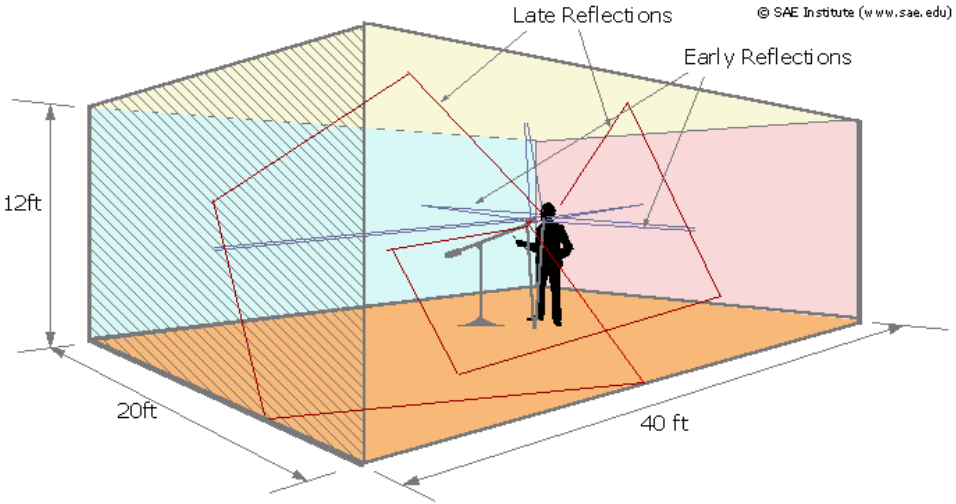
Speech Dereverberation

Scott Bradley and James Fosburgh

What is it?

- Process of removing reverberation from a speech signal
- Reverberation
  - Caused by reflections of a source signal
  - Reflections echo the original signal and decay over time
  - Example: room acoustics
- Characteristics
  - Reverberated noise is similar to clean signal
  - Same reverberation in the same space



Applications

- Telecommunications
  - Phone calls
  - Skype
  - Facetime
- Music Production
  - Vocal enhancement
- Automatic Speech Recognition
  - Assistants
    - Siri
    - Google
    - Alexa
  - Speech to text

Challenges

- Unwanted noise similar to clean signal
  - Echoed speech is similar to the clean speech
  - Difficult to identify echo from source
- Each acoustic environment will have different characteristics
  - RT60
  - Filtering

Solutions

- Inverse Filtering
- Spectral Subtraction
- Neural Networks

Inverse Filtering

- Uses Room Impulse Response (RIR)
- Room acoustics act as LTI filter
- Inverse of the RIR applied to the signal will completely negate reverberation

Inverse Filtering

1) Magnitude and phase of RIR

$$H'(\omega) = |H(\omega)| \exp[i\phi(\omega)].$$

1) Minimum Phase Component

$$\phi(\omega) = \phi_m(\omega) + \phi_a(\omega).$$

1) Minimum phase and Allpass parts

$$H(\omega) = M(\omega)A(\omega),$$

where

$$M(\omega) = |H(\omega)| \exp[i\phi_m(\omega)],$$

and

$$A(\omega) = \exp[i\phi_a(\omega)].$$

1) Inverse Filter

$$G(k) = 1/M(k),$$

Challenges

- Requires perfect RIR
  - Any imperfections will cause noise and distortion in the output
  - Only works if you can accurately measure the RIR
- Realistic rooms have nonminimal phase RIRs
- Does not work for a source and receiver separation greater than 8 in
  - Nonminimum phase effect > 8 in
- Even authors suggest using other techniques

Spectral Subtraction

- Depends on overlap-masking
  - Energy of previous speech smoothed over time
  - Causing noise on primary speech
- Breaks speech into phonemes
  - Individual impulses that combine to form speech
- Identifies past phonemes
- Estimates reverberation time of acoustic environment
  - RT60 of RIR
- Removes past phonemes echoed onto current clean speech signal

Spectral Subtraction

Simple Version

$$
\begin{aligned}
|\hat{S}(m,k)| &= |X(m,k)| - \hat{\gamma}_{rr}^{1/2}(m,k) \\
&= G(m,k)|X(m,k)|,
\end{aligned}
$$

$$
\begin{aligned}
G(m,k) &= \frac{|X(m,k)| - \hat{\gamma}_{rr}^{1/2}(m,k)}{|X(m,k)|} \\
&= 1 - \frac{1}{\sqrt{SNR_{pos}+1}},
\end{aligned}
$$

$$
SNR_{pos} = \frac{|X|^2}{\hat{\gamma}_{rr}} - 1,
$$

$$
\hat{\gamma}_{rr}(m,k) = \mathrm{e}^{-2\Delta T}\hat{\gamma}_{xx}(m-T,k).
$$

$$
\hat{\gamma}_{xx}(m,k) = \beta\hat{\gamma}_{xx}(m-1,k) + (1-\beta)|X(m,k)|^2.
$$

Spectral Subtraction

Improvements

1) Smoothing Negative PSD

$$SNR_{pri}(m,k) = \beta SNR_{pri}(m-1,k) + (1-\beta)\max\left[0, SNR_{pos}\right].$$

1) Spectral subtraction floor

$$G(m,k) = \begin{cases} 1 - \frac{1}{\sqrt{SRN_{pri}+1}} & \\ \qquad \text{if } |\hat{S}(m,k)| \geq \lambda\sqrt{\hat{\gamma}_{rr}(m,k)}, & \\ \lambda\sqrt{\hat{\gamma}_{rr}(m,k)}/|X(m,k)| \qquad \text{otherwise.} \end{cases}$$

Spectral Subtraction

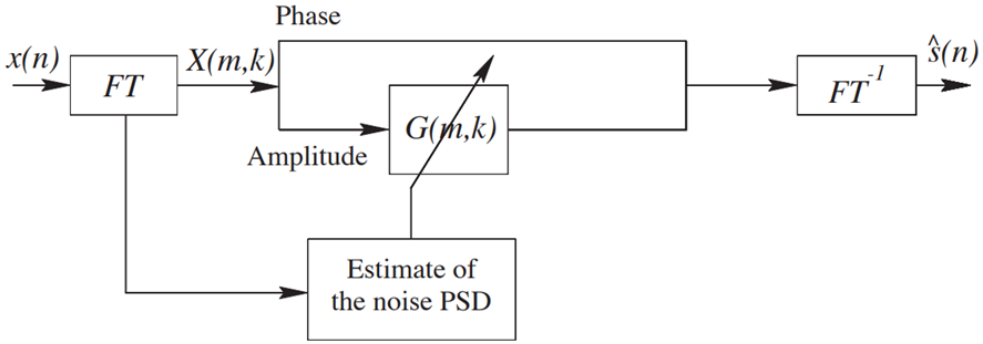


Figure 1. Overview of the algorithm.

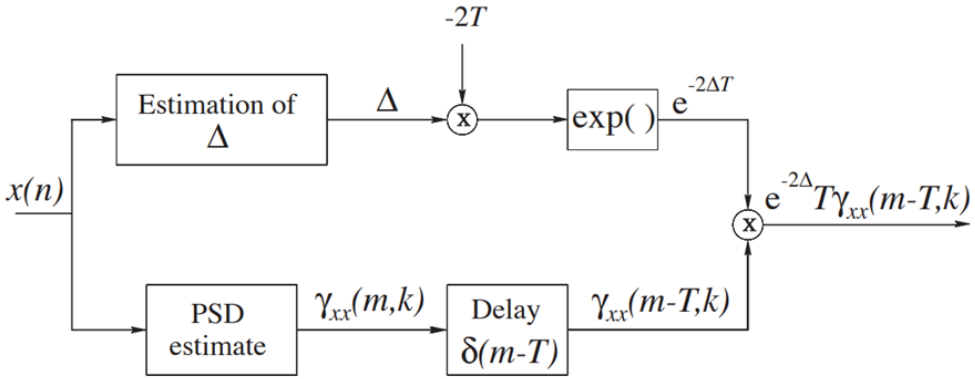


Figure 2. Estimation of the reverberant PSD.

Spectral Subtraction

Challenges

- Only accounts for reverberant energy
  - Any other noise will affect this systems performance
  - Unrealistic to have a perfect environment where the only noise is reverberation
- Still leaves behind residual musical noise
- Only intended for speech recognition applications
- Quality of clean speech not accounted for or solved

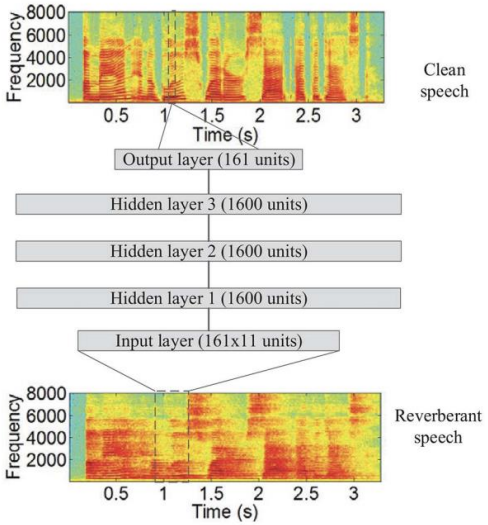
DNN Spectral Mapping

Han et al

Deep Neural Net used to learn mapping between reverberant/noisy speech and clean signal

Steps:

1. Pre-Processing
2. Model Training
3. Post-Processing



DNN Spectral Mapping

Han et al

Pre-Processing - Convert time signal to spectrogram

$$\mathbf{x}(m) = [X(m,1), X(m,2), \ldots, X(m,161)]^T$$

Model Training - Input to DNN

$$\tilde{\mathbf{x}}(m) = [\mathbf{x}(m-d), \ldots, \mathbf{x}(m), \ldots, \mathbf{x}(m+d)]^T$$

Model Training - Error Function

$$\mathcal{L}(\mathbf{y}, \mathbf{x}; \mathbf{\Theta}) = \sum_{c=1}^{C} (y_c - f_c(\mathbf{x}))^2$$

DNN Spectral Mapping

Han et al

Post-Processing

Straight reconstruction doesn't account for phase

Iterative reconstruction used taking original phase into account

Algorithm 1 Iterative signal reconstruction

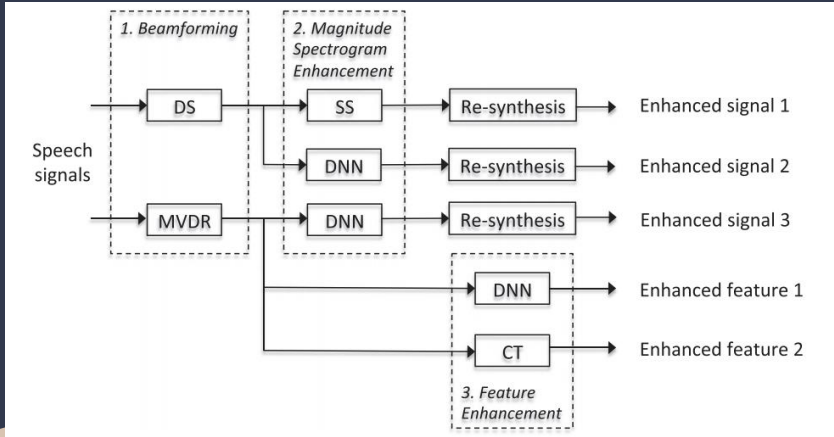
Input: Target magnitude Y^0, noisy phase ϕ^0 and iteration number N

Output: Time-domain signal s

$Y \leftarrow Y^0, \phi \leftarrow \phi^0, n \leftarrow 1$
while $n \leq N$ **do**
  $s^n \leftarrow \text{iSTFT}(Y, \phi)$
  $(Y^n, \phi^n) \leftarrow \text{STFT}(s^n)$
  $Y \leftarrow Y^0$
  $\phi \leftarrow \phi^n$
  $n \leftarrow n + 1$
end while
$s \leftarrow s^N$

DNN Using Dynamic Features

Xiao et al



Similar to Han et al but with additions made to the process:

- Beamforming
- Goal dependent inputs to DNN
- Changes to DNN to increase continuity

DNN Using Dynamic Features

Xiao et al

Beamforming used to pre-process input to DNN

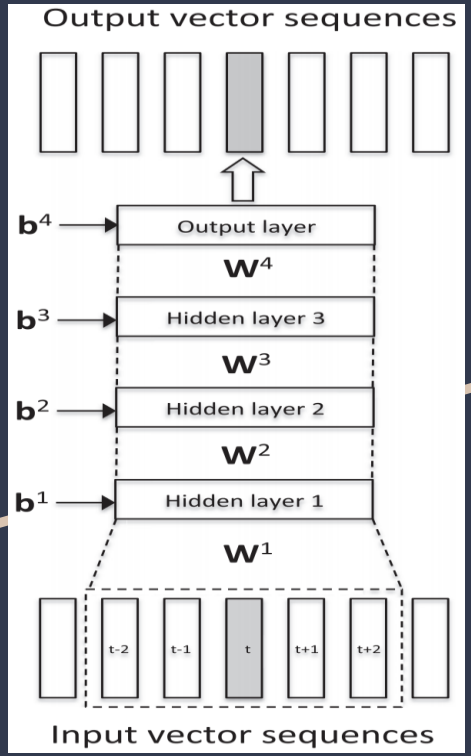
Acts as spatial filter to improve clarity of input signal with respect to target

Two methods tested:

- Delay and Sum (DS)
- Minimum Variance Distortionless Response

DNN Using Dynamic Features

Xiao et al



DNN Training

Restrictive Boltzmann machine used to initialize DNN coefficients

Input depends on goal:

- Speech Enhancement - Spectrogram
- ASR Improvement - MFCCs

DNN Using Dynamic Features

Xiao et al

DNN Improvements

- Least-Squares estimation
  - Delta (first-order time derivative) and Acceleration (second-order time derivative) feature vectors added to DNN target
  - Improves continuity, but more computationally expensive
- Sequential Training
  - Delta and Acceleration values added to DNN cost function