Digital Audio Signal Processing

DASP

Chapter-2
Single-Channel Noise Reduction

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

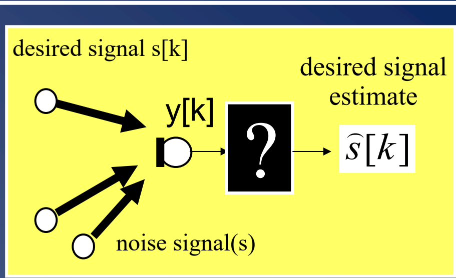
1

Single-Channel Noise Reduction

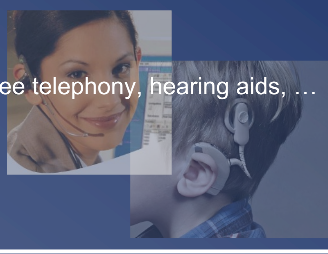
- **Microphone signal** is

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

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



- **Goal:** Estimate s[k] based on y[k]
- **Applications:**
  - Speech enhancement in conferencing, handsfree telephony, hearing aids, …
  - Automatic speech recognition
  - Digital audio restoration

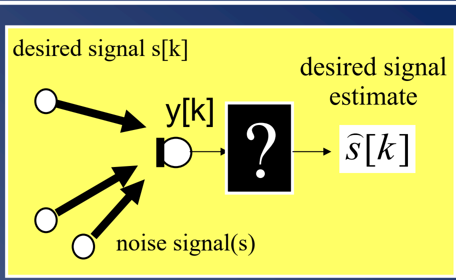


2

Single-Channel Noise Reduction

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

desired signal contribution ($s[k]$) — noise contribution ($n[k]$)



- **Will consider methods that do not rely on a priori information** (e.g. expected noise spectrum)
- **Will consider <u>speech applications</u>:** s[k] = speech signal
  Speech is on/off signal, where speech pauzes can be used to estimate noise spectrum

Overview

- **Frequency Domain Methods / Spectral Subtraction Methods**
  - Spectral subtraction basics
  - Gain functions
  - Realization
  - Musical noise
  - Signal model based spectral subtraction
- **Time Domain Methods**
  - Kalman Filter Based Noise Reduction
  - Subspace Methods

Spectral Subtraction: Basics

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

- Signal segmented into `frames' (e.g. 10..20msec), for each frame a frequency domain representation (=spectrum) is

  $Y_i(\omega) = S_i(\omega) + N_i(\omega)$ *(i-th frame)*

- However, as speech signal is an on/off signal, some frames have **speech+noise**, i.e.

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

  some frames have **noise only**, i.e.

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

- A **speech detection algorithm** (a.k.a. ‘voice activity detection’, VAD) is needed to distinguish between these 2 types of frames (based on energy/dynamic range/statistical properties,…)

5

Spectral Subtraction: Basics

- Definition: $\mu_{\text{noise},i}(\omega) = E\{|N_i(\omega)|\}$, $\sigma^2_{\text{noise},i}(\omega) = E\{|N_i(\omega)|^2\}$

  i.e. noise spectrum expected magnitude & power (frame *i*)

- Assumption: $\mu_{\text{noise}}(\omega) = \mu_{\text{noise},i}(\omega)$, $\sigma^2_{\text{noise}}(\omega) = \sigma^2_{\text{noise},i}(\omega)$

  i.e. noise is (short-term) stationary (+ ergodicity)

- Hence $\mu_{noise}(\omega)$, $\sigma_{noise}^2(\omega)$ can be estimated by averaging over noise-only frames

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

  $$\hat{\sigma}^2_{\text{noise}}(\omega) = \frac{1}{\tilde{N}} \sum_{\text{noise-only frames}} |Y_i(\omega)|^2 \qquad \tilde{N} = \#\ \text{noise-only frames}$$

- For each frame estimate clean speech spectrum $Si(\omega)$ by using noisy speech spectrum $Yi(\omega)$ together with $\mu\hat{}_{noise}(\omega)$, $\sigma\hat{}_{noise}^2(\omega)$

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

  based on `gain function’

  $$G_i(\omega) = f(Y_i(\omega), \hat{\mu}_{noise}(\omega), \hat{\sigma}^2_{\text{noise}})$$

6

Spectral Subtraction: Basics

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

- **PS**: Gain will be real-valued scalar between 0 and 1
  - Only the magnitude of $Y_i(\omega)$ is changed, noisy phase is unchanged
  - Techniques for estimating clean phase also exist (details omitted)
- **PS**: Applying a gain function as $\hat{S}_i(\omega) = G_i(\omega)Y_i(\omega)$
  - can improve signal-to-noise ratio (SNR) of the signal as a whole (i.e. in the time domain) but does not improve the SNR for a particular radial frequency (i.e. speech and noise are equally scaled)
  - hence impact on speech intelligibility is found to be minimal (or non-existing) but 'listening comfort' is said to be improved

For true SNR & speech intelligibility improvement, see multi-channel noise reduction (Chapter 3-4)

Spectral Subtraction: Gain Functions

Examples

- Magnitude Subtraction: $G_i(\omega) = \left[1 - \frac{\hat{\mu}_{\text{noise}}(\omega)}{|Y_i(\omega)|}\right]$
- Spectral Subtraction: $G_i(\omega) = \sqrt{1 - \rho . \frac{\hat{\sigma}^2_{\text{noise}}(\omega)}{|Y_i(\omega)|^2}}$
- Wiener Estimation: $G_i(\omega) = \left[1 - \frac{\hat{\sigma}^2_{\text{noise}}(\omega)}{|Y_i(\omega)|^2}\right]$
- Maximum Likelihood: $G_i(\omega) = \frac{1}{2}\left[1 + \sqrt{1 - \frac{\hat{\sigma}^2_{\text{noise}}(\omega)}{|Y_i(\omega)|^2}}\right]$
- Ephraim-Malah: See next slide (=most frequently used in practice)

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:

$$M[\theta] = e^{-\frac{\theta}{2}}\left[(1+\theta)I_0\left(\frac{\theta}{2}\right)+\theta I_1\left(\frac{\theta}{2}\right)\right]$$

$$\mathrm{SNR}_{\mathrm{post}}(\omega) = \frac{|Y_i(\omega)|^2}{\hat{\sigma}^2_{\mathrm{noise}}(\omega)}$$

modified Bessel functions

$$\mathrm{SNR}_{\mathrm{prio}}(\omega) = (1-\alpha)\max(\mathrm{SNR}_{\mathrm{post}}-1,0)+\alpha\frac{|G_{i-1}(\omega)Y_{i-1}(\omega)|^2}{\hat{\sigma}^2_{\mathrm{noise}}(\omega)}$$

- This corresponds to a **MMSE** (*) estimation of the 'speech spectral amplitude' $|Si(\omega)|$ based on observation $Yi(\omega)$ (estimate equal to **E{** $|Si(\omega)|$ **|** $Yi(\omega)$ **}**) assuming Gaussian a priori distributions for $Si(\omega)$ and $Ni(\omega)$ [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:

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

='instantaneous estimate' (see also p.15)

  - 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

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

  - Solution:

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

←cross-correlation
←auto-correlation

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

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

  - PS: half-wave rectification

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

='instantaneous estimate' (see also p.15)

Overview

- **Frequency Domain Methods / Spectral Subtraction Methods**
  - Spectral subtraction basics
  - Gain functions
  - Realization
  - Musical noise
  - Signal model based spectral subtraction
- **Time Domain Methods**
  - Kalman Filter Based Noise Reduction
  - Subspace Methods

Spectral Subtraction: Realization

Realization requires time-frequency analysis

(see DSP-CIS Chapter 14 on Short-Time Fourier Transform (STFT))

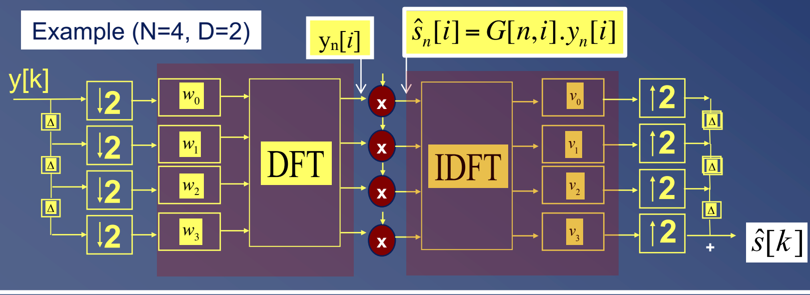
w[k] and v[k] = your favorite window functions

N = window length = DFT/DFT size = frequency resolution

D = window shift

$y_n[i]$ = n-th DFT output at time i = estimate for $Y(\omega_n)$ at time i (i-th frame)

G[n,i] = gain for ω_n at time i (i-th frame)



Spectral Subtraction: Realization

This is a 'Weigthed Overlap-Add' (WOLA) Filter Bank (FB)

(see DSP-CIS Chapter 14)

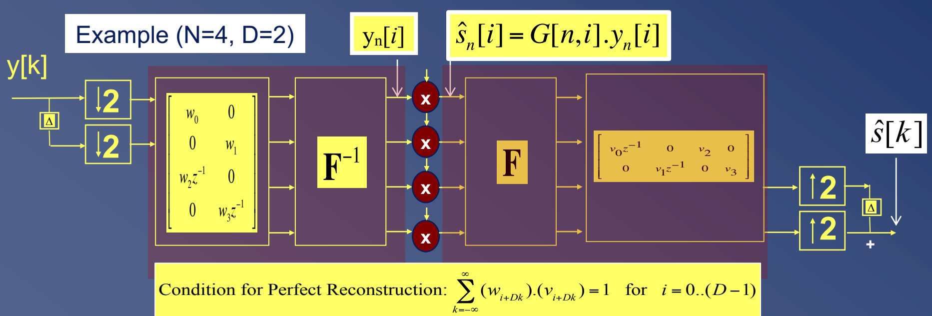
w[k] and v[k] = prototype filters for DFT-modulated analysis/synthesis FB

N = number of filter bank channels

D = downsampling factor (e.g. 2-fold oversampling: D=N/2, 3-fold: D=N/3)

$y_n[i]$ = n-th subband signal

G[n,i] = subband processing in n-th subband



Spectral Subtraction: Musical Noise

- Audio demo:

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

- Artifact: **musical noise**

  <u>**What?**</u>

  Instantaneous/short-time estimates of |*Yi(ω)*| fluctuate randomly 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

<u>Solutions?</u>

– <u>Magnitude averaging</u>: replace *Yi(ω)* in calculation of *Gi(ω)* by a local average over frames

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

average (↑ $G(\omega)$) instantaneous (↑ $Y_i(\omega)$)

– EMSR (see p.9)

– Augment *Gi*(*ω*) with soft-decision VAD:

$$Gi(\omega) \rightarrow P(H_1 \mid Yi(\omega)).\ Gi(\omega)$$

probability that speech is present, given observation

– …

Overview

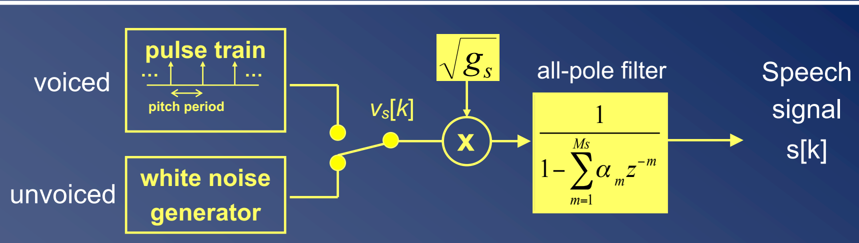
- **Frequency Domain Methods / Spectral Subtraction Methods**
  - Spectral subtraction basics
  - Gain functions
  - Realization
  - Musical noise
  - Signal model based spectral subtraction
- **Time Domain Methods**
  - Kalman Filter Based Noise Reduction
  - Subspace Methods

Signal model based spectral subtraction

- **Basic**:
  Spectral subtraction with improved spectra estimation based on parametric speech signal model

- **Procedure**:
  1. Estimate (initialize) parameters of speech model from noisy signal
  2. Using estimated parameters, perform noise reduction (e.g. Wiener estimation, p.11)
  3. Re-estimate parameters of speech model from the speech signal estimate
  4. Iterate 2 & 3

Signal model based spectral subtraction



frequency domain: $S(\omega)=\frac{\sqrt{g_s}}{1-\sum_{m=1}^{Ms}\alpha_m e^{-j\omega m}}V_s(\omega)$

time domain: $s[k]=\sum_{m=1}^{Ms}\alpha_m s[k-m]+\sqrt{g_s}\,v_s[k]$

$\mathbf{a}=\begin{bmatrix}\alpha_1 & \cdots & \alpha_{Ms}\end{bmatrix}^T$ = linear prediction (LP) parameters

19

Signal model based spectral subtraction

For each frame

1. Estimate $g_{s,i}$ and $\mathbf{a}_i=\begin{bmatrix}\alpha_{1,i} & \cdots & \alpha_{Ms,i}\end{bmatrix}^T$ (=time domain LP)
2. Construct Wiener Filter (p.11)

$$G_i(\omega)=..=\frac{P_{ss,i}(\omega)}{P_{ss,i}(\omega)+P_{nn,i}(\omega)}$$

with:

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

3. Apply $G_i(\omega)$'s and obtain speech estimate (=time domain)

Repeat until some error criterion is satisfied

20

Overview

- **Frequency Domain Methods / Spectral Subtraction Methods**
  - Spectral subtraction basics
  - Gain functions
  - Realization
  - Musical noise
  - Signal model based spectral subtraction
- **Time Domain Methods**
  - Kalman Filter Based Noise Reduction
  - Subspace Methods

Kalman Filter Recap 1/3

State space model of a time-varying discrete-time system

$$\begin{cases} \mathbf{x}[k+1] = \mathbf{A}[k].\mathbf{x}[k] + \mathbf{B}[k].u[k] + \mathbf{v}[k] \\ y[k] = \mathbf{C}[k].\mathbf{x}[k] + \mathbf{D}[k].u[k] + w[k] \end{cases}$$

process noise

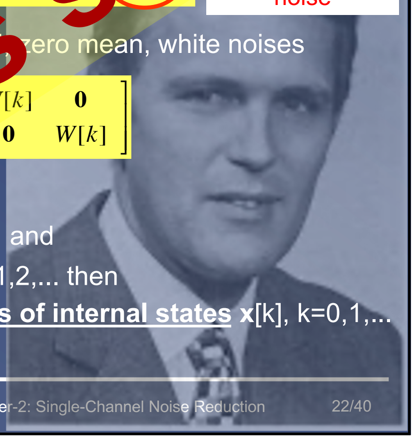
measurement noise

with **v**[*k*] and w[*k*]: mutually uncorrelated, zero mean, white noises

$$E\{\begin{bmatrix} \mathbf{v}[k] \\ w[k] \end{bmatrix} . \begin{bmatrix} \mathbf{v}[k]^H & w[k]^H \end{bmatrix}\} = \begin{bmatrix} \mathbf{V}[k] & \mathbf{0} \\ \mathbf{0} & W[k] \end{bmatrix}$$

Then:

Given **A[k], B[k], C[k], D[k], V[k],** W**[k]** and input/output observations u[k],y[k], k=0,1,2,... then Kalman filter produces MMSE **estimates of internal states** x[k], k=0,1,...



Skip this slide

Kalman Filter Recap 2/3

Definition: $\hat{\mathbf{x}}_{k|l}$ = MMSE-estimate of **x**[k] using all available data up until time *l*

- *`FILTERING'* = estimate $\hat{\mathbf{x}}_{k|k}$
- *`PREDICTION'* = estimate $\hat{\mathbf{x}}_{k|k-n}, n>0$
- *`SMOOTHING'* = estimate $\hat{\mathbf{x}}_{k|k+n}, n>0$

Skip this slide

Kalman Filter Recap 3/3

Initalization:

$$E\{\mathbf{x}_0\} = \hat{\mathbf{x}}_{0|-1}$$
$$E\{\underbrace{(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)}_{\mathbf{e}_0}(\hat{\mathbf{x}}_{0|-1}-\mathbf{x}_0)^T\} = P_{0|-1} = P_{0|-1}^{\frac{1}{2}} P_{0|-1}^{\frac{T}{2}}$$ = error covariance matrix

➔ **'Conventional' Kalman Filter:** For k=0,1,2,..

Given $\hat{\mathbf{x}}_{k|k-1}$ and corresponding error covariance matrix $\mathbf{P}_{k|k-1}$:

Step 1: Measurement Update

(produces 'filtered' estimate)

(compare to standard RLS!)

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}C_k^T(W_k + C_kP_{k|k-1}C_k^T)^{-1}C_kP_{k|k-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + P_{k|k}C_k^TW_k^{-1}\cdot(y_k - C_k\hat{\mathbf{x}}_{k|k-1} - D_ku_k)$$

Step 2: Time Update

(produces '1-step prediction')

$$P_{k+1|k} = A_kP_{k|k}A_k^T + V_k$$
$$\hat{\mathbf{x}}_{k+1|k} = A_k\cdot\hat{\mathbf{x}}_{k|k} + B_k\cdot u_k$$

➔ **Better: 'Square Root' Algorithm** (see DSP-CIS Chapter 14)

Skip this slide

PS: shorthand notation used here, i.e. $\mathbf{x}_k$, u_k, A_k.. instead of **x**[k], u[k],A[k],..

Kalman Filter Based Noise Reduction

- Assume AR model of speech and noise

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

$$n[k] = \sum_{m=1}^{Mn} \beta_m n[k-m] + \sqrt{g_n}\, v_n[k]$$

$v_s[k]$, $v_n[k]$ = zero mean, unit variance,white noise

- Equivalent state-space model is…

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

y[k] = microphone signal

Skip this slide

25

Kalman Filter Based Noise Reduction

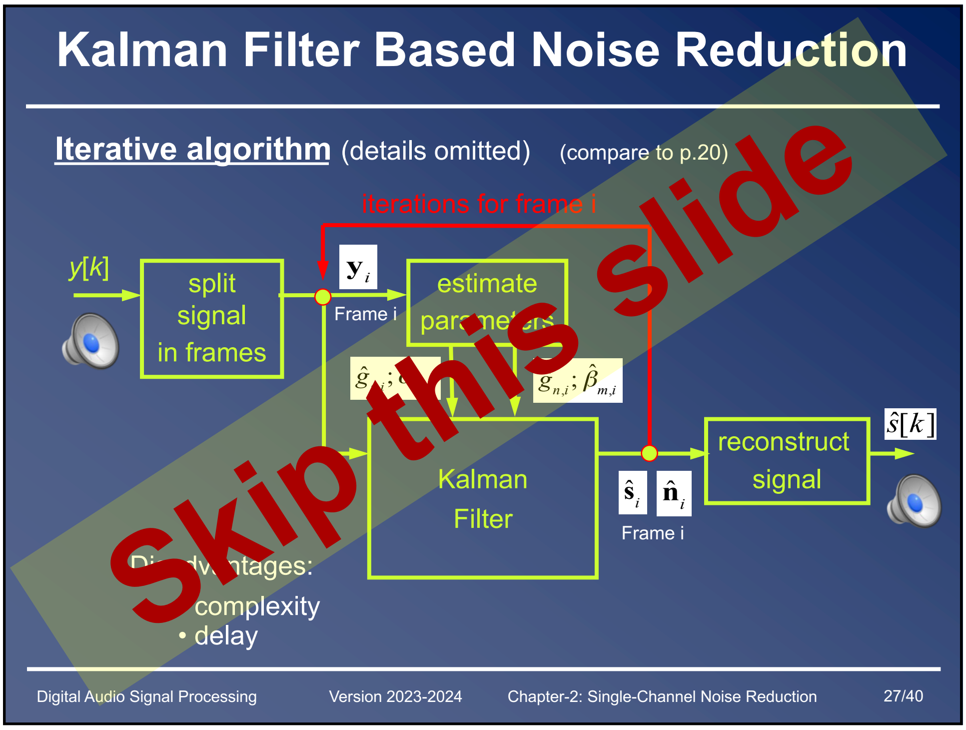
$$\underbrace{\begin{bmatrix} s[k-M_s+2] \\ \vdots \\ s[k] \\ s[k+1] \\ n[k-M_s+2] \\ \vdots \\ n[k] \\ n[k+1] \end{bmatrix}}_{\mathbf{x}[k+1]} = \underbrace{\begin{bmatrix} 0 & 1 & \cdots & 0 & & & & \\ \vdots & 0 & \vdots & 0 & & & & \\ 0 & \cdots & 0 & 1 & & & & \\ \alpha_{Ms} & \alpha_{Ms-1} & \cdots & \alpha_1 & & & & \\ & & & & 0 & 1 & \cdots & 0 \\ & & & & \vdots & 0 & \vdots & 0 \\ & & & & 0 & \cdots & 0 & 1 \\ & & & & \beta_{Mn} & \beta_{Mn-1} & \cdots & \beta_1 \end{bmatrix}}_{\mathbf{A}[k]} \cdot \underbrace{\begin{bmatrix} s[k-M_s+1] \\ \vdots \\ s[k-1] \\ s[k] \\ n[k-M_s+1] \\ \vdots \\ n[k-1] \\ n[k] \end{bmatrix}}_{\mathbf{x}[k]} + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{g_s}.v_s[k] \\ 0 \\ \vdots \\ 0 \\ \sqrt{g_n}.v_n[k] \end{bmatrix}}_{\mathbf{v}[k]}$$

$$y[k] = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\mathbf{C}[k]} . \mathbf{x}[k]$$

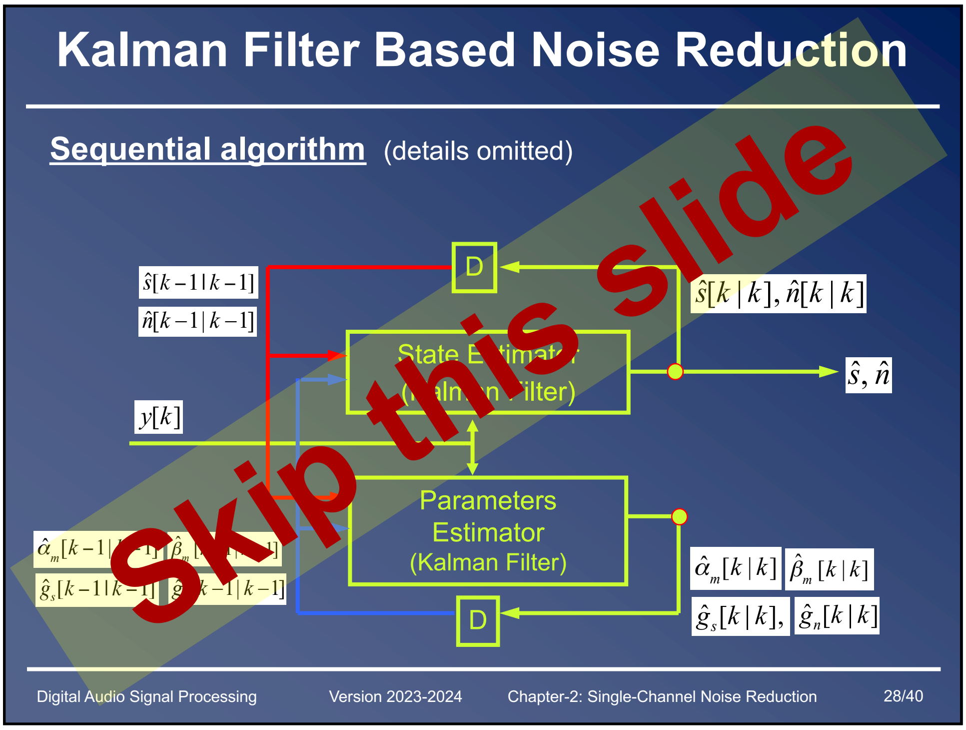
s[k] and n[k] are included in state vector, hence can be estimated by a Kalman Filter (assuming α's, β's, … are known)

Skip this slide

26


Kalman Filter Based Noise Reduction
Iterative algorithm (details omitted)
(compare to p.20)
iterations for frame i
y[k]
split
signal
in frames
Frame i
estimate
parameters
Kalman
Filter
Frame i
reconstruct
signal
• complexity
• delay
Skip this slide
Digital Audio Signal Processing
Version 2023-2024
Chapter-2: Single-Channel Noise Reduction
27/40

27


Kalman Filter Based Noise Reduction
Sequential algorithm (details omitted)
D
State Estimator
(Kalman Filter)
Parameters
Estimator
(Kalman Filter)
D
Skip this slide
Digital Audio Signal Processing
Version 2023-2024
Chapter-2: Single-Channel Noise Reduction
28/40

28

Overview

- **Frequency Domain Methods / Spectral Subtraction Methods**
  - Spectral subtraction basics
  - Gain functions
  - Realization
  - Musical noise
  - Signal model based spectral subtraction
- **Time Domain Methods**
  - Kalman Filter Based Noise Reduction
  - Subspace Methods

Subspace Methods

- Signal model: $y[k] = s[k] + n[k], \quad k = 1,\ldots,N$
- Construct Toeplitz matrix (LxM, L=N-M+1) :

$$\mathbf{Y} = \begin{bmatrix} y[M] & y[M-1] & y[M-2] & \cdots & y[1] \\ y[M+1] & y[M] & y[M-1] & \cdots & y[2] \\ y[M+2] & y[M+1] & y[M] & \cdots & y[3] \\ y[M+3] & y[M+2] & y[M+1] & \cdots & y[4] \\ \vdots & \vdots & \vdots & & \vdots \\ y[N-2] & y[N-3] & y[N-4] & \cdots & y[N-M-1] \\ y[N-1] & y[N-2] & y[N-3] & \cdots & y[N-M] \\ y[N] & y[N-1] & y[N-2] & \cdots & y[N-M+1] \end{bmatrix}$$

Hence: $\mathbf{Y} = \mathbf{S} + \mathbf{N}$ (with **S** and **N** similarly constructed)

Subspace Methods

$$\mathbf{Y} = \mathbf{S} + \mathbf{N}$$

Assumptions: Consider L>>M ('tall thin' matrices)

– Clean signal matrix **S** is rank deficient
  (e.g. 'sum of sinusoids' speech model)

  rank(**S**) = K < M

– Clean signal matrix **S** is orthogonal (≈uncorrelated) to noise

  $(1/L) \,.\, \mathbf{S}^T \mathbf{N} = \mathbf{0}$

– White noise (see p.38 for coloured noise)

  $(1/L) \,.\, \mathbf{N}^T \mathbf{N} = \sigma^2_{noise} \,.\mathbf{I}_M$

Assumptions better satisfied as L➔∞

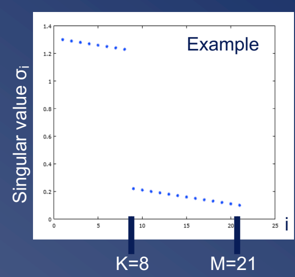
Subspace Methods

Tool: Singular value decomposition (SVD) of Y

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

U is LxM orthogonal matrix $U^T.U=I_{MxM}$
  Columns of U are 'left singular vectors'

V is MxM orthogonal matrix $V^T.V=I_{MxM}$
  Columns of V are 'right singular vectors'



Σ is MxM diagonal matrix
  Diagonal elements are 'singular values' σ_i
  Σ_1 has K largest σ_i corresponding to 'signal+noise subspace'
  Σ_2 has M-K smallest $\sigma_i \approx \sigma_{noise}$ corresponding to 'noise subspace'

Subspace Methods

$$\mathbf{Y} = \mathbf{S} + \mathbf{N}$$

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

Procedure

- **Step-1 : From SVD of Y, find estimate $\hat{\mathbf{S}}$ of S**
  - Least squares estimate
  - Minimum variance estimate
- **Step-2 : Construct $\hat{s}[k], k = 1, \ldots, N$ from $\hat{\mathbf{S}}$**

Subspace Methods

- **Step-1 (v1.0) : Least squares estimator $\mathrm{Y_{LS}}$**
  - Goal: approximate **Y** by a matrix $\mathbf{Y}_{\mathrm{LS}}$ of rank *K*

$$\min_{\mathrm{rank}(\mathbf{Y}_{\mathrm{LS}}=K)} \left\| \mathbf{Y} - \mathbf{Y}_{\mathrm{LS}} \right\|_F^2$$

  - Solution is obtained by setting *M-K* smallest singular values to zero (`truncation')

$$\hat{\mathbf{S}} = \mathbf{Y}_{\mathrm{LS}} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} = \mathbf{U}_1\Sigma_1\mathbf{V}_1$$

  - Problem: proper determination of order *K*

Subspace Methods

- **Step-1 (v2.0): Minimum variance estimator Y_{MV}**
  – Find a matrix **T** that minimizes

$$\min_{\mathbf{T}} \|\mathbf{Y}\mathbf{T}-\mathbf{S}\|_F^2$$

(assume S is given, then i-th column of T is optimal linear filter for i-th column of S)

  – Then:

$$\begin{aligned}\hat{\mathbf{S}} = \mathbf{Y}_{MV} &= \mathbf{Y}.\overbrace{(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{S}}^{\mathbf{T}_{MV}} \\ &= \mathbf{Y}\cdot(\mathbf{V}\Sigma^2\mathbf{V}^T)^{-1}\cdot(\underbrace{\mathbf{Y}^T\mathbf{Y}}_{\mathbf{V}\Sigma^2\mathbf{V}^T}-\underbrace{\mathbf{N}^T\mathbf{N}}_{\sigma^2_{noise}\mathbf{I}}) \\ &= \underbrace{\mathbf{Y}}_{\mathbf{U}\Sigma\mathbf{V}^T}\cdot\mathbf{V}\cdot diag\{\frac{\sigma_i^2-\sigma^2_{noise}}{\sigma_i^2}\}\cdot\mathbf{V}^T \quad (*) \\ &= \mathbf{U}\cdot diag\{\frac{\sigma_i^2-\sigma^2_{noise}}{\sigma_i}\}\cdot\mathbf{V}^T\end{aligned}$$

PS : rank-deficiency condition not needed here

PS : σ^2_{noise} has to be known (e.g. estimated during noise-only)

Subspace Methods

- **Step-2 : Signal Reconstruction**

Goal: Given $\hat{\mathbf{S}}$, construct $\hat{s}[k], k=1,\ldots,N$

Problem: $\hat{\mathbf{S}}$ does not have Toeplitz structure

Solution:

Restore Toeplitz structure by arithmetically averaging every diagonal of the matrix

Hence:

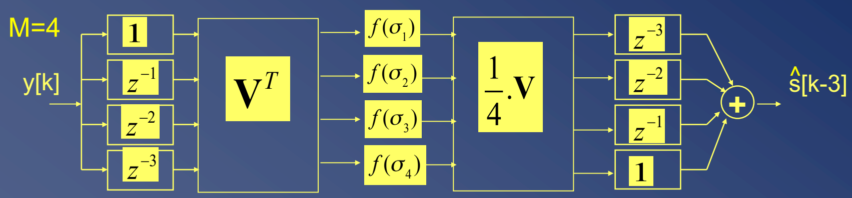
$$\hat{s}[i]=\frac{1}{\beta-\alpha+1}\sum_{k=\alpha}^{\beta}\hat{\mathbf{S}}[k+i-M,k]$$

$\alpha=\max(1,M+1-i)$ (summation starts in column α)

$\beta=\min(M,M+L-i)$ (summation ends in column β)

Subspace Methods

- **<u>PS</u>: Link with spectral subtraction?**
  - Compare (*) p.35 with gain function on p.11 (Wiener)
  - Check that p.35-36 (for samples k=M,…,N-M+1) corresponds to



which represents an analysis/synthesis filter bank with subband processing (be it without up/downsampling)

  - **Interpretation:** Subspace method provides a 'spectral subtraction' in signal dependent filter bank (analysis filter bank V^T, synthesis filter bank V) as an alternative to signal independent STFT/WOLA filter bank

Subspace Methods

- **Audio Demo**

Subspace Methods

- **PS: What if coloured noise** (1/L) .$\mathbf{N}^T\mathbf{N} \neq \sigma^2_{noise}\mathbf{I}_M$ **?**
  - Estimate $\mathbf{N}^T\mathbf{N}$ during noise-only periods
  - Similar procedure, now including 'pre-whitening' of Toeplitz matrix prior to SVD and 'de-whitening' step after estimation of $\hat{S}$
  - Computational scheme can use 'generalized SVD' instead of SVD (with 'built-in' pre-/de-whitening)
  - Easy, but details omitted...

Conclusion

- Many single-channel noise reduction procedures available
- Performance never spectacular
- Seek performance improvement with multi-channel processing (=Chapter 3-4)

PS: Recently strong shift towards **machine learning** based single-channel noise reduction (not covered)