\omega)|^2 + 1}$$

(prove it!)

Multi-Channel Wiener Filter: Implementation

- Implementation with short-time Fourier transform: see p.10
- Implementation with time-domain **linear filtering**:



Multi-Channel Wiener Filter: Implementation

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

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

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

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

Solution is…

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

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

compute during speech+noise periods

compute during noise-only periods

Overview

Continue only after having studied Lecture-6

- **Spectral subtraction for <u>single</u>-micr. noise reduction**
  - Single-microphone noise reduction problem
  - Spectral subtraction basics (=<u>spectral filtering</u>)
  - Features: gain functions, implementation, musical noise,…
  - Iterative Wiener filter based on speech signal model
- **Multi-channel Wiener filter for <u>multi</u>-micr. noise red.**
  - Multi-microphone noise reduction problem
  - Multi-channel Wiener filter (=<u>spectral+spatial</u> filtering)
- **Kalman filter based noise reduction**
  - Kalman filters : See Lecture-6
  - Kalman filters for noise reduction

Kalman filter for Speech Enhancement

- Assume AR model of speech and noise

$$s[k] = \sum_{\bar{n}=1}^{N_s} \alpha_{\bar{n}} s[k-\bar{n}] + \sqrt{g_s}\, u[k]$$

$$n[k] = \sum_{\bar{n}=1}^{N_n} \beta_{\bar{n}} n[k-\bar{n}] + \sqrt{g_n}\, w[k]$$

$u[k]$, $w[k]$ = zero mean, unit variance,white noise

- Equivalent state-space model is…

$$\begin{cases} \mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{v}[k] \\ y[k] = \mathbf{c}^T \mathbf{x}[k] \end{cases}$$

y=microphone signal

Kalman filter for Speech Enhancement

with:

$$\mathbf{x}^T[k] = \begin{bmatrix} s[k-N_s+1] & \cdots & s[k] & n[k-N_n+1] & \cdots & n[k] \end{bmatrix}$$

$$\mathbf{v}[k] = \mathbf{G}.\begin{bmatrix} u[k] & w[k] \end{bmatrix}^T \Rightarrow \mathbf{Q} = \mathbf{G}.\mathbf{G}^T$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_n \end{bmatrix};\ \mathbf{C}^T = \begin{bmatrix} \overbrace{0 \ \cdots \ 0 \ 1}^{M} & \overbrace{0 \ \cdots \ 0 \ 1}^{N} \end{bmatrix}; \mathbf{G} = \begin{bmatrix} \mathbf{g}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_n \end{bmatrix}$$

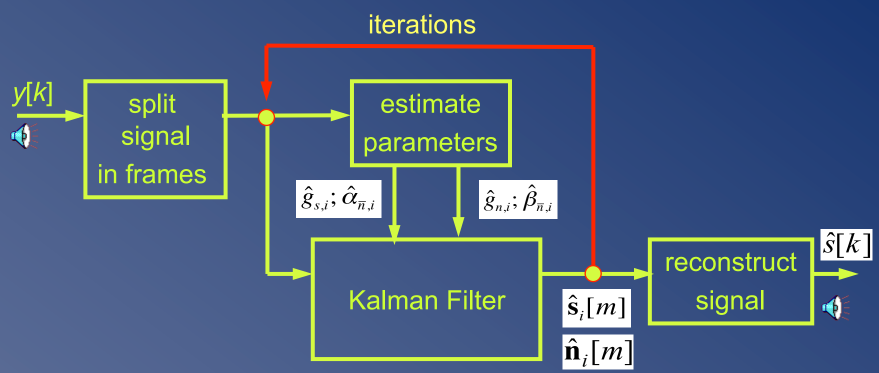
$$\mathbf{A}_s = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ \alpha_{Ns} & \alpha_{Ns-1} & \cdots & \alpha_1 \end{bmatrix};\ \mathbf{A}_n = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ \beta_{N_n} & \beta_{N_n-1} & \cdots & \beta_1 \end{bmatrix}$$

$$\mathbf{g}_s^T = \begin{bmatrix} 0 & \cdots & 0 & \sqrt{g_s} \end{bmatrix};\ \mathbf{g}_n^T = \begin{bmatrix} 0 & \cdots & 0 & \sqrt{g_n} \end{bmatrix};$$

s[k] and n[k] are included in state vector, hence can be estimated by Kalman Filter

Kalman filter for Speech Enhancement

Iterative algorithm (details omitted)



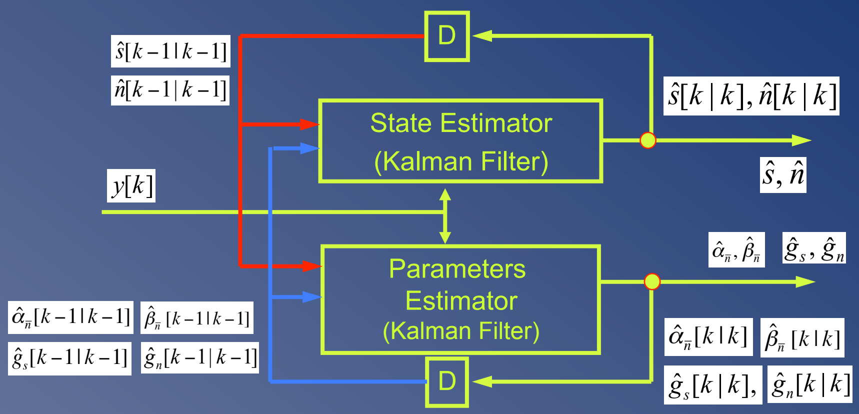
Disadvantages iterative approach:

- complexity
- delay

Kalman filter for Speech Enhancement

Sequential algorithm (details omitted)

iteration index ➡ time index (no iterations)



CONCLUSIONS

- **Single-channel noise reduction**
  - Basic system is spectral subtraction
  - Only spectral filtering, hence can only exploit differences in spectra between noise and speech signal:
    - noise reduction at expense of speech distortion
    - achievable noise reduction may be limited
- **Multi-channel noise reduction**
  - Basic system is MWF,
  - Provides spectral + spatial filtering (links with beamforming!)
- **Iterative Wiener filter & Kalman filtering**
  - Signal model based approach