Single and Multi Channel Feature Enhancement for Distant Speech Recognition

John McDonough(1), Matthias Wölfel(2), Friedrich Faubel(3)
Overview

I  - Applications of DSR
II - Characteristics of Human Speech
III - The Acoustic Environment
IV - Speech feature enhancement
V  - Speaker Tracking
VI - Digital Filter Banks
VII - Beamforming
Part I
Applications of Distant Speech Recognition
Application: Meeting Rooms
Application: Humanoid Robot
Application: Car Navigation System
Application: Car Control System
Application: Portable Devices

Between Japanese and English

Between Japanese and Chinese
Architecture of a Complete DSR System
Part II
Characteristics of Human Speech
Voiced and Unvoiced Speech

- Human speech consists of voice and unvoiced phones.
- During voiced phones, like vowels, the vocal chords vibrate and the speech is nearly periodic.
- During unvoiced phones, like fricatives, the vocal chords are still and the excitation is supplied by forcing air through a constriction.
A spectrogram is a representation of speech in the time-frequency plane.

- Dark areas of the spectrogram indicate regions of high energy, light areas indicate low energy.
- In a spectrogram, it is easy to distinguish voiced and unvoiced segments.
Voiced and Unvoiced Speech

- Speech is largely, but not entirely, periodic.
- This periodicity is accounted for by the well-known source-filter model of speech.
The sparseness of speech in the subband domain is largely due to the overtone series associated with periodic or voiced speech.

The fundamental frequency, typically denoted as $f_0$, is the rate at which the vocal cords open and close.

Because of this periodicity, there will be a great deal of energy in those subbands that fall directly on an integer multiple or harmonic of $f_0$.

Between the harmonics there will be very little energy in the spectral domain.
In both time and subband domains, speech displays decidedly super-Gaussian statistical characteristics.

Long-term histogram of speech in time and frequency domain and different probability density function approximations.

The frequency shown is 1.6 kHz.
Why is Speech Super-Gaussian?

- The entire field of independent component analysis (ICA) is founded on the assumption that all signals of real interest are not Gaussian-distributed.
- This argument is grounded on two points:
  - The central limit theorem implies that the sum of several r.v.s will be closer to Gaussian than any of the individual components.
  - It is well known that a Gaussian r.v. has the highest entropy of all random variables with a given variance. Hence, a Gaussian r.v. is the least predictable of all random variables.
Part III

The Acoustic Environment
Sound waves are disturbances of air molecules, which propagate as wave fronts of compressed air, followed by decompressed air.
Sound waves follow the law of superposition, i.e. individual waves from different sources add up at each point in the room.
Sound waves are reflected by obstacles, such as walls, columns, chairs and tables.
Sound waves are reflected by obstacles, such as walls, columns, chairs and tables.
Sound waves are reflected by obstacles, such as walls, columns, chairs and tables.
Sound waves are reflected by obstacles, such as walls, columns, chairs and tables.
Sound waves are reflected by obstacles, such as walls, columns, chairs and tables.
Obstacles do not only reflect sound. They also absorb some portions in a frequency dependent manner.
Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.
Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.
Sound Propagation: Coloration

- Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.
Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.

\[
d_2 + d_3 - d_1 = n\lambda
\]
Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.

\[ d_2 + d_3 - d_1 = n\lambda + \frac{\lambda}{2} \]
Constructive and destructive interference of reflections can cause comb filter like “coloration” of the sound.
Sound waves spread out behind small openings and bend around corners.
Sound Propagation: Other Effects

- **Refraction**: wave passes from one medium to another at an angle
- **Scattering**: diffuse reflections of waves
- **Surface waves**: longitudinal and transverse motion along surfaces; lower attenuation than normal sound waves
Model for a single sound source with one reflection:

\[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]
Room Impulse Response: Introduction

- Model for a single sound source with one reflection:

\[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]
Model for a single sound source with one reflection:

\[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]

Generalization:

\[ x(t) = \sum_{i=0}^{\infty} a_i \cdot s(t - \tau_i) \]
Room Impulse Response: Introduction

- Model for a single sound source with one reflection:

\[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]

- Generalization:

\[ x(t) = \sum_{i=0}^{\infty} a_i \cdot s(t - \tau_i) \]

- Impulse response model:

\[ x(t) = \sum_{i=0}^{\infty} a_i \delta(t - \tau_i) \ast s(t) \]
Room Impulse Response: Introduction

- Model for a single sound source with one reflection:

\[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]

- Generalization: \( x(t) = \sum_{i=0}^{\infty} a_i \cdot s(t - \tau_i) \)

- Impulse response model: \( x(t) = h(t) \ast s(t) \)
  with impulse response \( h(t) = \sum_{i=0}^{\infty} a_i \delta(t - \tau_i) \)
Room Impulse Response: Introduction

- Model for a single sound source with one reflection:
  \[ x(t) = a_d \cdot s(t - \tau_d) + a_r \cdot s(t - \tau_r) \]

- Generalization: \[ x(t) = \sum_{i=0}^{\infty} a_i \cdot s(t - \tau_i) \]

- Impulse response model: \[ x(t) = h(t) * s(t) \]
- **Early reflections**: semi-distinct reflections from various reflective surfaces with a delay of up to 50ms; reinforce the sound (<35ms).
**Room Impulse Response: Reinforcement & Reverberation**

- **Early reflections**: semi-distinct reflections from various reflective surfaces with a delay of up to 50ms; reinforce the sound (<35ms).

- **Late reflections**: reflections with lower amplitude that are closely spaced in time; responsible for “reverberant” sound.
**Echo:** distinct, strong late reflection with a delay of more than 100ms.
- Reverberation smears spectra in time.
- Noise fills in spectral valleys.
Effects of Noise

- Shown in the figure is a simplified plot of relative sound pressure vs. time for an utterance of the word “cat” in additive noise.

- Note that the final /t/ is obscured by the noise floor.
- Hence the word is indistinguishable from “cab” or “cap”.
Shown in the figure is a simplified plot of relative sound pressure vs. time for an utterance of the word “cat” in the presence of reverberation.

Note that the final /t/ is still obscured, but this time by the ring down of the preceding phones.

Hence the word is still indistinguishable from “cab” or “cap”.

Effects of Reverberation
A Model of the Acoustic Environment

- Model for one desired source with background noise:

\[ x(t) = h(t) \ast s(t) + n(t) \]
Central Limit Theorem

- Plot of the Gaussian pdf and the pdf obtained by summing together $N$ Laplacian random variables for several values of $N$.
- As predicted by the central limit theorem, the sum becomes ever more Gaussian with increasing $N$. 

![Graph showing the comparison between the Gaussian pdf and the pdf obtained by summing together $N$ Laplacian random variables for several values of $N$ on both linear and logarithmic scales.](image-url)
Both noise and reverberation have the effect of making speech subband samples more nearly Gaussian.

This begs the question: Can speech be enhanced by restoring its original super-Gaussian characteristics?
Part IV
Speech Feature Enhancement
Part IV-1

Speech Feature Enhancement
Motivation
A speech recognition system converts speech to text.
A speech recognition system converts speech to text.

It basically consists of two components:

1. **Front End**
2. **Decoder**
A speech recognition system converts speech to text.

It basically consists of two components:

- **Front End**: extracts speech features from the audio signal
A speech recognition system converts speech to text.

It basically consists of two components:

- **Front End**: extracts speech features from the audio signal
- **Decoder**: finds that sentence (sequence of acoustical states), which is the most likely explanation for the observed sequence of speech features
Speech Feature Extraction: Windowing
Speech Feature Extraction: Windowing
Speech Feature Extraction: Windowing
Speech Feature Extraction: Windowing
Performing spectral analysis separately for each frame yields a time-frequency representation.
Performing spectral analysis separately for each frame yields a time-frequency representation.
Emulation of the **logarithmic frequency and intensity** perception of the human auditory system
Background noise distorts speech features

- **Result**: features do not match the features used during training

- **Consequence**: severely degraded recognition performance
Speech Feature Enhancement: Motivation

- **Idea:**
  - train speech recognition system on clean speech
  - try to map distorted features to clean speech features
Part II-2

Speech Feature Enhancement
The Interaction Function
Principle of Superposition: signals are additive
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

- **noisy speech**
- **noise**
- **clean speech**
In the signal domain we have the following relationship:

\[ y = s + n \]
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]

Taking the magnitude square on both sides, we get:

\[ \| Y_F \|^2 = \| S_F + N_F \|^2 \]
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]

Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = (S_F + N_F)(S_F + N_F)^* \]
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]

Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* + S_F N_F^* + S_F^* N_F \]
In the **signal domain** we have the following *relationship*:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]

Taking the magnitude square on both sides, we get:

\[ Y_F^* Y_F = S_F^* S_F + N_F^* N_F + S_F N_F^* + S_F^* N_F \]

*zero in average*
In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

After **Fourier transformation**, this becomes:

\[ Y_F = S_F + N_F \]

Taking the magnitude square on both sides, we get:

\[ Y_F^* Y_F = S_F^* S_F + N_F^* N_F \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]

Hence, in the **power spectral domain** we have:

\[ Y_p = S_p + N_p \]
- Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]

- Hence, in the **power spectral domain** we have:

\[ Y_P = S_P + N_P \]

- In the **log power spectral domain** that becomes:

\[ e^{Y_l} = e^{S_l} + e^{N_l} \quad n_l = \log(N_P) \]

\[ = Y_P \quad = S_P \quad = N_P \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]

Hence, in the **power spectral domain** we have:

\[ Y_P = S_P + N_P \]

In the **log power spectral domain** that becomes:

\[ e^{Y_i} = e^{S_i} + e^{N_i} \]

\[ = Y_p \quad = S_p \quad = N_p \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]

Hence, in the **power spectral domain** we have:

\[ Y_p = S_p + N_p \]

In the **log power spectral domain** that becomes:

\[ \log(e^{Y_i}) = \log(e^{S_i} + e^{N_i}) \]
Taking the magnitude square on both sides, we get:

\[ Y^*_F Y_F = S^*_F S_F + N^*_F N_F \]

Hence, in the **power spectral domain** we have:

\[ Y_P = S_P + N_P \]

In the **log power spectral domain** that becomes:

\[ y_i = \log(e^{S_i} + e^{N_i}) \]
Taking the magnitude square on both sides, we get:

\[ Y_F Y_F^* = S_F S_F^* + N_F N_F^* \]

Hence, in the **power spectral domain** we have:

\[ Y_P = S_P + N_P \]

In the **log power spectral domain** that becomes:

\[ Y_l = \log(e^{S_l} + e^{N_l}) \]

Acero, 1990
Part IV-2

Speech Feature Enhancement
Transforming Probabilities
- In the **signal domain** we have the following **relationship**:

\[ y = s + n \]

- In the **log Mel domain** that translates to:

\[ y_i = \log(e^s + e^n) \]

*nonlinear interaction function*
Transforming Probabilities

Motivation

noise power

noisy speech power

clean speech power
Transforming Probabilities

Motivation

- Noisy speech power
- Clean speech power
- Noise power
Transforming Probabilities

Motivation
Transforming Probabilities
Motivation
Transformation results in a non-Gaussian probability distribution for noisy speech features.

\[ Y_i = \log(e^{S_i} + e^{N_i}) \]
Transformation of a random variable

- Transformation  \( X \xrightarrow{f} Y \)
- Probability density function  \( p_X(x) \)
Transformation of a random variable

- Transformation: $X \xrightarrow{f} Y$
- Probability density function: $p_X(x)$

The transformation maps each $x$ to a $y$: $x \rightarrow f(x)$
Transformation of a random variable

- Transformation: $X \xrightarrow{f} Y$
- Probability density function: $p_X(x)$

The transformation maps each $x$ to a $y$: $x \rightarrow f(x)$

Conversely, each $y$ can be identified with $f^{-1}(y)$
Transformation of a random variable

- Transformation \( X \xrightarrow{f} Y \)
- Probability density function \( p_X(x) \)

**Idea:** use \( f^{-1} \) to map the distribution of \( Y \) to the distribution of \( X \)

\[
p_Y(y) = p_X(f^{-1}(y))
\]
**Transformation of a random variable**

- Transformation: $X \xrightarrow{f} Y$
- Probability density function: $p_X(x)$

**Idea:** Use $f^{-1}$ to map the distribution of $Y$ to the distribution of $X$

$$p_Y(y) = p_X(f^{-1}(y))$$

**change of variables**
Transforming Probabilities

Introduction

Transformation of a random variable

- Transformation $X \xrightarrow{f} Y$
- Probability density function $p_X(x)$

**Idea:** use $f^{-1}$ to map the distribution of $Y$ to the distribution of $X$

$$p_Y(y) = p_X(f^{-1}(y)) \left[ \text{det } J(f^{-1}(y)) \right]$$

Jacobian determinant
Transformation of a random variable

- Transformation $\mathcal{X} \xrightarrow{f} \mathcal{Y}$
- Probability density function $p_X(x)$

**Idea:** use $f^{-1}$ to map the distribution of $Y$ to the distribution of $X$

$$p_Y(y) = p_X\left(f^{-1}(y)\right) \cdot \left|\text{det } J\left(f^{-1}(y)\right)\right|$$

**Fundamental Transformation Law of Probabilities**
Transforming Probabilities
Monte Carlo

**Idea:** approximate probability distribution by samples drawn from the distribution.

\[
p(x) \approx \sum_{n=1}^{N} \frac{1}{N} \delta(x - x^{(n)})
\]
**Transforming Probabilities**

**Monte Carlo**

- **Idea:** approximate probability distribution by samples drawn from the distribution.

![pdf](image1)  
**pdf**

![cumulative density function](image2)  
**cumulative density function**
- **Idea:** approximate probability distribution by samples drawn from the distribution.

- **Then:** transform each sample $y^{(n)} = f(x^{(n)})$
Idea: approximate probability distribution by samples drawn from the distribution.

Then: transform each sample $y^{(n)} = f(x^{(n)})$
**Idea:** Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.

**Note:** A linear transformation of a Gaussian random variable results in a Gaussian random variable.
Transforming Probabilities
Local Linearization

- **Idea:** Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.

*Note:* a linear transformation of a Gaussian random variable results in a Gaussian random variable.

Vector Taylor Series Approach

Moreno, 1996
Transforming Probabilities
Local Linearization

- **Idea:** Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.

  Vector Taylor Series Approach

  Also used in the extended Kalman filter.

  **Note:** A linear transformation of a Gaussian random variable results in a Gaussian random variable.

  Moreno, 1996
Transforming Probabilities
Local Linearization

- **Idea:** Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.
Transforming Probabilities
Local Linearization

**Idea:** Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.
Idea: Locally linearize the interaction function around the mean of speech and noise, using a first order Taylor series expansion.
Idea: similar as in Monte Carlo, select points in a deterministic fashion and in such a way that they capture the mean and co-variance of the distribution
Transforming Probabilities
The Unscented Transform

Idea: similar as in Monte Carlo, select points in a deterministic fashion and in such a way that they capture the mean and covariance of the distribution.
Transforming Probabilities
The Unscented Transform

select points
Transforming Probabilities
The Unscented Transform

select points

transform points
Transforming Probabilities
The Unscented Transform

1. Select points
2. Re-estimate parameters of the Gaussian distribution
3. Transform points
Transforming Probabilities
The Unscented Transform

Comparison to local linearization:

local linearization

unscented transform
Transforming Probabilities
The Unscented Transform

Comparison to local linearization:

- The unscented transform is exact for linear transforms
- For nonlinear transforms the mean and covariance estimates are accurate up to the second order term of the Taylor series expansion.
Density approximation with the Adaptive Level of Detail Transform (ALoDT):
Density approximation with the Adaptive Level of Detail Transform (ALoDT):
Density approximation with the Adaptive Level of Detail Transform (ALoDT):
Density approximation with the Adaptive Level of Detail Transform (ALoDT):

Faubel 2010
Density approximation with the Adaptive Level of Detail Transform (ALoDT):
Density approximation with the Adaptive Level of Detail Transform (ALoDT):

Faubel 2010
Part IV-3

Speech Feature Enhancement
The MMSE Solution
Speech Feature Enhancement
The MMSE Solution

- **Idea:**
  - train speech recognition system on clean speech
  - try to map distorted features to clean speech features

- **Systematic Approach:**
  - derive an estimator for clean speech given noisy speech
Let $\hat{S} = \mathcal{D}(y)$ be an estimator for clean speech $S$, given noisy speech $y$. 
Let $\hat{S} = \delta(y)$ be an estimator for clean speech $S$, given noisy speech $Y$.

Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta | y] = E\left[ \left\| \delta(y) - s \right\|^2 | y \right]$$

$$= \int \left\| \delta(y) - s \right\|^2 p(s | y) dx$$
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta \mid y] = E \left[ \left\| \delta(y) - s \right\|^2 \mid y \right]$$

$$= \int \left\| \delta(y) - s \right\|^2 p(s \mid y) dx$$
Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right] \\
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]
Then the expected mean square error introduced by using \( \hat{s} \) instead of the true \( s \) is:

\[
MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right] \\
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]

Taking the derivative with respect to \( \delta(y) \) yields ...

\[
\frac{\partial MSE[\delta | y]}{\partial \delta(y)} = \frac{\partial \int \| \delta(y) - s \|^2 p(s | y) dx}{\partial \delta(y)}
\]
Speech Feature Enhancement
The MMSE Solution

Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[δ | y] = E \left[ \| δ(y) - s \|^2 | y \right]
\]

\[
= \int \| δ(y) - s \|^2 p(s | y) dx
\]

Taking the derivative with respect to \( δ(y) \) yields ...

\[
\frac{∂MSE[δ | y]}{∂δ(y)} = \frac{∂}{∂δ(y)} \int \| δ(y) - s \|^2 p(s | y) dx
\]

pulling the derivative into the integral
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

\[
MSE[\delta | y] = E\left[ \| \delta(y) - s \|^2 | y \right]
\]

\[
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]

Taking the derivative with respect to $\delta(y)$ yields ...

\[
\frac{\partial MSE[\delta | y]}{\partial \delta(y)} = \int \frac{\partial \| \delta(y) - s \|^2}{\partial \delta(y)} p(s | y) dx
\]
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta \mid y] = E \left[ \left\| \delta(y) - s \right\|^2 \mid y \right] \\
= \int \left\| \delta(y) - s \right\|^2 p(s \mid y)dx
\]

- Taking the derivative with respect to \( \delta(y) \) yields ...

\[
\frac{\partial MSE[\delta \mid y]}{\partial \delta(y)} = \int \frac{\partial \left\| \delta(y) - s \right\|^2}{\partial \delta(y)} p(s \mid y)dx
\]

and then taking the derivative
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta \mid y] = E\left[\|\delta(y) - s\|^2 \mid y\right]$$

$$= \int \|\delta(y) - s\|^2 p(s \mid y)dx$$

Taking the derivative with respect to $\delta(y)$ yields ...

$$\frac{\partial MSE[\delta \mid y]}{\partial \delta(y)} = \int 2(\delta(y) - s) p(s \mid y)dx$$
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta | y] = E\left[ \| \delta(y) - s \|^2 | y \right] \\
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]

- Taking the derivative with respect to \( \delta(y) \) yields ...

\[
\frac{\partial MSE[\delta | y]}{\partial \delta(y)} = \int 2(\delta(y) - s)p(s | y) dx
\]
Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta \mid y] = E\left[ \left\| \delta(y) - s \right\|^2 \mid y \right]
\]

\[
= \int \left\| \delta(y) - s \right\|^2 p(s \mid y) dx
\]

... which when equated to zero gives:

\[
\frac{\partial MSE[\delta \mid y]}{\partial \delta(y)} = \int 2(\delta(y) - s)p(s \mid y) dx = 0
\]
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta | y] = E\left[\|\delta(y) - s\|^2 | y\right]$$

$$= \int \|\delta(y) - s\|^2 p(s | y) dx$$

... which when equated to zero gives:

$$\int 2(\delta(y) - s)p(s | y) ds = 0$$
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta \mid y] = E\left[ \left\| \delta(y) - s \right\|^2 \mid y \right]$$

$$= \int \left\| \delta(y) - s \right\|^2 p(s \mid y) dx$$

... which when equated to zero gives:

$$\int 2(\delta(y) - s)p(s \mid y)ds = 0$$

pull scalar out of the integral
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta \mid y] = E\left[ \|\delta(y) - s\|^2 \mid y \right]$$

$$= \int \|\delta(y) - s\|^2 p(s \mid y)dx$$

- ... which when equated to zero gives:

$$2\int (\delta(y) - s)p(s \mid y)ds = 0$$

\text{drop scalar}
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right]$$

$$= \int \| \delta(y) - s \|^2 p(s | y)dx$$

... which when equated to zero gives:

$$\int (\delta(y) - s)p(s | y)ds = 0$$
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta \mid y] = E \left[ \|\delta(y) - s\|^2 \mid y \right]$$

$$= \int \|\delta(y) - s\|^2 p(s \mid y)dx$$

- ... which when equated to zero gives:

$$\int (\delta(y) - s)p(s \mid y)ds = 0$$

expand
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right] = \int \| \delta(y) - s \|^2 p(s | y) dx
\]

- ... which when equated to zero gives:

\[
\int \delta(y)p(s | y) - sp(s | y)ds = 0
\]
Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right] \\
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]

... which when equated to zero gives:

\[
\int \delta(y)p(s | y) - sp(s | y) ds = 0
\]

making use of the linearity of integration
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right]
\]

\[
= \int \| \delta(y) - s \|^2 p(s | y) dx
\]

- ... which when equated to zero gives:

\[
\int \delta(y)p(s | y) ds - \int sp(s | y) ds = 0
\]
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta | y] = E\left[ \|\delta(y) - s\|^2 \mid y \right]$$

$$= \int \|\delta(y) - s\|^2 p(s \mid y)dx$$

... which when equated to zero gives:

$$\int \delta(y)p(s \mid y)ds - \int sp(s \mid y)ds = 0$$

pulling this out of the integral
Then the expected mean square error introduced by using $\hat{S}$ instead of the true $S$ is:

$$MSE[\delta | y] = E \left[ \| \delta(y) - s \|^2 | y \right]$$

$$= \int \| \delta(y) - s \|^2 p(s | y) dx$$

... which when equated to zero gives:

$$\delta(y) \int p(s | y) ds - \int sp(s | y) ds = 0$$
Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
MSE\left[ \delta \mid y \right] = E\left[ \left\| \delta(y) - s \right\|^2 \mid y \right] = \int \left\| \delta(y) - s \right\|^2 p(s \mid y)dx
\]

... which when equated to zero gives:

\[
\delta(y)\frac{\int p(s \mid y)ds}{\int s p(s \mid y)ds} - \int sp(s \mid y)ds = 0
\]

\(=1\)
Speech Feature Enhancement
The MMSE Solution

- Then the expected mean square error introduced by using \( \hat{S} \) instead of the true \( S \) is:

\[
\text{MSE}[\delta \mid y] = E \left[ \| \delta(y) - s \|^2 \mid y \right]
= \int \| \delta(y) - s \|^2 p(s \mid y) dx
\]

- Minimizing the MSE with respect to \( \delta(y) \) yields the optimal estimator with respect to the MMSE criterion:

\[
\delta_{\text{MMSE}}(y) = \int sp(s \mid y) ds
\]
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int sp(s \mid y)ds$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int sp(s | y)ds$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{\text{MMSE}}(y) = \int sp(s | y)ds
$$

But how to obtain this distribution?
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \int sp(s \mid y)ds$$

- **Idea:** assume that the joint distribution of $S$ and $Y$ is Gaussian

$$\rho_{S,Y} \left( \begin{bmatrix} s \\ y \end{bmatrix} \right) = N \left( \begin{bmatrix} s \\ y \end{bmatrix} ; \begin{bmatrix} \mu_s \\ \mu_y \end{bmatrix} , \begin{bmatrix} \Sigma_{ss} & \Sigma_{sy} \\ \Sigma_{ys} & \Sigma_{yy} \end{bmatrix} \right)$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \int sp(s \mid y)ds$$

- *Idea:* assume that the joint distribution of $S$ and $Y$ is Gaussian

$$p_{S,Y}
\begin{bmatrix}
s \\
y
\end{bmatrix}
= \mathcal{N}
\begin{bmatrix}
s \\
y
\end{bmatrix};
\begin{bmatrix}
\mu_s \\
\mu_y
\end{bmatrix},
\begin{bmatrix}
\Sigma_{SS} & \Sigma_{SY} \\
\Sigma_{YS} & \Sigma_{YY}
\end{bmatrix}$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int sp(s \mid y)ds$$

- **Idea:** assume that the joint distribution of $S$ and $Y$ is Gaussian

$$p_{S,Y}\left(\begin{bmatrix} s \\ y \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_s \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{SS} & \Sigma_{SY} \\ \Sigma_{YS} & \Sigma_{YY} \end{bmatrix}\right)$$

Stereo-Based Stochastic Mapping

Afify, 2007
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{MMSE}(y) = \int sp(s | y)ds
$$

- **Idea:** assume that the joint distribution of $S$ and $Y$ is Gaussian

$$
\rho_{S,Y}
\begin{bmatrix}
s \\ y
\end{bmatrix}
= \mathcal{N}
\begin{bmatrix}
s \\ y
\end{bmatrix};
\begin{bmatrix}
\mu_S \\ \mu_Y
\end{bmatrix},
\begin{bmatrix}
\Sigma_{SS} & \Sigma_{SY} \\
\Sigma_{YS} & \Sigma_{YY}
\end{bmatrix}
$$
Speech Feature Enhancement
The MMSE Solution

Idea: assume that the joint distribution of S and Y is Gaussian

\[
p_{s,y}(s, y) = \mathcal{N}\left(\begin{bmatrix} s \\ y \end{bmatrix} ; \begin{bmatrix} \mu_s \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{ss} & \Sigma_{sy} \\ \Sigma_{ys} & \Sigma_{yy} \end{bmatrix}\right)
\]
Speech Feature Enhancement
The MMSE Solution

**Idea:** assume that the joint distribution of S and Y is Gaussian

\[
p_{S,Y}(\begin{bmatrix} s \\ y \end{bmatrix}) = \mathcal{N}(\begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_S \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{SS} & \Sigma_{SY} \\ \Sigma_{YS} & \Sigma_{YY} \end{bmatrix})
\]
Speech Feature Enhancement
The MMSE Solution

- **Idea:** assume that the joint distribution of $S$ and $Y$ is Gaussian

  $$p_{S,Y}(\begin{bmatrix} s \\ y \end{bmatrix}) = \mathcal{N}(\begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_s \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{SS} & \Sigma_{SY} \\ \Sigma_{YS} & \Sigma_{YY} \end{bmatrix})$$

- Then the conditional distribution of $S|Y$ is again Gaussian:

  $$p(s | y) = \mathcal{N}(s; \mu_{S|Y}, \Sigma_{SS|Y})$$

  with conditional mean and covariance matrix

  $$\mu_{S|Y} = \mu_s + \Sigma_{SY} \Sigma_{YY}^{-1} (y - \mu_y), \quad \Sigma_{SS|Y} = \Sigma_{SS} - \Sigma_{SY} \Sigma_{YY}^{-1} \Sigma_{YS}$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \int sp(s \mid y) ds$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \int sp(s \mid y)ds$$

Under the Gaussian assumption, this integral is easily obtained:

$$\mu_{S\mid y} = \mu_s + \Sigma_{SY} \Sigma_{YY}^{-1} (y - \mu_y)$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int sp(s \mid y)ds$$

Under the Gaussian assumption, this integral is easily obtained:

$$\mu_{s\mid y} = \mu_s + \Sigma_{s\mid y} \Sigma_{y\mid y}^{-1} (y - \mu_y)$$

Problem: speech is known to be multi modal
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int s p(s | y) ds$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int s \sum_{k=1}^{K} p(s, k | y) ds$$

Introduce the index $k$ of the mixture component as a hidden variable.
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \int s \sum_{k=1}^{K} p(s, k \mid y) \, ds$$

Then rewrite this as

$$p(s, k \mid y) = p(s \mid y, k)p(k \mid y)$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \int s \sum_{k=1}^{K} p(s \mid y, k) p(k \mid y) \, ds$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{MMSE}(y) = \int s \sum_{k=1}^{K} p(s \mid y, k) p(k \mid y) \, ds
$$

pull the sum out of the integral
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \sum_{k=1}^{K} \int s \ p(s \mid y,k)p(k \mid y) ds$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \sum_{k=1}^{K} \int s \ p(s \mid y, k) p(k \mid y) ds$$

independent of $s$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \sum_{k=1}^{K} \int s \ p(s \mid y, k) p(k \mid y) ds$$

pull this out of the integral
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{MMSE}(y) = \sum_{k=1}^{K} p(k \mid y) \int s \ p(s \mid y, k) \, ds
$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{MMSE}(y) = \sum_{k=1}^{K} p(k | y) \int s \, p(s | y, k) \, ds$$

Probability that clean speech originated from the $k$th Gaussian given the noisy speech spectrum $y$. 
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \sum_{k=1}^{K} p(k \mid y) \int s \ p(s \mid y, k) \, ds$$

Clean speech estimate of the $k$-th Gaussian:

$$\mu_{s \mid y, k} = \mu_{s \mid k} + \sum_{s \mid y \mid k} \Sigma_{y \mid k}^{-1} (y - \mu_{y \mid k})$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{MMSE}(y) = \sum_{k=1}^{K} p(k \mid y) \int s p(s \mid y, k) \, ds
$$

$$
p(k \mid y) = \frac{p(y \mid k)p(k)}{p(y)} \quad \text{Bayes’ theorem}
$$
Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$
\delta_{\text{MMSE}}(y) = \sum_{k=1}^{K} p(k \mid y) \int s \ p(s \mid y, k) \, ds
$$

$$
p(k \mid y) = \frac{p(y \mid k)p(k)}{\sum_{k=1}^{K} p(y \mid k)p(k)}
$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \sum_{k=1}^{K} p(k \mid y) \int s \, p(s \mid y, k) \, ds$$

$$\frac{p(y \mid k)p(k)}{\sum_{k=1}^{K} p(y \mid k)p(k)} = p(k \mid y)$$

$$\mu_{s\mid y,k} = \mu_{s\mid k} + \sum_{s\mid k} \sum_{y\mid k}^{-1} (y - \mu_{y\mid k})$$
Speech Feature Enhancement
The MMSE Solution

- Minimizing the MSE with respect to $\delta(y)$ yields the optimal estimator with respect to the MMSE criterion:

$$\delta_{\text{MMSE}}(y) = \sum_{k=1}^{K} p(k \mid y) \int s \, p(s \mid y, k) \, ds$$

$$\frac{p(y \mid k)p(k)}{\sum_{k=1}^{K} p(y \mid k)p(k)} = p(k \mid y)$$

$$\mu_{s\mid y,k} = \mu_{s\mid k} + \sum_{sY\mid k} \sum_{Y\mid k}^{-1} (y - \mu_{y\mid k})$$

$$p_{s,y\mid k} \left( \begin{bmatrix} s \\ y \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} s \\ y \end{bmatrix} ; \begin{bmatrix} \mu_{s\mid k} \\ \mu_{y\mid k} \end{bmatrix}, \begin{bmatrix} \Sigma_{s\mid k} & \Sigma_{sY\mid k} \\ \Sigma_{YS\mid k} & \Sigma_{Y\mid k} \end{bmatrix} \right)$$

joint distribution
Part IV-4

Speech Feature Enhancement
Model Based Enhancement
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
- Noise is modeled as a single Gaussian
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
- Noise is modeled as a single Gaussian
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
- Noise is modeled as a single Gaussian
- Presence of noise changes the clean speech distribution according to the interaction function
Model-based Speech Feature Enhancement:

- Distribution of clean speech is modeled as **Gaussian Mixture**
- Noise is modeled as a single Gaussian
- Presence of noise changes the clean speech distribution according to the *interaction function*

Construct the joint distribution of clean and noisy speech based on this model:

\[
\rho_{s,y|k} \left( \begin{bmatrix} s \\ y \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_{s|k} \\ \mu_{y|k} \end{bmatrix}, \begin{bmatrix} \Sigma_{ss|k} & \Sigma_{sy|k} \\ \Sigma_{ys|k} & \Sigma_{yy|k} \end{bmatrix} \right)
\]
Model Based Feature Enhancement: Basics

- Construct the joint distribution of clean and noisy speech based on this model

\[
\rho_{s,y|k}\left(\begin{bmatrix} s \\ y \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_{s|k} \\ \mu_{y|k} \end{bmatrix}, \begin{bmatrix} \Sigma_{ss|k} & \Sigma_{sy|k} \\ \Sigma_{ys|k} & \Sigma_{yy|k} \end{bmatrix}\right)
\]
Model Based Feature Enhancement: Basics

- Construct the joint distribution of clean and noisy speech based on this model

\[ p_{s,y|k} \left( \begin{bmatrix} s \\ y \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} s \\ y \end{bmatrix}; \begin{bmatrix} \mu_{s|k} \\ \mu_{y|k} \end{bmatrix}, \begin{bmatrix} \Sigma_{ss|k} & \Sigma_{sy|k} \\ \Sigma_{ys|k} & \Sigma_{yy|k} \end{bmatrix} \right) \]
Construct the joint distribution of clean and noisy speech based on this model:

\[
p_{s,n|k} \left( \begin{bmatrix} s \\ n \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} s \\ n \end{bmatrix}; \begin{bmatrix} \mu_{s|k} \\ \mu_N \end{bmatrix}, \begin{bmatrix} \Sigma_{ss|k} & 0 \\ 0 & \Sigma_{nn} \end{bmatrix} \right)
\]
Construct the joint distribution of clean and noisy speech based on this model:

\[
p_{S,N|k}
\begin{pmatrix}
  s \\
  n
\end{pmatrix}
= \mathcal{N}
\begin{pmatrix}
  s \\
  n
\end{pmatrix};
\begin{pmatrix}
  \mu_{S|k} \\
  \mu_N
\end{pmatrix},
\begin{pmatrix}
  \Sigma_{SS|k} & 0 \\
  0 & \Sigma_{NN}
\end{pmatrix}
\]

\[
\begin{pmatrix}
  S | k \\
  N
\end{pmatrix}
\xrightarrow{f}
\begin{pmatrix}
  S | k \\
  Y | k
\end{pmatrix}
\]

\[
\begin{pmatrix}
  s \\
  n
\end{pmatrix}
\xrightarrow{f}
\begin{pmatrix}
  s \\
  y = f(s,n)
\end{pmatrix}
\]
Construct the joint distribution of clean and noisy speech based on this model

\[
p_{S,N|k}(s|n) = \mathcal{N}
\begin{bmatrix}
    s \\
    n
\end{bmatrix};
\begin{bmatrix}
    \mu_{S|k} \\
    \mu_{N}
\end{bmatrix},
\begin{bmatrix}
    \Sigma_{SS|k} & 0 \\
    0 & \Sigma_{NN}
\end{bmatrix}
\]

\[
p_{S,Y|k}(s|y) = \mathcal{N}
\begin{bmatrix}
    s \\
    y
\end{bmatrix};
\begin{bmatrix}
    \mu_{S|k} \\
    \mu_{Y|k}
\end{bmatrix},
\begin{bmatrix}
    \Sigma_{SS|k} & \Sigma_{SY|k} \\
    \Sigma_{YS|k} & \Sigma_{YY|k}
\end{bmatrix}
\]
Noise Estimation:

- Find that noise distribution, which is the most likely explanation for the observed, noisy speech features
Noise Estimation:

- Find that noise distribution, which is the most likely explanation for the observed, noisy speech features

\[
\max_{\mu_n, \Sigma_n} p(y_1, \ldots, y_T \mid \mu_n, \Sigma_n)
\]

mean and covariance of the noise
Noise Estimation:

- Find that noise distribution, which is the most likely explanation for the observed, noisy speech features

\[
\max_{\mu_n, \Sigma_n} p(y_1, \ldots, y_T \mid \mu_n, \Sigma_n)
\]

- **Problem**: the observations are also dependent on speech!

\[
Y_i = \log(e^{S_i} + e^{N_i}) =: f(S_i, N_i)
\]
Problem: the observations are also dependent on speech!
Problem: the observations are also dependent on speech!
Model Based Feature Enhancement: Noise Estimation

Noise Estimation:

- Find that noise distribution, which is the most likely explanation for the observed, noisy speech features

\[
\max_{\mu_n, \Sigma_n} p(y_1, \ldots, y_T \mid \mu_n, \Sigma_n)
\]

- **Problem**: the observations are also dependent on speech!

\[
Y_i = \log(e^{S_i} + e^{N_i}) =: f(S_i, N_i)
\]
Noise Estimation:

- Find that noise distribution, which is the most likely explanation for the observed, noisy speech features

\[
\max_{\mu_n, \Sigma_n} p(y_1, \ldots, y_T | \mu_n, \Sigma_n)
\]

- **Problem**: the observations are also dependent on speech!

- Hence, the Expectation Maximization algorithm is used.

---

Rose, 1994

Moreno, 1996
Sequential noise estimation:

- Sequential expectation maximization (SEM), Kim, 1998
Sequential noise estimation:

- Sequential expectation maximization (SEM), Kim, 1998
- Interacting Multiple Model (IMM) Kalman Filter, Kim, 1999
Model Based Feature Enhancement: Noise Estimation

Sequential noise estimation:

- Sequential expectation maximization (SEM), Kim, 1998
- Interacting Multiple Model (IMM) Kalman Filter, Kim, 1999
- Particle filter, Yao, 2001
Part IV-5

Speech Feature Enhancement
Experimental Results
Experimental Results

Speech Recognition Experiments

- **clean speech**: from MC-WSJ-AV corpus
- **noise**: from the NOISEX-92 database (artificially added)
- **ASR system**:
  - features: MFCC with 13 components, stacking of 15 frames, LDA
  - codebooks: 1,743 states, fully continuous, 70K Gaussians
  - decoder: WFST, fast on the fly composition
Experimental Results

WER, destroyer engine noise

- noisy
- EM (N,C)
- oracle(N)
- oracle (N,C)

05dB 10dB 15dB
Experimental Results

WER, factory noise

- **noisy**
- **EM (N,C)**
- **oracle(N)**
- **oracle (N,C)**
Part V

Speaker Tracking
Speaker-tracking is the determination of the physical location of one or more active speakers.

This information is useful foremost because it is required for beamforming.

As we will see, however, a speaker tracking system can also provide other useful information.
Many important problems in science and engineering can be solved with a Bayesian filter.

Such a formulation involves estimating using a sequence of observations to estimate an observable state.
The best known Bayesian filter is undoubtedly the Kalman filter.
The Kalman filter has a predictor-corrector structure.
This structure is illustrated schematically in the figure.
The operation of the Kalman filter is shown schematically in the figure.
For acoustic speaker tracking, the sequence $y_k$ of observations corresponds to time delays of arrival between microphone pairs.

The state sequence $x_k$ corresponds to the position of the speaker.

The problem of speaker tracking is then one of estimating $x_k$ based on $y_k$. 
Shown in the figure are the instrumented meeting rooms at the Universities of Karlsruhe and Edinburgh.
The *joint probabilistic data association filter* (JPDAF) is an extension of the Kalman filter that can simultaneously maintain several active tracks.

In the figure, two active tracks are shown with current position estimates.

The JPDAF is capable of using multiple observations at each time step \( k \), and making a probabilistic association between observations and active tracks.
JPDAF Schematic
Speaker Tracking Demonstration

Camera 1

Camera 2

Camera 3

Camera 4
Part VI
Digital Filter Banks
Sampling and Aliasing

Below the Nyquist Rate

1. Original Signal: $X_1(\omega)$
2. Sampling: $S(\omega)$
3. Discrete Signal: $X_d(\omega)$
4. Lowpass Filtering: $H_{LP}(\omega)$
5. Reconstructed Signal: $X_r(\omega)$

Above the Nyquist Rate

1. Original Signal: $X_1(\omega)$
2. Sampling: $S(\omega)$
3. Discrete Signal: $X_d(\omega)$
4. Lowpass Filtering: $H_{LP}(\omega)$
5. Reconstructed Signal: $X_r(\omega)$

Aliasing
- Power vs Frequency
Consider a bank of M discrete-time filters.

The impulse response of the m-th filter is

\[ h_m[n] = h_0[n]e^{j\omega_m n} \]

where

\[ \omega_m = \frac{2\pi}{M} \]

for \( m = 0, \ldots, M-1 \) and \( h_0[n] \) is the filter bank prototype.

The output of the m-th filter is

\[ y_m[n] = \sum_{r=-\infty}^{\infty} x[r]h_m[n-r] = \sum_{r=-\infty}^{\infty} x[r]h_0[n-r]e^{j\omega_m(n-r)} \]
Digital Filter Bank

- This implies that in the frequency domain

\[ H_m(e^{j\omega}) = H_0(e^{j(\omega-2\pi m/M)}) \]

- Similarly in the z-transform domain

\[ H_m(z) = H_0(zW_M^m) \]

where \( W_M \) is the \( M \)th root of unity.

- We also consider a set of synthesis filter banks with the form

\[ g_m[n] = g_0[n]e^{j\omega_m n} \]

which implies

\[ G_m(z) = G_0(zW_M^m) \]
Shown in the figure is a *direct form* implementation of a uniform DFT filter bank.
Digital Filter Bank

- The properties of the filter bank are determined by the analysis and synthesis prototypes.
- A particularly simple prototype impulse response is given by

\[
    h_0[n] = \begin{cases} 
        1, & 0 \leq n \leq M - 1, \\
        0, & \text{otherwise.}
    \end{cases}
\]

- In the z-transform domain,

\[
    H_0(z) = 1 + z^{-1} + \cdots + z^{-(M-1)} = \frac{1 - z^M}{1 - z}
\]

- The frequency response of the simple prototype is

\[
    H_0(z) = \frac{\sin(M\omega/2)}{\sin(\omega/2)} e^{j\omega(M-1)/2}
\]
The simple prototype provides only 13 dB of suppression between the main and first sidelobe.
The analysis filter bank can be implemented as shown in the figure, where $\mathbf{W}$ is the discrete Fourier transform matrix,

$$[\mathbf{W}]_{mn} = W_{M}^{-mn} = e^{j2\pi mn/M}$$
The short-time Fourier transform can be represented pictorially as shown in the figure.

Such windows are used for speech recognition to isolated segments of 15 to 25 ms duration.

In a digital filter bank, the length of the window is typically several times the number of subbands.

This provides better stopband suppression and a smaller transition band.
Consider a filter

\[ H(z) = \sum_{n=-\infty}^{\infty} h[2n]z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h[2n+1]z^{-2n} \]

Upon defining the polyphase components

\[ E_0(z) = \sum_{n=-\infty}^{\infty} h[2n]z^{-n}, E_1(z) = \sum_{n=-\infty}^{\infty} h[2n+1]z^{-n} \]

\( H(z) \) can be rewritten as

\[ H(z) = E_0(z^2) + z^{-1}E_1(z^2) \]
Type 1 and 2 Polyphase Representations

- For arbitrary $M$, the Type 1 polyphase representation of $H(z)$ is

$$H(z) = \sum_{n=0}^{M-1} z^{-n} E_n(z^M),$$

where

$$E_n(z) = \sum_{m=-\infty}^{\infty} h[mM + n] z^{-m}.$$  

- The Type 2 polyphase representation of $H(z)$ is

$$H(z) = \sum_{n=0}^{M-1} z^{-(M-1-n)} R_n(z^M),$$

where

$$R_n(z) = E_{M-1-n}(z).$$
The uniform DFT filter bank admits the polyphase representation shown in the figure.

This representation has important implications.
As it stands, the uniform DFT filter bank will produce a $M$-fold increase in the data rate.

In order to reduce this massive increase, the outputs of the analysis filters are decimated, such that

$$y_D[n] = x[nM].$$

In the frequency domain,

$$Y_D(e^{j\omega}) = \frac{1}{M} \sum_{m=0}^{M-1} x(e^{j(\omega-2\pi m)/M}).$$
Effect of Decimation

Original Spectrum

\( X(e^{j\omega}) \)

Stretched Spectrum

\( X(e^{j4\omega}) \)

Shifted Copy of the Stretched Spectrum

\( X(e^{j2\omega}) \)

Shifted Copy of the Stretched Spectrum

\( X(e^{j4\omega}) \)

Sum of Stretched Spectrum and Shifted Copies

\( 3Y(e^{j\omega}) \)
An $L$-fold expander takes as input $x[n]$ and produces as output

$$y_E[n] = \begin{cases} 
  x[n/L], & \text{if } n \text{ is an integer - multiple of } L, \\
  0, & \text{otherwise.}
\end{cases}$$

In the z-transform domain,

$$Y_E(z) = X(z^L).$$
Effect of Expansion

Original Spectrum

\[ X(e^{j\omega}) \]

-8/3π -2π -4/3π -2/3π 0 2/3π 4/3π 2π 8/3π

Expanded Spectrum

\[ Y_E(e^{j\omega}) \]

-8/3π -2π -4/3π -2/3π 0 2/3π 4/3π 2π 8/3π
The *noble identities* can be represented as shown in the figure.

These identities are very useful in the theory and practice of digital filter banks.

**Identity 1**

\[
x[n] \rightarrow M \rightarrow G(z) \rightarrow y_1[n] \equiv x[n] \rightarrow G(z^M) \rightarrow M \rightarrow y_2[n]
\]

**Identity 2**

\[
x[n] \rightarrow G(z) \rightarrow L \rightarrow y_3[n] \equiv x[n] \rightarrow L \rightarrow G(z^L) \rightarrow y_4[n]
\]
Using the noble identities, the final form of the analysis/synthesis filter bank can be represented as shown in the figure. This form is computationally efficient in that all components run at the lowest possible rate.
Part VII

Beamforming
Beamforming is the combination of the several signals at the output of a microphone array such that the desired source is maintained while noise is suppressed.

Beamforming exploits the physical geometry of the microphone array and the finite speed of sound.

The simplest beamformer is the \textit{delay-and-sum beamformer}, which considers only the position of the desired source.
Frequency-Dependent Beam Pattern

Delay-and-Sum Beam Pattern

Superdirective Beam Pattern
Adaptive beamforming algorithms optimize some criterion subject to a distortionless constraint.

Minimizing mean square error can null out an interfering signal.
The generalized sidelobe canceller (GSC) is a popular form for an adaptive beamformer.

- The quiescent weight vector $\mathbf{w}_q$ is set to fulfill a distortionless constraint in the look direction.
- The blocking matrix $\mathbf{B}$ is orthogonal to $\mathbf{w}_q$, such that $\mathbf{B} \mathbf{w}_q = \mathbf{0}$.

$$
\begin{align*}
\mathbf{X}(\omega) & \xrightarrow{\mathbf{w}_q^H} Y_c(\omega) + Y_b(\omega) \\
& \xleftarrow{\mathbf{B}_a^H \mathbf{Z}(\omega)} \xrightarrow{\mathbf{w}_a^H} \mathbf{Y}(\omega)
\end{align*}
$$
Conventional adaptive beamforming algorithms are typically designed to minimize a squared-error criterion, such as output power, subject to a distortionless constraint.

Such a design was originally developed for operation in a *free field*.

Real acoustic environments are very seldom well-represented by a free field.

Minimizing variance subject to a distortionless constraint can lead to *signal cancellation*.

Hence, a different optimization criterion is needed for acoustic beamforming prior to speech recognition.
The magnitude of subband speech samples can be modeled with the generalized Gaussian pdf:

\[
p_{GG}(y) = \frac{1}{2\Gamma(1+1/f)B(f)\sigma} \exp\left[ -\frac{y}{B(f)\sigma} \right]^{f}
\]

where

\[
B(f) = \left[ \frac{\Gamma(1/f)}{\Gamma(3/f)} \right]^{1/2}
\]
The GG with \( f = 1 \) corresponds to the Laplacian pdf.

Setting \( f = 2 \) yields the conventional Gaussian pdf.
Kurtosis is defined as

$$kurt(Y) = E\{Y^4\} - 3(E\{Y^2\})^2$$

Like negentropy, kurtosis is a measure of how non-Gaussian a pdf is.
The Gaussian pdf has zero kurtosis.
Super-Gaussian pdfs have positive kurtosis; sub-Gaussians pdfs have negative kurtosis.
The scale and shape factors can be learned from training data.

The optimal factors are subband-dependent.

Shown in the figure are the parameters of the GG pdf for each frequency bin versus the scale factor the shape factor $f$, and the kurtosis.
The entropy for a continuous complex-valued r.v. $Y$, is defined as

$$H(Y) = -\int p_Y(y) \log p_Y(y)dy = -E\{\log p_Y(y)\}$$

As mentioned previously, a Gaussian variable has the largest entropy among all r.v.s of equal variance.

The entropy of $Y_{gauss}$ can be expressed as

$$H_G(Y) = 1 + \log \pi \sigma^2$$

Negentropy $J$ for a complex-valued r.v. $Y$ is defined as

$$J(Y) = H(Y_{gauss}) - H(Y)$$

where $Y_{gauss}$ is a Gaussian variable which has the same variance as $Y$.

Note that negentropy is non-negative, and it is zero if and only if $Y$ has a Gaussian distribution.
Our experiments were based on four passes of speech recognition:

1\textsuperscript{st} pass: no speaker adaptation
2\textsuperscript{nd} pass: VTLN & CMLLR
3\textsuperscript{rd} pass: VTLN, CMLLR & MLLR
4\textsuperscript{th} pass: VTLN, CMLLR & MLLR & ML-SAT.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Word error rate (WER)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1\textsuperscript{st} pass</td>
</tr>
<tr>
<td>SDM</td>
<td>87.0 %</td>
</tr>
<tr>
<td>D&amp;S BF</td>
<td>80.1 %</td>
</tr>
<tr>
<td>GEV BF</td>
<td>78.7 %</td>
</tr>
<tr>
<td>MK BF (proposed)</td>
<td>76.6 %</td>
</tr>
<tr>
<td>MN BF (proposed)</td>
<td>75.1 %</td>
</tr>
<tr>
<td>CTM (ref.)</td>
<td>52.9 %</td>
</tr>
</tbody>
</table>
Yet another advantage of MN (and MK) beamforming

- Maximum negentropy (or kurtosis) beamforming can coexist with conventional adaptive beamforming techniques.

The conventional adaptive beamformer can be used instead of delay-and-sum beamformer

Find the active weight vector which provides the maximum negentropy of beamformer’s outputs
Minimum Variance Distortionless Response (MVDR) beamforming

Minimum variance distortionless response (MVDR) beamforming is an alternate implementation of the GSC beamformer which minimizes beamformer’s outputs.

Solution for the MVDR beamformer

With a noise covariance matrix $R_N$, the weight vector of the MVDR beamformer is obtained by finding

$$\arg\min_{w_m} w_m^H R_N w_m$$

subject to the distortionless constraint for the look direction

$$B_m^H d_m = 0.$$

The solution, which is equivalent to the GSC, can be expressed as

$$W_{mvdr,m} = \frac{R_N d_m}{d_m^H R_N d_m}.$$
Advantages of the super-directive beamformer:

- It is data-independent and does not require the measurements of the noise data.
Block diagram of the $m$th band of the hybrid beamformer

*For the sake of simplification, weights are expressed as a vector.

Multi-channel input $X_m$

- Active weight vector for maximizing negentropy of beamformers outputs

- Super-directive beamformer

- Blocking matrix for maintaining the distortionless constraint so that $B_m^H w_{sd,m} = 0$

- By maximizing the *negentropy* of beamformer’s outputs $\{Y_m\}$ while maintaining the distortionless constraint, undesired signals are suppressed.
## Recognition Results

<table>
<thead>
<tr>
<th>Beamforming algorithm</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2(^{nd}) pass</td>
</tr>
<tr>
<td>D &amp; S beamforming</td>
<td>39.9 %</td>
</tr>
<tr>
<td>MVDR beamforming (MMSE)</td>
<td>35.4 %</td>
</tr>
<tr>
<td>Super-directive beamforming</td>
<td>31.9%</td>
</tr>
<tr>
<td>Conventional MN beamforming</td>
<td>32.7%</td>
</tr>
<tr>
<td><strong>MN beamforming</strong></td>
<td>32.1%</td>
</tr>
<tr>
<td><strong>with a super-directive beamformer</strong></td>
<td></td>
</tr>
<tr>
<td>CTM</td>
<td>21.5 %</td>
</tr>
</tbody>
</table>

- The best recognition performance is obtained by using the super-directive beamformer in the upper branch.

- Results will be submitted to EUSIPCO 2010.
Summary

I - Applications of DSR

II - Characteristics of Human Speech

III - The Acoustic Environment

IV - Speech feature enhancement

V - Speaker Tracking

VI - Digital Filter Banks

VII - Beamforming