CS578- Speech Signal Processing

Lecture 8: Speech Enhancement

Yannis Stylianou

University of Crete, Computer Science Dept., Multimedia Informatics Lab
yannis@csd.uoc.gr

Univ. of Crete

Outline

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Introduction

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Introduction

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Introduction

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

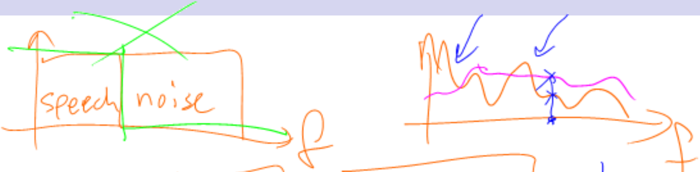
Introduction

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Introduction

- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Introduction



- Types of noise: Additive and Convolutional
- Speech distortion
- Enhancement foundations:
  - Spectral Subtraction,
  - Cepstral Mean Subtraction
  - Wiener Filter
- Enhanced speech judgements: by humans, by machines

Outline

Additive Noise

- A discrete-time noisy sequence:

$$y[n] = x[n] + b[n]$$

- with power spectra:

$$S_y(\omega) = S_x(\omega) + S_b(\omega)$$

- Working with STFT:

$$y_{pL}[n] = w[pL - n](x[n] + b[n])$$

- in the frequency domain:

$$Y(pL, \omega) = X(pL, \omega) + B(pL, \omega)$$

- Our target:

$$\hat{X}(pL, \omega) = |X(pL, \omega)| e^{j \measuredangle Y(pL, \omega)}$$

Additive Noise

- A discrete-time noisy sequence:

$$y[n] = x[n] + b[n]$$

- with power spectra:

$$S_y(\omega) = S_x(\omega) + S_b(\omega)$$

- Working with STFT:

$$y_{pL}[n] = w[pL - n](x[n] + b[n])$$

- in the frequency domain:

$$Y(pL, \omega) = X(pL, \omega) + B(pL, \omega)$$

- Our target:

$$\hat{X}(pL, \omega) = |X(pL, \omega)| e^{j\measuredangle Y(pL,\omega)}$$

Additive Noise

- A discrete-time noisy sequence:

$$y[n] = x[n] + b[n]$$

- with power spectra:

$$S_y(\omega) = S_x(\omega) + S_b(\omega)$$

- Working with STFT:

$$y_{pL}[n] = w[pL - n](x[n] + b[n])$$

- in the frequency domain:

$$Y(pL, \omega) = X(pL, \omega) + B(pL, \omega)$$

- Our target:

$$\hat{X}(pL, \omega) = |X(pL, \omega)| e^{j\measuredangle Y(pL, \omega)}$$

Additive Noise

- A discrete-time noisy sequence:

$$y[n] = x[n] + b[n]$$

- with power spectra:

$$S_y(\omega) = S_x(\omega) + S_b(\omega)$$

- Working with STFT:

$$y_{pL}[n] = w[pL - n](x[n] + b[n])$$

- in the frequency domain:

$$Y(pL, \omega) = X(pL, \omega) + B(pL, \omega)$$

- Our target:

$$\hat{X}(pL, \omega) = |X(pL, \omega)| e^{j\measuredangle Y(pL, \omega)}$$

Additive Noise

- A discrete-time noisy sequence:

$$y[n] = x[n] + b[n]$$

- with power spectra:

$$S_y(\omega) = S_x(\omega) + S_b(\omega)$$

- Working with STFT:

$$y_{pL}[n] = w[pL - n](x[n] + b[n])$$

- in the frequency domain:

$$Y(pL, \omega) = X(pL, \omega) + B(pL, \omega)$$

- Our target:

$$\hat{X}(pL, \omega) = |X(pL, \omega)| e^{j \measuredangle Y(pL, \omega)}$$

Convolutional Distortion

- A discrete-time convolutional distorted sequence:

$$y[n] = x[n] \star g[n]$$

  where $g[n]$ is the impulse response of a linear time-invariant distortion filter.

- Working with a frame-by-frame analysis:

$$y_{pL}[n] = w[pL - n](x[n] \star g[n])$$

- In the frequency domain, we can show that:

$$Y(pL, \omega) = X(pL, \omega)G(\omega)$$

Convolutional Distortion

- A discrete-time convolutional distorted sequence:

$$y[n] = x[n] \star g[n]$$

where $g[n]$ is the impulse response of a linear time-invariant distortion filter.

- Working with a frame-by-frame analysis:

$$y_{pL}[n] = w[pL - n](x[n] \star g[n])$$

- In the frequency domain, we can show that:

$$Y(pL, \omega) = X(pL, \omega)G(\omega)$$

Convolutional Distortion

- A discrete-time convolutional distorted sequence:

$$y[n] = x[n] \star g[n]$$

where $g[n]$ is the impulse response of a linear time-invariant distortion filter.

- Working with a frame-by-frame analysis:

$$y_{pL}[n] = w[pL - n](x[n] \star g[n])$$

- In the frequency domain, we can show that:

$$Y(pL, \omega) = X(pL, \omega)G(\omega)$$

Standard Spectral Subtraction

Assuming that noise and target (object) signal are uncorrelated:

- Estimate of object's short-time squared spectral magnitude

$$
\begin{aligned}
|\hat{X}(pL,\omega)|^2 &= |Y(pL,\omega)|^2 - \hat{S}_b(\omega) \quad \text{if } |Y(pL,\omega)|^2 - \hat{S}_b(\omega) \geq 0 \\
&= 0 \quad \text{otherwise}
\end{aligned}
$$

- STFT estimate:

$$\hat{X}(pL,\omega) = |\hat{X}(pL,\omega)|e^{j\measuredangle Y(pL,\omega)}$$

Spectral Subtraction as a filtering operation

- We can show:

$$
\begin{aligned}
|\hat{X}(pL,\omega)|^2 &= |Y(pL,\omega)|^2 - \hat{S}_b(\omega) \\
&\approx |Y(pL,\omega)|^2 \left[1 + \frac{1}{R(pL,\omega)}\right]^{-1}
\end{aligned}
$$

where

$$
R(pL,\omega) = \frac{|X(pL,\omega)|^2}{\hat{S}_b(\omega)}
$$

- Suppression filter frequency response

$$
H_s(pL,\omega) = \left[1 + \frac{1}{R(pL,\omega)}\right]^{-1/2}
$$

Spectral Subtraction as a filtering operation

- We can show:

$$
\begin{aligned}
|\hat{X}(pL,\omega)|^2 &= |Y(pL,\omega)|^2 - \hat{S}_b(\omega) \\
&\approx |Y(pL,\omega)|^2 \left[1 + \frac{1}{R(pL,\omega)}\right]^{-1}
\end{aligned}
$$

where

$$R(pL,\omega) = \frac{|X(pL,\omega)|^2}{\hat{S}_b(\omega)}$$

- Suppression filter frequency response

$$H_s(pL,\omega) = \left[1 + \frac{1}{R(pL,\omega)}\right]^{-1/2}$$

The role of the analysis window

Let $x[n] = A\cos(\omega_0 n)$ be in a stationary white noise $b[n]$ of variance σ^2 and $w[n]$ be a short-time window. Then:

- Average short-time signal power at ω_0:

$$\hat{S}_x(pL,\omega_0) = E[|X(pL,\omega_0)|^2] \approx \frac{A^2}{4}\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2$$

- Average power of the windowed noise

$$\hat{S}_b(pL,\omega) = E[|B(pL,\omega)|^2] = \sigma^2 \sum_{n=-\infty}^{\infty} w^2[n]$$

- Ratio at ω_0:

$$\frac{E[|Y(pL,\omega)|^2]}{\hat{S}_b(pL,\omega_0)} = 1 + \frac{A^2/4}{[\sigma^2\Delta_w]}$$

where

$$\Delta_w = \frac{\sum_{n=-\infty}^{\infty} w^2[n]}{\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2}$$

The role of the analysis window

Let $x[n] = A\cos(\omega_0 n)$ be in a stationary white noise $b[n]$ of variance σ^2 and $w[n]$ be a short-time window. Then:

- Average short-time signal power at ω_0:

$$\hat{S}_x(pL,\omega_0) = E[|X(pL,\omega_0)|^2] \approx \frac{A^2}{4}\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2$$

- Average power of the windowed noise

$$\hat{S}_b(pL,\omega) = E[|B(pL,\omega)|^2] = \sigma^2 \sum_{n=-\infty}^{\infty} w^2[n]$$

- Ratio at ω_0:

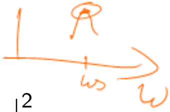
$$\frac{E[|Y(pL,\omega)|^2]}{\hat{S}_b(pL,\omega_0)} = 1 + \frac{A^2/4}{[\sigma^2\Delta_w]}$$

where

$$\Delta_w = \frac{\sum_{n=-\infty}^{\infty} w^2[n]}{\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2}$$

The role of the analysis window

Let $x[n] = A\cos(\omega_0 n)$ be in a stationary white noise $b[n]$ of variance σ^2 and $w[n]$ be a short-time window. Then:



- Average short-time signal power at ω_0:

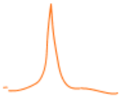
$$\hat{S}_x(pL,\omega_0) = E[|X(pL,\omega_0)|^2] \approx \frac{A^2}{4}\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2$$

- Average power of the windowed noise

$$\hat{S}_b(pL,\omega) = E[|B(pL,\omega)|^2] = \sigma^2 \sum_{n=-\infty}^{\infty} w^2[n]$$

- Ratio at ω_0:

$$\frac{E[|Y(pL,\omega)|^2]}{\hat{S}_b(pL,\omega_0)} = 1 + \frac{A^2/4}{[\sigma^2 \Delta_w]}$$



where

$$\Delta_w = \frac{\sum_{n=-\infty}^{\infty} w^2[n]}{\left|\sum_{n=-\infty}^{\infty} w[n]\right|^2}$$

Cepstral Mean Subtraction

Let $y[n] = x[n] \star g[n]$. Then:

- Logarithm of the STFT of $y[n]$:

$$Y(pL, \omega) \approx \log\left[X(pL, \omega)\right] + \log\left[G(\omega)\right]$$

- Cepstrum:

$$\begin{aligned} \hat{y}[n, \omega] &\approx F_p^{-1}(\log\left[X(pL, \omega)\right]) + F_p^{-1}(\log\left[G(\omega)\right]) \\ &= \hat{x}[n, \omega] + \hat{g}[0, \omega]\delta[n] \end{aligned}$$

- Cepstral filter:

$$\hat{x}[n, \omega] \approx l[n]\hat{y}[n, \omega]$$

where $l[n] = u[n-1]$

Cepstral Mean Subtraction

Let $y[n] = x[n] \star g[n]$. Then:

- Logarithm of the STFT of $y[n]$:

$$Y(pL,\omega) \approx \log\left[X(pL,\omega)\right] + \log\left[G(\omega)\right]$$

- Cepstrum:

$$\begin{aligned}\hat{y}[n,\omega] &\approx F_p^{-1}(\log\left[X(pL,\omega)\right]) + F_p^{-1}(\log\left[G(\omega)\right]) \\ &= \hat{x}[n,\omega] + \hat{g}[0,\omega]\delta[n]\end{aligned}$$

- Cepstral filter:

$$\hat{x}[n,\omega] \approx l[n]\hat{y}[n,\omega]$$

where $l[n] = u[n-1]$

Cepstral Mean Subtraction

Let $y[n] = x[n] \star g[n]$. Then:

- Logarithm of the STFT of $y[n]$:

$$Y(pL, \omega) \approx \log\left[X(pL, \omega)\right] + \log\left[G(\omega)\right]$$

- Cepstrum:

$$\begin{aligned} \hat{y}[n, \omega] &\approx F_p^{-1}(\log\left[X(pL, \omega)\right]) + F_p^{-1}(\log\left[G(\omega)\right]) \\ &= \hat{x}[n, \omega] + \hat{g}[0, \omega]\delta[n] \end{aligned}$$

- Cepstral filter:

$$\hat{x}[n, \omega] \approx l[n]\hat{y}[n, \omega]$$

where $l[n] = u[n-1]$

Outline

1. Introduction
2. Preliminaries
   - Problem Formulation
   - Spectral Subtraction
   - Cepstral Mean Subtraction
3. Wiener Filtering
   - Estimating the Object Spectrum
   - Adaptive smoothing
   - Application to Speech
   - Optimal Spectral Magnitude Estimation
   - Binaural Representation
4. Model-Based Processing
5. Auditory Masking
   - Frequency-Domain Masking Principles
   - Calculation of the Masking Threshold
   - Exploiting Frequency Masking in Noise Reduction
6. Acknowledgments

Wiener Filtering

- Stochastic optimization:
  if $y[n] = x[n] + b[n]$, find $h[n]$ such that $\hat{x}[n] = y[n] \star h[n]$ minimizes
$$e = E[|\hat{x}[n] - x[n]|^2]$$
- Frequency domain solution (*Wiener filter*):
$$H_w = \frac{S_x(\omega)}{S_x(\omega) + S_b(\omega)}$$
- Time-varying Wiener filter:
$$H_w(pL,\omega) = \frac{\hat{S}_x(pL,\omega)}{\hat{S}_x(pL,\omega) + \hat{S}_b(\omega)}$$
- Or
$$H_w(pL,\omega) = \left[1 + \frac{1}{R(pL,\omega)}\right]^{-1}$$
  where
$$R(pL,\omega) = \frac{\hat{S}_x(pL,\omega)}{\hat{S}_b(\omega)}$$

Wiener Filtering

- Stochastic optimization:
  if $y[n] = x[n] + b[n]$, find $h[n]$ such that $\hat{x}[n] = y[n] \star h[n]$ minimizes
  $$e = E[|\hat{x}[n] - x[n]|^2]$$
- Frequency domain solution (*Wiener filter*):
  $$H_w = \frac{S_x(\omega)}{S_x(\omega) + S_b(\omega)}$$
- Time-varying Wiener filter:
  $$H_w(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_x(pL, \omega) + \hat{S}_b(\omega)}$$
- Or
  $$H_w(pL, \omega) = \left[1 + \frac{1}{R(pL, \omega)}\right]^{-1}$$
  where
  $$R(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_b(\omega)}$$

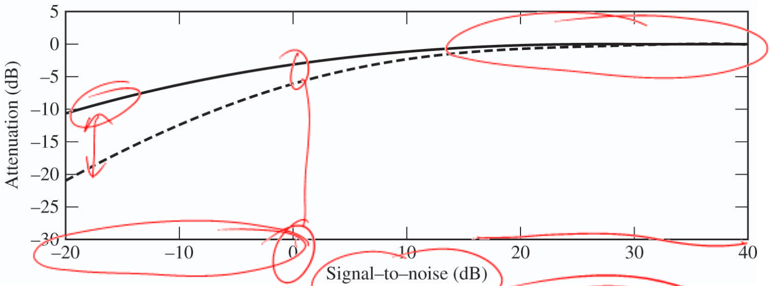
Wiener Filtering

- Stochastic optimization:
  if $y[n] = x[n] + b[n]$, find $h[n]$ such that $\hat{x}[n] = y[n] \star h[n]$ minimizes
  $$e = E[|\hat{x}[n] - x[n]|^2]$$
- Frequency domain solution (*Wiener filter*):
  $$H_w = \frac{S_x(\omega)}{S_x(\omega) + S_b(\omega)}$$
- Time-varying Wiener filter:
  $$H_w(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_x(pL, \omega) + \hat{S}_b(\omega)}$$
- Or
  $$H_w(pL, \omega) = \left[1 + \frac{1}{R(pL, \omega)}\right]^{-1}$$
  where
  $$R(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_b(\omega)}$$

Wiener Filtering

- Stochastic optimization:
  if $y[n] = x[n] + b[n]$, find $h[n]$ such that $\hat{x}[n] = y[n] \star h[n]$ minimizes
  $$e = E[|\hat{x}[n] - x[n]|^2]$$
- Frequency domain solution (*Wiener filter*):
  $$H_w = \frac{S_x(\omega)}{S_x(\omega) + S_b(\omega)}$$
- Time-varying Wiener filter:
  $$H_w(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_x(pL, \omega) + \hat{S}_b(\omega)}$$
- Or
  $$H_w(pL, \omega) = \left[1 + \frac{1}{R(pL, \omega)}\right]^{-1}$$
  where
  $$R(pL, \omega) = \frac{\hat{S}_x(pL, \omega)}{\hat{S}_b(\omega)}$$

Comparing the two suppression filters



Solid line: Spectral Subtraction. Dashed-line: Wiener filter

A BASIC APPROACH

- We assume that the Wiener filter of $p-1$ frame is known, then:

$$\hat{X}(pL,\omega) = Y(pL,\omega)H_w((p-1)L,\omega)$$

- Updating the Wiener filter:

$$H_w(pL,\omega) = \frac{|\hat{X}(pL,\omega)|^2}{|\hat{X}(pL,\omega)|^2 + \hat{S}_b(\omega)}$$

- Smooth power spectrum:

$$\tilde{S}_x(pL,\omega) = \tau\tilde{S}_x((p-1)L,\omega) + (1-\tau)\hat{S}_x(pL,\omega)$$

where $\hat{S}_x(pL,\omega) = |\hat{X}(pL,\omega)|^2$

- Initialization: spectral subtraction

Adaptive smoothing

- Wiener filter estimator adapts to the "degree of stationarity" of the measured signal.
- A measure of the degree of stationarity

$$\Delta Y(pL) = h_{\Delta}[p] \star \left[\frac{1}{\pi} \int_0^{\pi} |Y(pL,\omega) - Y((p-1)L,\omega)|^2 d\omega \right]^{1/2}$$

- Time varying smoothing constant:

$$\tau(p) = Q[1 - 2(\Delta Y(pL) - \bar{\Delta Y})]$$

where

$$Q(x) = \begin{cases} x, & 0 \le x \le 1 \\ 0, & x < 0 \\ 1, & x > 1 \end{cases}$$

- Smooth object spectrum:

$$\tilde{S}_x(pL,\omega) = \tau(p)\tilde{S}_x((p-1)L,\omega) + [1-\tau(p)]\hat{S}_x(pL,\omega)$$

Adaptive smoothing

- Wiener filter estimator adapts to the "degree of stationarity" of the measured signal.
- A measure of the degree of stationarity

$$\Delta Y(pL) = h_{\Delta}[p] \star \left[\frac{1}{\pi} \int_0^{\pi} |Y(pL,\omega) - Y((p-1)L,\omega)|^2 d\omega \right]^{1/2}$$

- Time varying smoothing constant:

$$\tau(p) = Q[1 - 2(\Delta Y(pL) - \overline{\Delta Y})]$$

where

$$Q(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 0, & x < 0 \\ 1, & x > 1 \end{cases}$$

- Smooth object spectrum:

$$\tilde{S}_x(pL,\omega) = \tau(p)\tilde{S}_x((p-1)L,\omega) + [1 - \tau(p)]\hat{S}_x(pL,\omega)$$

Adaptive smoothing

- Wiener filter estimator adapts to the "degree of stationarity" of the measured signal.
- A measure of the degree of stationarity

$$\Delta Y(pL) = h_\Delta[p] \star \left[\frac{1}{\pi} \int_0^\pi |Y(pL,\omega) - Y((p-1)L,\omega)|^2 d\omega \right]^{1/2}$$

- Time varying smoothing constant:

$$\tau(p) = Q[1 - 2(\Delta Y(pL) - \bar{\Delta Y})]$$

where

$$Q(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 0, & x < 0 \\ 1, & x > 1 \end{cases}$$

- Smooth object spectrum:

$$\tilde{S}_x(pL,\omega) = \tau(p)\tilde{S}_x((p-1)L,\omega) + [1 - \tau(p)]\hat{S}_x(pL,\omega)$$

Adaptive smoothing

- Wiener filter estimator adapts to the "degree of stationarity" of the measured signal.
- A measure of the degree of stationarity

$$\Delta Y(pL) = h_{\Delta}[p] \star \left[\frac{1}{\pi} \int_0^{\pi} |Y(pL,\omega) - Y((p-1)L,\omega)|^2 d\omega \right]^{1/2}$$

- Time varying smoothing constant:

$$\tau(p) = Q[1 - 2(\Delta Y(pL) - \overline{\Delta Y})]$$

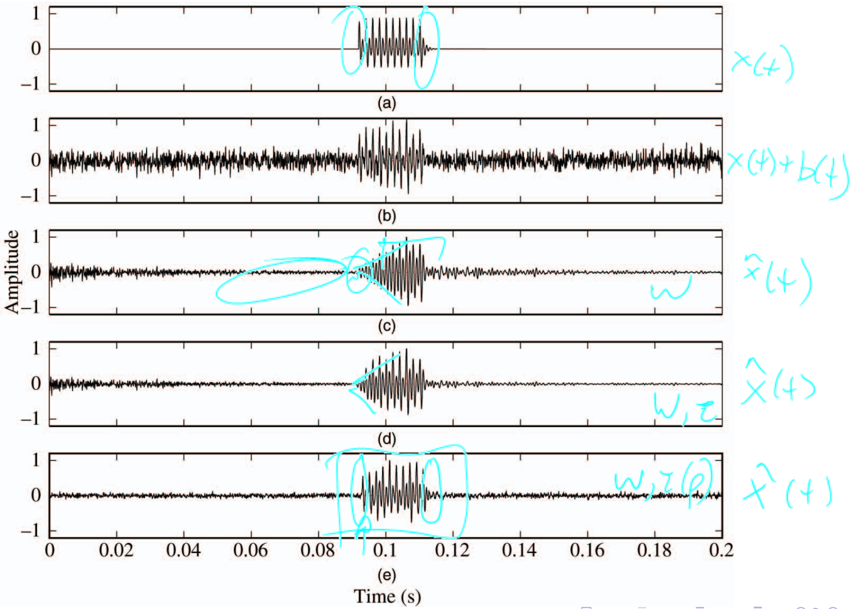
where

$$Q(x) = \begin{cases} x, & 0 \le x \le 1 \\ 0, & x < 0 \\ 1, & x > 1 \end{cases}$$

- Smooth object spectrum:

$$\tilde{S}_x(pL,\omega) = \tau(p)\tilde{S}_x((p-1)L,\omega) + [1 - \tau(p)]\hat{S}_x(pL,\omega)$$

Example of enhancement

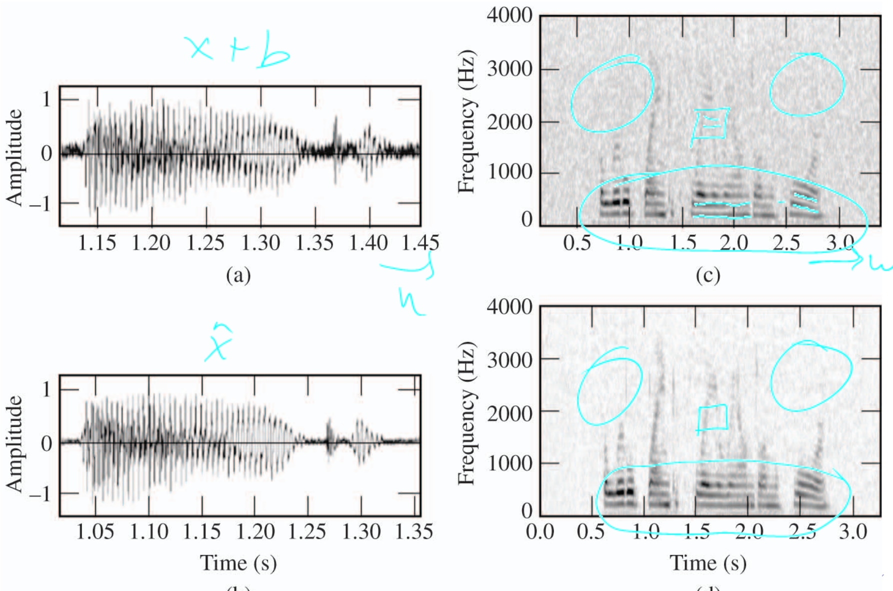


Application to Speech

Satisfying enhanced speech quality with Wiener filter is obtained if:

- Window: triangular
- Frame length: 4ms
- Frame interval (rate): 1ms
- OLA synthesis

Example of enhancement in speech



Minimum mean-Square Error

If

$$y[n] = x[n] + b[n]$$

compute the expected value of:

$$E\{|X(pL,\omega)| \mid y[n]\}$$

(Ephraim and Malah, 1984)

Suppression Filter

- Suppression Filter of Ephraim and Malah

$$H_s(pL,\omega) = \sqrt{\frac{\pi}{2}}\sqrt{\left(\frac{1}{1+\gamma_{po}(pL,\omega)}\right)\left(\frac{\gamma_{pr}(pL,\omega)}{1+\gamma_{pr}(pL,\omega)}\right)} \times G\left[\frac{\gamma_{pr}(pL,\omega)+\gamma_{po}(pL,\omega)\gamma_{pr}(pL,\omega)}{1+\gamma_{pr}(pL,\omega)}\right]$$

where

$$G(x) = e^{-x/2}[(1+x)I_0(x/2) + xI_1(x/2)]$$

- *a posteriori* SNR:

$$\gamma_{po}(pL,\omega) = \frac{P[|Y(pL,\omega)|^2 - \hat{S}_b(\omega)]}{\hat{S}_b(\omega)}$$

- *a priori* SNR:

$$\gamma_{pr}(pL,\omega) = (1-a)P[\gamma_{po}(pL,\omega)] + a\frac{|H_s((p-1)L,\omega)Y((p-1)L,\omega)|^2}{\hat{S}_b(\omega)}$$

Binaural Representation

- Compute the enhanced signal (object) through $H_s(pL, \omega)$
- Compute its complement: $1 - H_s(pL, \omega)$
- Play a stereo signal: i.e., left channel for the object and right channel it complement
- Illusion: object and its complement come from different directions, and thus there is further enhancement!!!

Outline

Model-Based Processing

- Model-based Wiener Filter:

$$H(\omega) = \frac{\hat{S}_x(\omega)}{\hat{S}_x(\omega) + \hat{S}_b(\omega)}$$

- Power spectrum estimate of speech:

$$\hat{S}_x(\omega) = \frac{A^2}{|1 - \sum_{k=1}^{p} \hat{a}_k e^{-j\omega k}|^2}$$

Stochastic Estimation methods

- Maximum Likelihood, ML

$$\max_a p_{Y|A}(y|a)$$

- Maximum a posteriori, (MAP)

$$\max_a p_{A|Y}(a|y)$$

  knowing the a priori probability $p_A(a)$
- Minimum-Mean-Squared Error, (MMSE)

$$\text{mean of } p_{A|Y}(a|y)$$

Stochastic Estimation methods

- Maximum Likelihood, ML

$$\max_a p_{Y|A}(y|a)$$

- Maximum a posteriori, (MAP)

$$\max_a p_{A|Y}(a|y)$$

knowing the a priori probability $p_A(a)$

- Minimum-Mean-Squared Error, (MMSE)

mean of $p_{A|Y}(a|y)$

Stochastic Estimation methods

- Maximum Likelihood, ML

$$\max_a p_{Y|A}(y|a)$$

- Maximum a posteriori, (MAP)

$$\max_a p_{A|Y}(a|y)$$

knowing the a priori probability $p_A(a)$

- Minimum-Mean-Squared Error, (MMSE)

$$\text{mean of } p_{A|Y}(a|y)$$

Example of (L)MAP estimation for Enhancement

- Solution to the MAP problem requires solving a set of nonlinear equations.
- Instead we use a linearized approach of MAP:
  - Initial estimation of $\hat{a}^0$
  - Estimate speech spectrum $E[x|\hat{a}^0, y]$
  - Having a speech estimate, estimate a new parameter vector $\hat{a}^1$
  - Estimate speech spectrum:

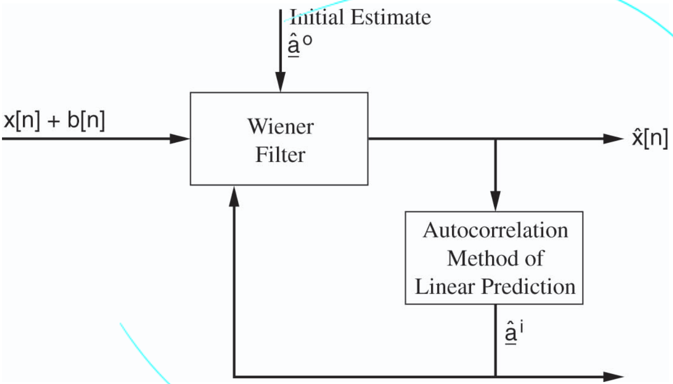
$$\hat{S}_x^1(\omega) = \frac{A^2}{|1 - \sum_{k=1}^{p} \hat{a}_k^1 e^{-j\omega k}|^2}$$

  - Estimate suppression filter:

$$H^1(\omega) = \frac{\hat{S}_x^1(\omega)}{\hat{S}_x^1(\omega) + \hat{S}_b(\omega)}$$

  - make iterations

Linearized MAP

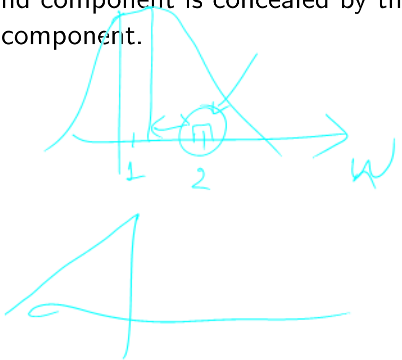


Outline

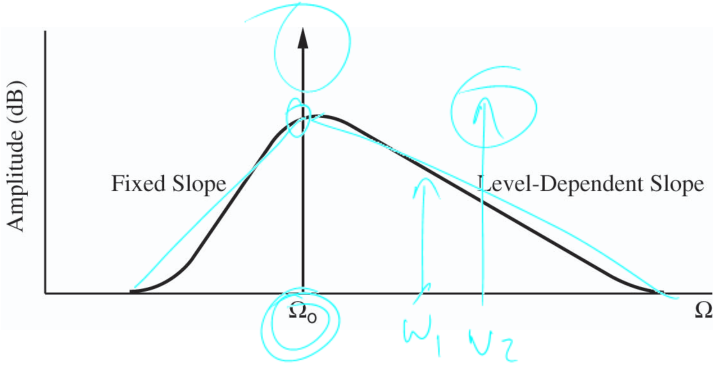
Auditory Masking

Auditory masking: one sound component is concealed by the presence of another sound component.

- Frequency masking
- Temporal masking
- Critical band
- Masking threshold
- Maskee
- Masker



Masking Threshold Curve



Frequency-Domain Masking Principles

- Physiologically-based/Psychoacoustically-based filters
- Critical Bands: Bandwidth of Psychoacoustically-based filters
- Quantized critical bands (*Bark Scale*):

$$z = 13 \arctan(0.76f) + 3.5 \arctan(f/7500)$$

- Quantized critical bands (*Mel Scale*):

$$m = 2595 \log_1 0(1 + f/700)$$

Frequency-Domain Masking Principles

- Physiologically-based/Psychoacoustically-based filters
- Critical Bands: Bandwidth of Psychoacoustically-based filters
- Quantized critical bands (*Bark Scale*):

$$z = 13 \arctan(0.76f) + 3.5 \arctan(f/7500)$$

- Quantized critical bands (*Mel Scale*):

$$m = 2595 \log_1 0(1 + f/700)$$

Frequency-Domain Masking Principles

- Physiologically-based/Psychoacoustically-based filters
- Critical Bands: Bandwidth of Psychoacoustically-based filters
- Quantized critical bands (*Bark Scale*):

$$z = 13 \arctan(0.76f) + 3.5 \arctan(f/7500)$$

- Quantized critical bands (*Mel Scale*):

$$m = 2595 \log_1 0(1 + f/700)$$

Frequency-Domain Masking Principles

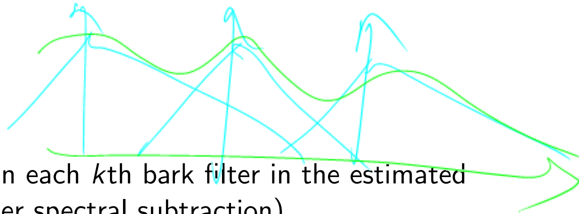
- Physiologically-based/Psychoacoustically-based filters
- Critical Bands: Bandwidth of Psychoacoustically-based filters
- Quantized critical bands (*Bark Scale*):

$$z = 13 \arctan(0.76f) + 3.5 \arctan(f/7500)$$

- Quantized critical bands (*Mel Scale*):

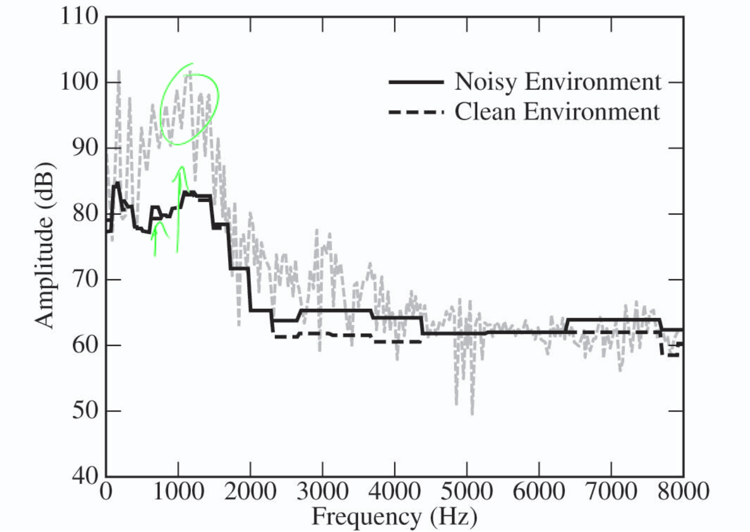
$$m = 2595 \log_1 0(1 + f/700)$$

Masking Threshold Calculation



- Compute energy E_k in each kth bark filter in the estimated speech spectrum (after spectral subtraction)
- Convolve each E_k with a “spreading function” h_k:
  $T_k = E_k \star h_k$
- Subtract a threshold offset depending if the masker is noise-like or tone-like.
- Map T_k to linear frequency scale to obtain $T(pL, \omega)$

Auditory Masking Threshold Curves



Approach 1

- Suppression filter:

$$
\begin{aligned}
H_s(pL,\omega) &= [1 - aQ(pL,\omega)^{\gamma_1}]^{\gamma_2}, \quad \text{if } Q(pL,\omega)^{\gamma_1} < \tfrac{1}{a+b} \\
&= [bQ(pL,\omega)^{\gamma_1}]^{\gamma_2}, \quad \text{otherwise}
\end{aligned}
$$

  where

$$
Q(pL,\omega) = \left[\frac{\hat{S}_b(\omega)}{|Y(pL,\omega)|^2}\right]^{1/2}
$$

- Requirements: (a) Estimation of $\hat{S}_b(\omega)$, and (b) a masking threshold curve on each frame $T(pL,\omega)$.

Approach 2

- From $y[n] = x[n] + b[n]$ go to $d[n] = x[n] + ab[n]$
- If $h_s[n]$ is the impulse response of the suppression filter, then the noise error is:

$$ab[n] - h_s[n] \star b[n]$$

with short-time power spectrum:

$$\hat{S}_e(pL, \omega) = |H_s(pL, \omega) - a|^2 \hat{S}_b(\omega)$$

- Constraint:

$$|H_s(pL, \omega) - a|^2 \hat{S}_b(\omega) < T(pL, \omega)$$

or:

$$a - \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}} < H_s(pL, \omega) < a + \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}}$$

Approach 2

- From $y[n] = x[n] + b[n]$ go to $d[n] = x[n] + ab[n]$
- If $h_s[n]$ is the impulse response of the suppression filter, then the noise error is:

$$ab[n] - h_s[n] \star b[n]$$

with short-time power spectrum:

$$\hat{S}_e(pL, \omega) = |H_s(pL, \omega) - a|^2 \hat{S}_b(\omega)$$

- Constraint:

$$|H_s(pL, \omega) - a|^2 \hat{S}_b(\omega) < T(pL, \omega)$$

or:

$$a - \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}} < H_s(pL, \omega) < a + \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}}$$

Approach 2

- From $y[n] = x[n] + b[n]$ go to $d[n] = x[n] + ab[n]$
- If $h_s[n]$ is the impulse response of the suppression filter, then the noise error is:

$$ab[n] - h_s[n] \star b[n]$$

with short-time power spectrum:

$$\hat{S}_e(pL, \omega) = |H_s(pL, \omega) - a|^2 \hat{S}_b(\omega)$$

- Constraint:

$$|H_s(pL, \omega) - a|^2 \hat{S}_b(\omega) < T(pL, \omega)$$

or:

$$a - \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}} < H_s(pL, \omega) < a + \sqrt{\frac{T(pL, \omega)}{\hat{S}_b(\omega)}}$$

Outline

Acknowledgments

Most, if not all, figures in this lecture are coming from the book:

T. F. Quatieri: Discrete-Time Speech Signal Processing,
principles and practice
2002, Prentice Hall

and have been used after permission from Prentice Hall