Function Documentation

Overview

The **denoise** function is a MATLAB implementation for denoising an input audio signal by estimating its noise profile and applying a time-vary Wiener filter. The function divides the input signal into segments, calculates the Zero Crossing Rate (ZCR) for each segment, and finds the maximum ZCR value to estimate the noise profile. Then, it applies the Wiener filter to each segment using the estimated noise profile and concatenates the filtered segments to form the denoised output signal.

Inputs

The function takes the following input arguments:
y: The input audio signal to be denoised. It should be a one-dimensional array representing the audio waveform.
fs: The sampling rate of the audio signal (in Hz).
noiseLengthSec (optional): The length of the noise profile in seconds. Default value is 3.0.
nfft (optional): The number of points in the FFT (Fast Fourier Transform). Default value is 4096.
noverlap (optional): The number of points of overlap between segments in the FFT. Default value is **nfft/2**.

Output

The function returns the following output:
xhat: The denoised version of the input audio signal **y**. It is a one-dimensional array representing the denoised audio waveform.

Algorithm

The **denoise** function works in the following steps:

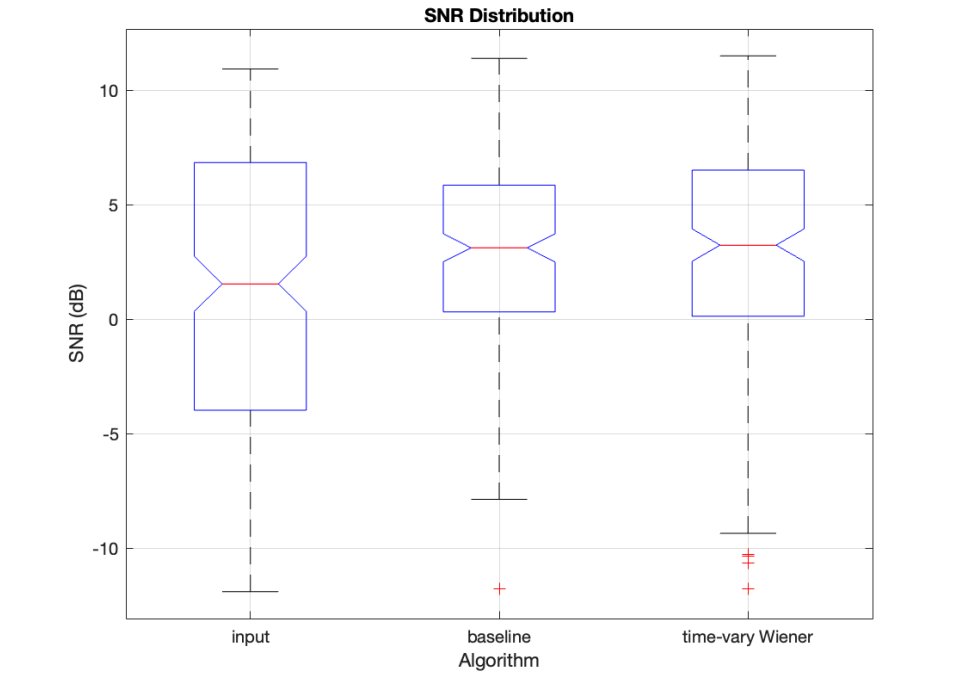
1. Initialize default values for optional input arguments if not provided by the user.
2. Divide the input audio signal **y** into segments of length **seg**.
3. For each segment, perform the following steps:
   - a. Calculate the Zero Crossing Rate (ZCR) for each frame within the segment.
   - b. Determine the maximum ZCR value and its corresponding index among the frames.
   - c. Estimate the noise profile based on the maximum ZCR value and its index.
   - d. Apply the Wiener filter to the segment using the estimated noise profile.
   - e. Store the denoised segment in **xhat**.
4. Process any remaining samples after the last complete segment.
5. Return the concatenated denoised segments as **xhat**.

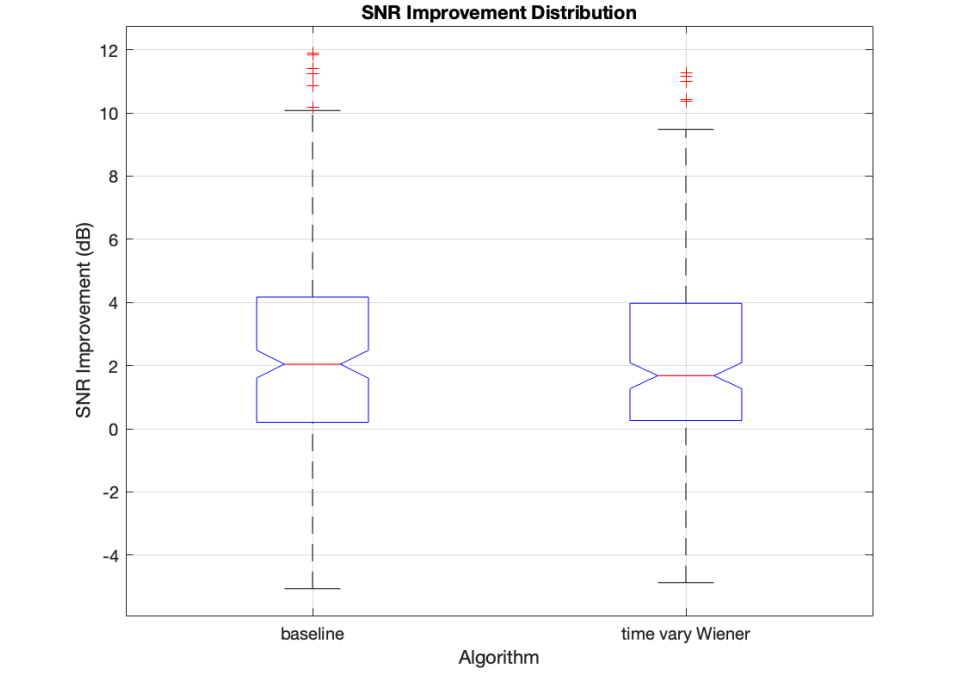
Example Usage

```
% Load an audio file
[y, fs] = audioread('input_audio.wav');
% Set optional parameters
noiseLengthSec = 3.0;
nfft = 4096;
noverlap = nfft / 2;
% Denoise the audio signal
xhat = denoise(y, fs, noiseLengthSec, nfft, noverlap);
% Save the denoised audio signal to a file
audiowrite('output_audio.wav', xhat, fs);
```

Results

Output SNR	min	median	mean	max	std
Baseline	-11.79	3.12	2.98	11.39	4.25
Our Algorithm	-11.79	3.24	2.70	11.50	4.89





SNR Change	min	median	mean	max	std
Baseline	-5.06	2.05	2.55	11.90	3.37
Our Algorithm	-4.87	1.68	2.27	11.27	3.01

Reference & Toolbox

Toolbox: Voicebox (http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html)

Contribution

Jiaying: implement the spectral subtraction method, write the documentation;

Joy: implement the time-varying Wiener filter method, write the comments;

Sawyer: fix the grammar and typo in the documentation and read through the code.