Speech Enhancement Using Wiener Filtering and PSR Phase Reconstruction

Geet Khatri

https://github.com/geetkhatri/

CHAPTER 1
INTRODUCTION

1.1. Speech Enhancement

Speech enhancement refers to the estimation of a desired speech signal from a noisy observation in order to increase the intelligibility of the speech signal.

Speech enhancement is required in many applications such cellular phones, hands-free communication, teleconferencing, hearing aids, speech recognition and audio fingerprinting among others.

If a speech signal is corrupted by additive noise, the goal of speech enhancement is to find an optimal estimate $\hat{s}(n)$ of the clean speech signal $s(n)$, given a noisy observation $x(n)$:

$$x(n) = s(n) + v(n) \tag{1.1}$$

where $v(n)$ is the additive noise.

1.1.1. Noise in Speech Systems

The presence of noise affects the underlying speech signal in several ways:

- It adds new, unwanted frequency components, causing *spectral masking*.
- It interferes with the dynamic properties of the underlying speech signal, causing a reduction in the dynamic range of the speech signal. The attacks in the sound also become less prominent.

These changes can greatly affect our perception of the speech. Thus, it is necessary to enhance the noisy signal in some way so that the intelligibility of the underlying speech is improved.

Noise affecting speech signals can be roughly classified as:

- **Microphone-related noise**: The *self-noise* of a microphone is the noise the microphone produces of itself, even when no sound source is present. It is specified in dB-A.[1]
- **Electrical noise**: This includes electromagnetically induced noise and thermal noise. Electromagnetic radiation present in the environment can cause interference in a conductor through the phenomenon of electromagnetic induction. Thermal noise is inherent to the conductor.
- **Environmental noise**: It refers to the unwanted sounds present in the environment. The types of unwanted sounds present in the environment can vary heavily. Examples of common sources of environmental noise include fans, traffic and other speakers in the vicinity.

While the use of better equipment can reduce microphone-related noise and electrical noise, it is much harder to reduce the effect of environmental noise. Use of directional microphones and beamformers can help reduce the effect of environmental noise to a certain extent.

One of the challenges associated with noise reduction is to reduce noise without disturbing the unvoiced and noise-like components of the speech signal itself.

1.1.2. Types of Speech Enhancement Algorithms

Speech enhancement algorithms can be roughly categorised into three classes:

- **Filtering**: In these algorithms, the noisy signal is filtered. The output of the filter gives the estimate of the clean speech signal. An example of such a filter is the Wiener filter, which gives an MMSE estimate of the speech signal.
- **Spectral restoration**: These algorithms estimate the spectrum of the clean speech signal. An example of such an algorithm is the MMSE-STSA estimator in which an MMSE estimate of the short-time spectral amplitudes (STSA) is obtained.
- **Speech-model-based**: Usually, these algorithms involve estimation of the parameters of the speech model.

---

[1] The “A” stands for A-weighting, which is applied to instrument-measured sound levels in order to account for the differences in the loudness of different frequencies as perceived by the human ear.

1.2. Voiced and Unvoiced Speech

Speech is composed of phonemes. The phonemes are produced by the vocal cord and the vocal tracts.

Voiced speech is produced when the vocal cords vibrate during the utterance of a phoneme such as */a/*, */e/*, */z/* or */v/*. Unvoiced speech does not involve the use of vocal cords. The utterance of the phonemes */s/*, */f/*, */t/* and */p/* are examples of unvoiced speech.

Voiced speech tends to be louder, whereas unvoiced speech tends be more abrupt.

1.3. Frequency Range of Speech

The voiced speech of a typical adult male has a fundamental frequency in the range 85 to 180 Hz, and that of a typical adult female has a fundamental frequency in the range 165 to 255 Hz.

In telephony, the voice frequency band ranges from approximately 300 Hz to 3400 Hz. Thus, the fundamental frequency of most speech falls below the bottom of the voice frequency band. However, the presence of the upper harmonics creates the impression of the presence of the fundamental frequency to the listener. This missing fundamental frequency, which is perceived by the listener even though it is not actually present, is known as *phantom fundamental*.

A lot of the information in a speech signal can be represented by the harmonics (fundamental frequencies plus the corresponding overtones). However, the unvoiced parts of the speech cannot be represented well by the harmonics alone.

If it's necessary to capture all the details, the entire audible frequency range (20 Hz to 20 kHz) must be considered. However, most of the information in speech signals can be represented using a bandwidth of about 8 kHz. Thus, a sampling frequency of 16 kHz is sufficient for most speech processing tasks.

CHAPTER 2
SPEECH MODELING

There are several different ways of modeling speech signals. Two speech models are discussed in this chapter:

- Autoregressive Model
- Harmonic Model

2.1. Autoregressive Model

In the autoregressive (AR) model, the speech signal $s(n)$ is modeled as the sum of a linear combination of past inputs and an excitation signal $e(n)$:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + e(n) \tag{2.1}$$

where $a_1, a_2, \ldots, a_p$ are the **linear prediction coefficients (LPCs)**. $e(n)$ is also known as the *prediction error*.

Equation (2.1) is a constant-coefficient difference equation describing an *all-pole system* whose system function is:

$$H(z) = \frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \tag{2.2}$$

The vocal tract can be thought of as a filter, whose system function is $1/A(z)$, that filters the excitation signal $e(n)$ generated by the larynx. When the speech is voiced, $e(n)$ can be modeled as quasi-periodic pulses. When the speech is unvoiced, $e(n)$ can be modeled as white noise. This is illustrated in figure 2.1.

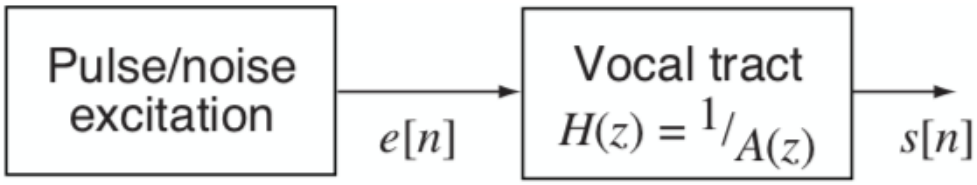


Fig. 2.1. Autoregressive model of speech

There are several advantages of using the AR model:

- It is mathematically tractable, since the speech signal is modeled as the output of an infinite impulse response (IIR) filter whose input is pulses or noise.
- It has an intuitive physical interpretation.
- It is quite accurate.
- It changes relatively slowly. Generally, $A(z)$ is updated every 10-20 ms.

A disadvantage of the AR model is that *it does not take into account zeros introduced by nasals*.

2.1.1. Estimation of LPCs

LPCs are estimated by minimizing the mean squared prediction error with respect to each $a_i \; \forall \; i \in \{1,2,...,p\}$.

The mean squared prediction error is:

$$\varepsilon = E[e^2(n)] \tag{2.3}$$

or,

$$\varepsilon = E\left[\left(s(n) - \sum_{i=1}^{p} a_i s(n-i)\right)^2\right] \tag{2.4}$$

To minimize ε, we differentiate it with respect $a_i \; \forall \; i \in \{1,2,...,p\}$ to get a set of p linear equations, which can be written in matrix notation as:

$$\begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \dots & R_{ss}(p-1) \\ R_{ss}(1) & R_{ss}(0) & \dots & R_{ss}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}(p-1) & R_{ss}(p-2) & \dots & R_{ss}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ \vdots \\ R_{ss}(p) \end{bmatrix} \tag{2.5}$$

or,

$$Ra = r \tag{2.6}$$

$R_{ss}(l)$ is the autocorrelation of $s(n)$ at lag l:

$$R_{ss}(l) = E[s(n)s(n-l)] \tag{2.7}$$

If R is invertible, equation (2.6) can be rewritten as:

$$a = R^{-1}r \tag{2.8}$$

Equations (2.5) are called the **Yule-Walker equations**.

2.2. Harmonic Model

In the harmonic model, the speech signal $s(n)$ is modeled as a superposition of time-varying sinusoids whose frequencies are integer multiples of the same fundamental frequency. By "time-varying", we mean that the amplitudes, frequencies and phases of the sinusoids vary with time. The integer multiples of the fundamental frequency are called *harmonics*, which is where this model gets its name. $s(n)$ can be written as:

$$s(n) = \sum_{r=1}^{R(n)} A_r(n)\, cos\left(2\pi r f_0(n) n + \varphi_r(n)\right) \tag{2.9}$$

where $f_0(n)$ is the normalized time-varying fundamental frequency[2] and $R(n)$ is the number of harmonics at time instant n.

[2] The normalized fundamental frequency is obtained by dividing the fundamental frequency in Hz by the sampling rate in Hz.

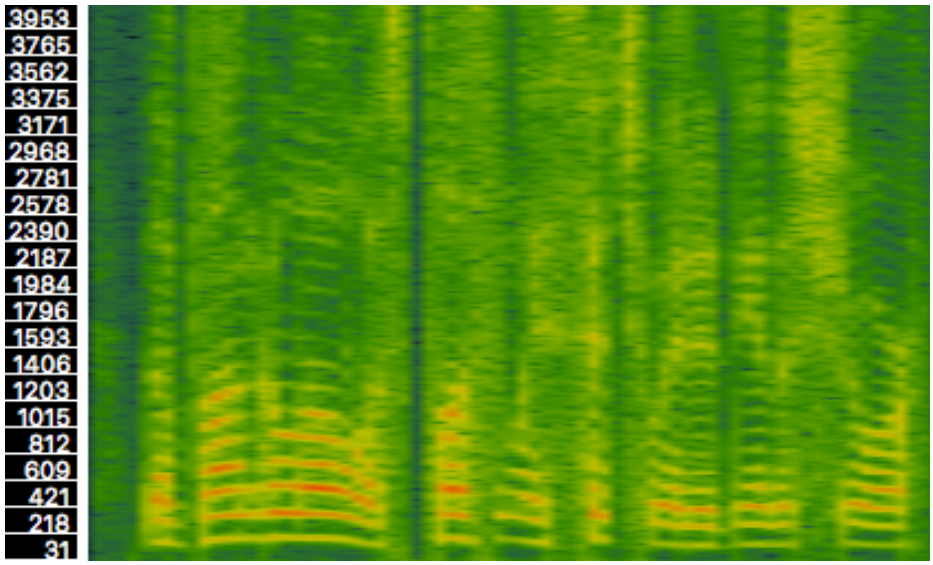


Fig. 2.2. Spectrogram of a speech signal. The horizontal axis represents time and the vertical axis represents frequency (in Hz). Notice how the dominant frequencies in the voiced regions are the fundamental frequency and its integer multiples (harmonics).

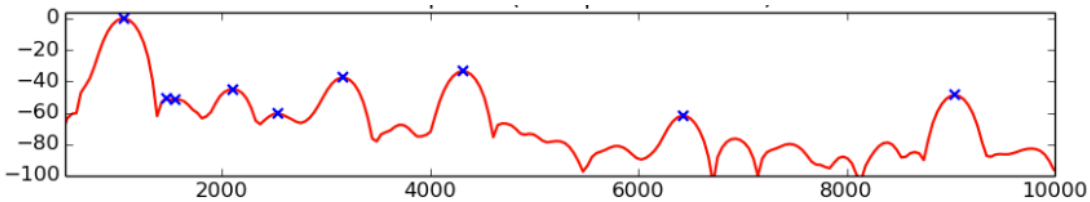


Fig. 2.3. Normalized magnitude spectrum of a short segment of a pitched sound. The frequency is in Hz. Observe that the peaks are at integer multiples of the fundamental frequency

The harmonic model is used to represent monophonic sounds, i.e., sounds that have a single pitch that varies with time. Since the voiced regions of speech are monophonic, the harmonic model is often used to represent speech signals. Figure 2.2 shows how the voiced regions comprise of a single time-varying pitch. Figure 2.3 shows the spectrum of a short segment of a pitched sound. Over the segment, the pitch is approximately constant.

In equation (2.9), the choice of $R(n)$ depends on how much we want to preserve the timbre of the sound. Usually, the upper harmonics tend to have lower energies than the lower harmonics. If we want to include all audible harmonics, the value of $R(n)$ is given by:

$$R(n) = \left\lfloor \frac{f_s}{2f_0(n)} \right\rfloor \tag{2.10}$$

where f_s is the sampling rate of the signal.

The harmonic model has several advantages:

- Once the parameters $f_0(n)$, $A_r(n)$ and $\varphi_r(n)$ have been determined, resynthesis is straightforward. This makes the harmonic model useful for speech synthesis and compression.
- The harmonic model is useful for processing in the time-frequency domain, since it directly represents the changes in the frequency content of the speech signal with time. For example, it may be used when the goal is to shift the pitch of a speech signal without changing its duration.

Some of the disadvantages of the harmonic model are:

- It does not represent the unvoiced regions of speech well.
- It does not take into account *inharmonicity*, which is the degree to which the frequencies of harmonics depart from integer multiples of the fundamental frequency. The effect of inharmonicity is more pronounced for higher harmonics.
- The analysis of signals in the presence of noise can be hard when the harmonic model is used.
- If noise is present in the original signal, the synthesis quality may not be good.

2.2.1 Harmonic + Noise Model

The harmonic model can be modified to take into account the unvoiced parts and noise-like aspects of speech. In the so-called harmonic + noise model (HNM), the speech signal is represented as:

$$s(n) = \sum_{r=1}^{R(n)} A_r(n)\, cos\left(2\pi r f_0(n) n + \varphi_r(n)\right) + u(n)[h(m,n) \star b(n)] \tag{2.11}$$

where $\star$ denotes convolution. The noise component of $s(n)$ can be thought of as being obtained by filtering white Gaussian noise $b(n)$ by a time-varying, all-pole filter $h(m,n)$ and then multiplying the output by an energy

envelope function $u(n)$. In other words, the noise component is basically filtered and time-shaped white Gaussian noise.

The HNM has applications in concatenative text-to-speech synthesis and vocoders. A disadvantage of the HNM is its complexity.

CHAPTER 3
SHORT-TIME FOURIER TRANSFORM (STFT)

3.1 Definition

The short-time Fourier transform (STFT) is used to determine the magnitude and phase spectra of short segments of a signal as the signal varies with time. It is computed by dividing the time-domain signal into short frames, with adjacent frames overlapping by a certain amount, and then computing the discrete Fourier transform (DFT) of each frame. Thus, the STFT is a function of both frequency and time.

The STFT of a discrete-time signal $x(n)$ is given by:

$$X_l(k) = \sum_{n=0}^{N-1} x(n + lH)w(n)e^{-j2\pi kn/N}$$
$$\forall\, l \in \{0,1,...,L-1\}, k \in \{0,1,...,N-1\} \qquad (3.1)$$

where H is the hop size, N is the DFT size, $w(n)$ is the window function, L is the total number of frames, k is the DFT bin index and l is the frame index.

The various parameters in the computation of STFT are explained below:

- **Window size** (M): It is the size of the window function $w(n)$. This means that $w(n) = 0 \quad \forall\, n \notin \{0,...,M-1\}$.
- **DFT size** (N): It is the size of the DFT. It is always greater than or equal to M, since the DFT size must be greater than or equal to the length of the sequence. It is usually a power of 2 to enable the use of fast Fourier

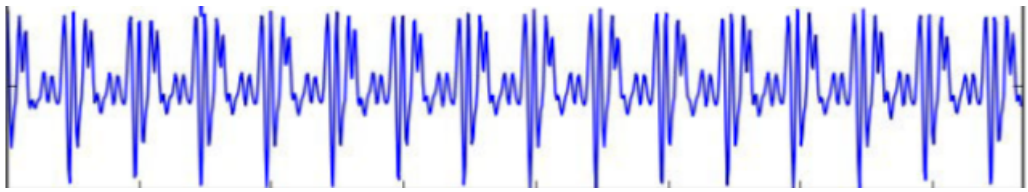
transform (FFT) in the computation of the DFT. For this reason, N is often called the *FFT size*.

- **Hop size** (H): It is the size of the step between two consecutive frames. H is always kept lower than M so that there is overlap between consecutive frames.

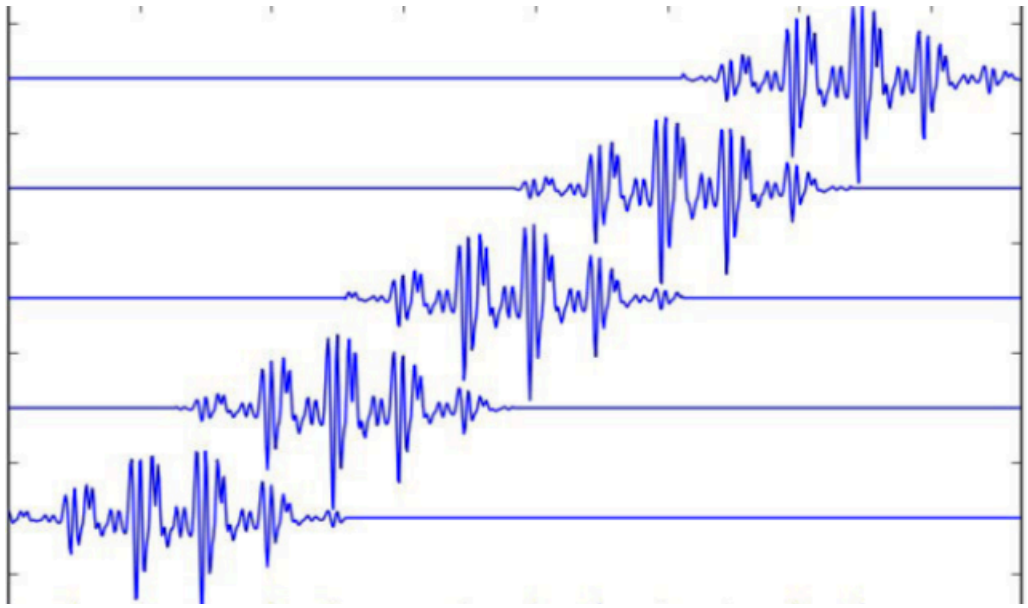
Since speech signals are nonstationary, the STFT is commonly used to represent changes in the frequency content of a speech signal with time.

3.2. Window Functions

Window functions are used to extract frames from the signal. Since the window function has a finite length, the extracted frame also has a finite length equal to the window size.



(a)



(b)

Fig. 3.1. (a) Original signal x(n)
(b) Frames obtained after windowing with window function w(n) (ℓ increases from bottom to top)

To extract a frame from the signal, the signal is shifted in time so that the start of the frame aligns with the start of the window, and then the shifted signal is multiplied with the window function. The l^{th} frame is obtained by multiplying the window function $w(n)$ by a time-shifted version of $x(n)$:

$$x_w(n, l) = x(n + lH)w(n) \tag{3.2}$$

Comparing equations (3.1) and (3.2), it can be seen that the STFT $X_l(k)$ is basically the DFT of the l^{th} frame $x_w(n, l)$.

In the form given in equations (3.1) and (3.2), the window function $w(n)$ is *causal*, meaning that $w(n) = 0 \, \forall n < 0$. From this point onwards, it will be assumed that $w(n)$ is casual.

Some of the commonly used window functions are rectangular window, Hann window, Hamming window, Blackman window and Blackman-Harris window.

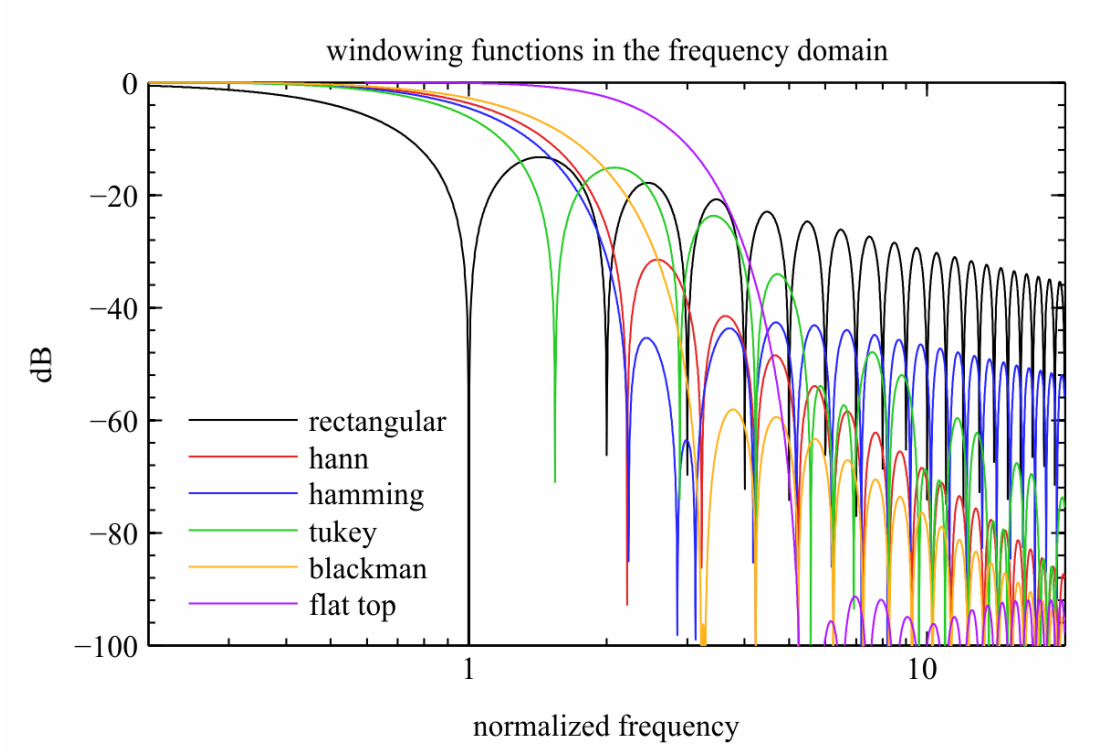


Fig. 3.2. Different window functions in the frequency domain

In general, in the frequency domain, a window function has a main lobe and several side lobes, as shown in Fig. 3.2. The presence of energy in the side lobes, which is undesirable, is called **spectral leakage**.

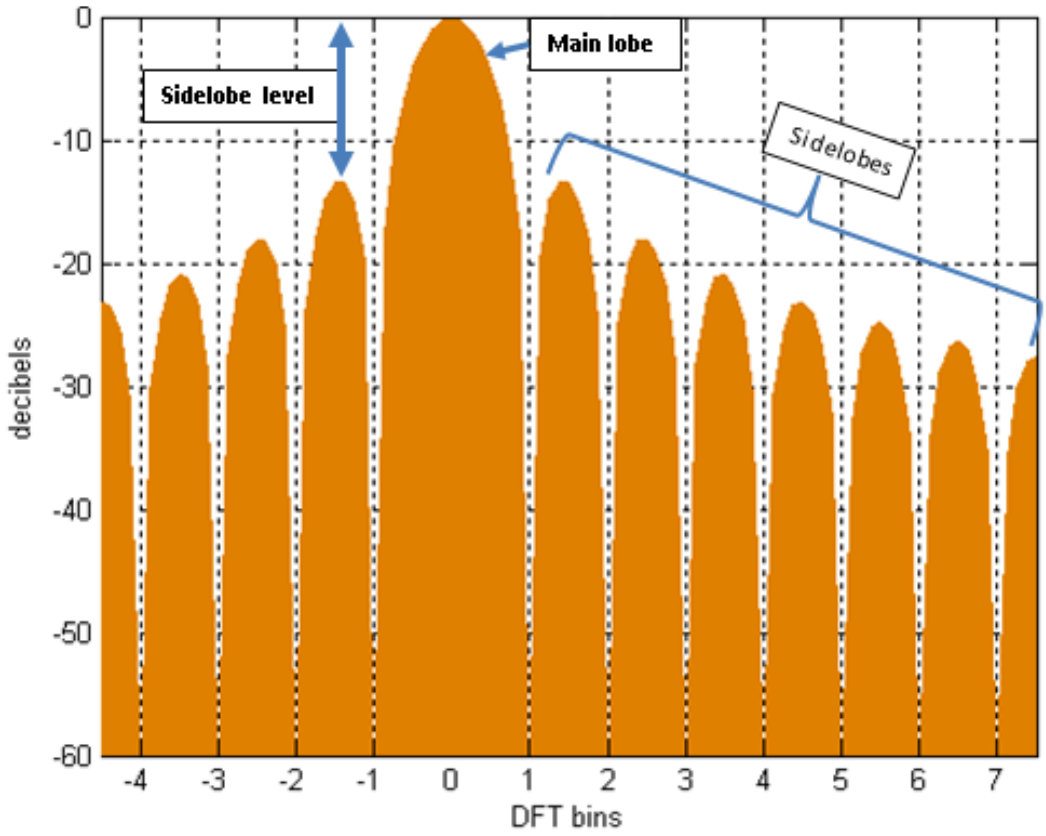


Fig. 3.3. Spectral leakage from the rectangular window

In the frequency domain, each windowed signal can be represented as the convolution of the DFT of the time-shifted version of $x(n)$ and the DFT of the window function $w(n)$.

- When main lobe width is low, the frequency resolution is high, i.e., two closely-spaced sinusoids can be resolved more easily. Thus, *we want the main lobe width to be minimum*.
- When side lobe attenuation is high, the leakage of one frequency component into other frequencies is low. Thus, *we want the side lobe attenuation to be maximum*.

However, these two requirements conflict with each other. This can be seen in figure 3.2. Thus, *there is a tradeoff between side lobe attenuation and*

frequency resolution. The window function must be chosen carefully depending on the requirement.

Two window functions that are commonly used in speech processing are periodic Hamming window and periodic Hann window.

The periodic Hamming window is defined as:

$$w(n) = 0.54 - 0.46\,cos\left(\frac{2\pi n}{M}\right), \quad 0 \leq n < M \tag{3.3}$$

The periodic Hann window is defined as:

$$w(n) = 0.5\left(1 - cos\left(\frac{2\pi n}{M}\right)\right), \quad 0 \leq n < M \tag{3.4}$$

In general, these two window functions offer a good tradeoff between side lobe attenuation and frequency resolution.

Table 3.1. Comparison of window functions

Window function	Side lobe attenuation (dB)	Main lobe width (bins)
Rectangular	-13.3	2
Hann	-31.5	4
Hamming	-42.7	4
Blackman	-58	6
Blackman-Harris	-92	8

3.3. Time-Frequency Compromise

A disadvantage of STFT is that *there is a tradeoff between frequency resolution and time resolution*. If the time resolution is increased, the frequency resolution decreases, and vice versa.

- If a smaller window is used (i.e., M is small), the time resolution is high and the attacks in the sound are better represented, but the frequency resolution low.

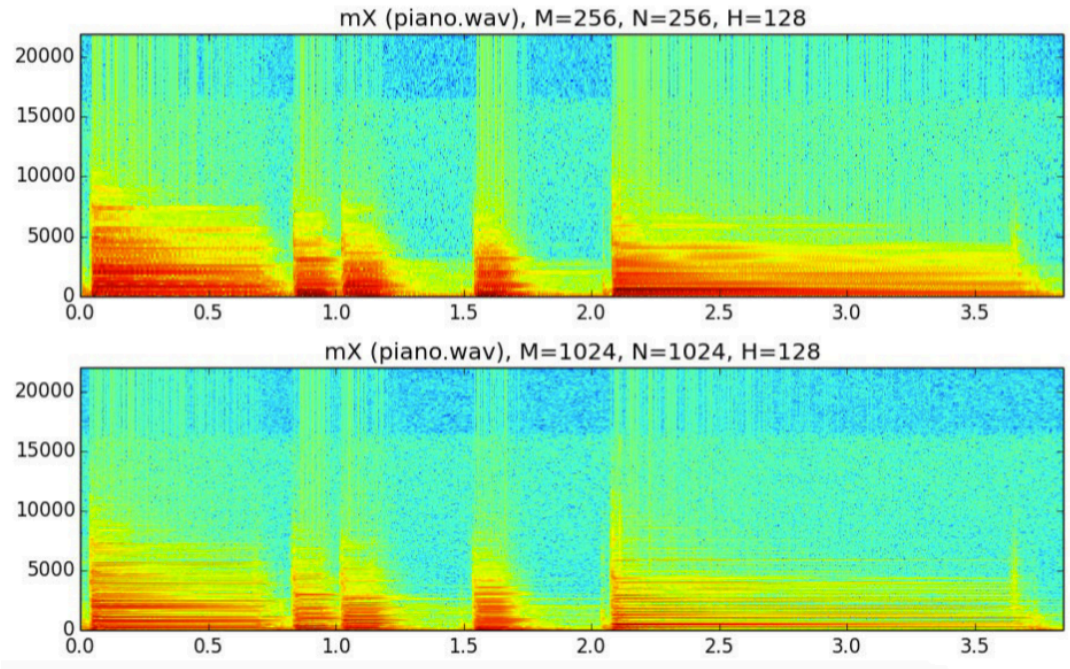


Fig. 3.4. Spectrograms (short-time magnitude spectra) of file 'piano.wav' for M = 256 and M = 1024. Observe that for small M, the frequency resolution is worse but the time resolution is better, and for larger M, the time resolution is worse but the frequency resolution is better.

- If a larger window is used (i.e., M is large), the frequency resolution is high, but the time resolution is low.

Thus, the window size M should be neither too low nor too high.

3.4. Inverse STFT

In order to obtain the original time-domain signal $x(n)$, we take the inverse STFT of $X_l(k)$. The inverse STFT $y(n)$ is computed using the **overlap-add (OLA) method**:

$$y(n) = \sum_{l=0}^{L-1} y_w(n - lH, l) \tag{3.5}$$

where $y_w(n)$ is the inverse DFT of $X_l(k)$:

$$y_w(n,l) = \frac{1}{N}\sum_{k=0}^{N-1} X_l(k)e^{j2\pi nk/N} \tag{3.6}$$

With the right choice of window function $w(n)$ and hop size H, the reconstruction error [i.e., error between $x(n)$ and $y(n)$] is very low.

The window function must have the constant overlap-add (COLA) property at hop size H for the OLA method to work:

$$\sum_{m=-\infty}^{\infty} w(n-mH) = 1 \tag{3.7}$$

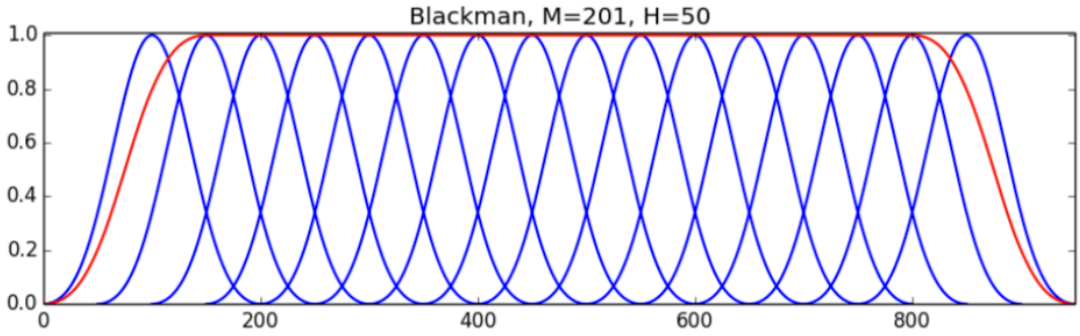


Fig. 3.5. COLA property of Blackman window of size 201 at hope size 50. Except for low and high values of time index n, the sum of all time-shifted windows tends to 1.

3.5. Pitch-Synchronous Representation

So far, we have only considered the case where all frames have fixed length M. This is the most common way of representing signals in the time-frequency domain.

In pitch-synchronous representation (PSR), the frames have different lengths. *The length of each frame is an integer multiple of the fundamental period (in samples) of the short-time speech signal in that frame*. Thus, the window size can be represented as a function of the fundamental frequency f_0 of the short-time speech signal in the l^{th} frame:

$$M(l) = K \frac{f_s}{f_0(l)} \tag{3.8}$$

where K is a positive integer, f_s is the sampling rate of the audio signal in Hz and $f_0(l)$ is also in Hz.

PSR is a general term for time-frequency representation of signals using window lengths which are multiples of the corresponding fundamental periods. If the Fourier basis is used, the representation is the STFT, which can be expressed as:

$$X_l(k) = \sum_{n=0}^{M(l)-1} x(n + lH) w_l(n) e^{-j2\pi kn / M(l)} \tag{3.9}$$

where $w_l(n)$ is the window function of length $M(l)$.

A window function belonging to the Blackman-Harris family (eg., Hamming and Hann) has the following property: if the window function is modulated by a sinusoid and the window size M is an integer multiple of the fundamental period of this sinusoid, the side lobes almost disappear. *Thus, the energy in the side lobes is almost zero when PSR is used.* This is a key advantage of PSR. This is illustrated in figure 3.6.

PSR has an additional advantage: there is a fixed mapping from DFT bin indices to the harmonic frequencies. Thus, assuming the noise level is sufficiently low, *the peaks of the DFT exactly correspond to peaks of the DTFT*. In other words, the peaks of the DFT are samples of the DTFT at its maxima. As a result, the SNR is maximum at the peaks of the DFT.

3.5.1. Reconstruction of Original Signal

Because of varying window sizes, the OLA method described in section 3.3 cannot be directly applied for the pitch-synchronous case. The procedure is slightly different:

1. The inverse DFT of $X_l(k)$ is computed for $l \in \{0,1,...,L-1\}$. The short-time signals obtained are windowed and then appropriately time-shifted and overlap-added to give $y_u(n)$:

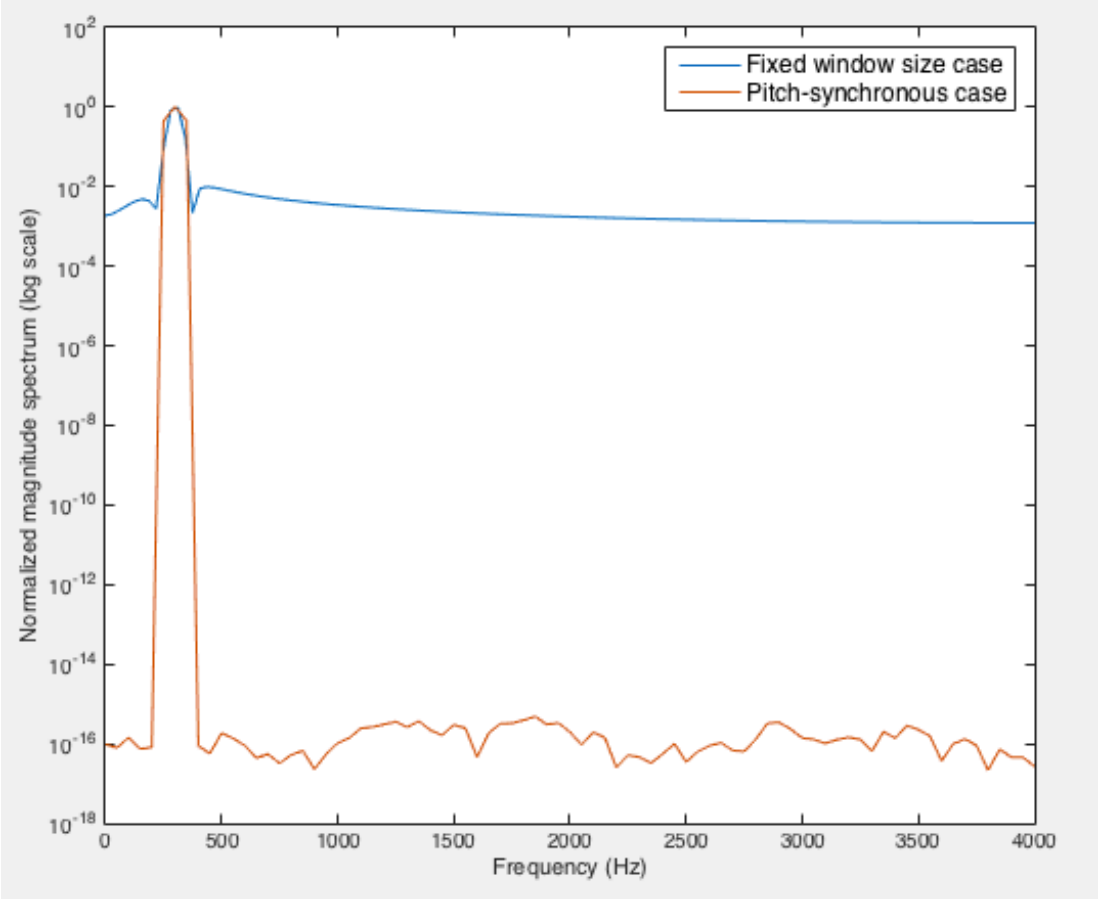


Fig. 3.6. Comparison of side lobe levels for two cases. In both cases, a Hamming window is modulated by a sinusoid of frequency 300 Hz. In the first case (blue line), the window size is fixed and equal to 256. In the second case (orange line), the window size is 160, an integer multiple of the period of the sinusoid (as is the case in PSR). Notice how the energy in the side lobes is almost zero in the pitch-synchronous case.

$$y_w(n,l) = \frac{1}{M(l)} w_l(n) \sum_{k=0}^{M(l)-1} X_l(k) e^{j2\pi nk\,/\,M(l)} \tag{3.10}$$

$$y_u(n) = \sum_{l=0}^{L-1} y_w(n - lH, l) \tag{3.11}$$

(The subscript u stands for "unnormalized".)

2. The squares of the window functions are appropriately time-shifted and overlap-added to give $\gamma(n)$:

$$\gamma(n) = \sum_{l=0}^{L-1} w_l^2(n - lH) \tag{3.12}$$

3. The reconstructed signal is obtained by normalizing $y_u(n)$ by dividing it with $\gamma(n)$:

$$y(n) = \frac{y_u(n)}{\gamma(n)} \tag{3.13}$$

CHAPTER 4
SPEECH ENHANCEMENT IN THE PITCH-SYNCHRONOUS STFT DOMAIN

Because of the advantages of PSR discussed in section 3.5, the amplitudes and phases of the sinusoids in the harmonic model can be estimated more accurately. In this chapter, we propose a method of speech enhancement in which the magnitude spectrum is estimated using Wiener filtering and the phase spectrum is estimated in the pitch-synchronous STFT domain.

4.1. Fundamental Frequency Detection

The estimation of the fundamental frequency of a sound is called fundamental frequency (f_0) detection.

For computing the pitch-synchronous STFT of a signal, it is necessary to first estimate the fundamental frequency of the signal at different instants of time. The time instants correspond to the centres of the frames. Thus, the difference (in samples) between consecutive instants of time is equal to the hop size H chosen for the computation of the STFT.

For monophonic sounds free from noise, f_0 detection is a relatively simple task. However, for polyphonic sounds, and for monophonic sounds corrupted by noise, f_0 detection is a complex task. In fact, accurate f_0 detection in the former case remains an open problem.

Some f_0 detection algorithms for monophonic signals are discussed next. For the discussion that follows, the short-time signal is denoted by $x(n)$ and its length is denoted by N. It is assumed to have a constant fundamental frequency. In order to track the changes in the fundamental frequency of a

longer signal with time, the signal is divided into short frames and the fundamental frequency for each frame is computed separately.

4.1.1. Autocorrelation Method

The autocorrelation method uses a time-domain approach. The autocorrelation function is defined as:

$$R_x(l) = \sum_{n=0}^{N-1-l} x(n)x(n+l) \tag{4.1}$$

where l is the lag.

The autocorrelation function shows peaks at multiples of the fundamental period. The value of l corresponding to the highest non-zero-lag peak is the fundamental period. The fundamental frequency is the inverse of the fundamental period.

The upper limit and lower limit of f_0 must be specified in advance for the autocorrelation method. Though it is a very simple method, it has several disadvantages:

- If the lower limit is too close to zero, the algorithm may erroneously choose the zero-lag peak.
- If the upper limit is too large, it may erroneously choose a higher-order peak.

4.1.2. YIN Algorithm

The YIN algorithm is an improvement over the autocorrelation method. Like the autocorrelation method, it uses a time-domain approach. It uses a difference function defined as:

$$d(l) = \sum_{n=0}^{N-1-l} \big(x(n) - x(n+l)\big)^2 \tag{4.2}$$

where, again, l is the lag.

The algorithm searches for the values of l for which the difference function is minimum. The values of l found are multiples of the period. The fundamental period can be determined this way.

The advantages of this method are:
- It is not prone to the “too low” error in the autocorrelation method.
- There is no upper limit on the frequency search range.
- It is relatively simple. It may be implemented efficiently and with low latency.

A disadvantage of this method is that it does not perform very well under high levels of noise.

4.1.3. Two-Way Mismatch Algorithm

The two-way mismatch (TWM) algorithm uses a frequency-domain approach. In this method, the estimated f_0 is chosen to minimize discrepancies between measured partial frequencies[3] and harmonic frequencies generated by trial values of f_0. For each trial f_0, mismatches between the harmonics generated and the measured partial frequencies are averaged over a fixed subset of the available partials. A weighting scheme is used to reduce the susceptibility of the procedure to the presence of noise or absence of certain partials in the spectral data.

The advantages of this method are:
- It can give reasonably good results with real signals corrupted by noise and reverberation.
- It can be modified for f_0 estimation of two simultaneous monophonic signals.

This method has some disadvantages:
- The upper and lower limits on the fundamental frequency have to be specified.
- It is more computationally expensive compared to YIN algorithm.

[3] A partial refers to any of the sinusoids of which a complex tone is composed. Its frequency is not necessarily an integer multiple of the fundamental frequency.

4.1.4. PEFAC

PEFAC uses a frequency domain approach.

The algorithm uses non-linear amplitude compression to attenuate narrow-band noise components and a comb-filter, whose impulse response is chosen to attenuate smoothly varying noise components, applied in the log-frequency domain.

The main advantage of this method is that it is robust to high levels of noise, which makes it suitable for use in speech enhancement.

4.2. Phase Reconstruction

Most speech enhancement methods, such as Wiener filtering, involve changing only the magnitude spectrum of the signal. However, work by Paliwal *et al.* shows that changes in the phase spectrum can have an effect on the intelligibility of speech as well.

The speech enhancement algorithm for reconstruction of STFT phase of voiced speech proposed by Krawczyk and Gerkmann improves speech intelligibility even when the magnitude spectrum is left unchanged. In their method, the STFT is computed using a window of constant size.

In this section, we propose an algorithm, based on Krawczyk and Gerkmann's method, for reconstruction of pitch-synchronous STFT phase. As will be seen shortly, the use of pitch-synchronous STFT offers a few advantages.

4.2.1. Proposed Method

The harmonic model (equation 2.9) is used to represent the underlying speech signal. Thus, the overall noisy signal is given by:

$$x(n) = \sum_{r=1}^{R(n)} A_r(n)\, cos\left(2\pi r f_0(n) n + \varphi_r(n)\right) + v(n) \quad (4.3)$$

where $v(n)$ is the noise contaminating the signal.

The STFT of $x(n)$ can be written as:

$$X_l(k) = S_l(k) + V_l(k) \tag{4.4}$$

where $S_l(k)$ is the STFT of the underlying speech signal and $V_l(k)$ is the STFT of the noise component.

We assume that $A_r(n), f_0(n)$ and $\varphi_r(n)$ vary relatively slowly over time so that they're approximately constant over one frame. Thus, for the l^{th} frame, we denote the amplitude, frequency and phase of the r^{th} harmonic as $A_{r,l}$, $f_{r,l}$ ($= rf_{0,l}$) and $\varphi_{r,l}$ respectively, and the number of harmonics as R_l.

The l^{th} frame obtained upon windowing is given by:

$$s(n + lH)w_l(n) = \left[\sum_{r=1}^{R_l} A_{r,l}\, cos\left(2\pi f_{r,l}(n + lH) + \varphi_{r,l}\right) \right] w_l(n)$$

or,

$$s(n + lH)w_l(n) = \left[\sum_{r=1}^{R_l} A_{r,l}\, cos\left(2\pi f_{r,l}n + 2\pi f_{r,l}lH + \varphi_{r,l}\right) \right] w_l(n)$$

(4.5)

Let

$$\varphi'_{r,l} = 2\pi f_{r,l}lH + \varphi_{r,l} \tag{4.6}$$

Substituting equation (4.6) in equation (4.5), we obtain:

$$s(n + lH)w_l(n) = \sum_{r=1}^{R_l} w_l(n)\, A_{r,l}\, cos\left(2\pi f_{r,l}n + \varphi'_{r,l}\right)$$

(4.7)

The STFT is obtained by taking the DFT of equation (4.7):

$$S_l(k) = \sum_{r=1}^{R_l} \frac{A_{r,l}}{2} \left[e^{j\varphi'_{r,l}} W_l((k - \tilde{k}_{r,l}))_{M(l)} + e^{-j\varphi'_{r,l}} W_l((k + \tilde{k}_{r,l}))_{M(l)} \right] \tag{4.8}$$

Here, $W_l(k)$ is the DFT of $w_l(n)$, $((\cdot))_N$ denotes the modulo N operator, and

$$\tilde{k}_{r,l} = M(l)\, f_{r,l} = K\, T_{0,l}\, r\, f_{0,l}$$

or,

$$\boxed{\tilde{k}_{r,l} = rK} \tag{4.9}$$

where $T_{0,l}\ (= 1 / f_{0,l})$ is the fundamental period of the l^{th} frame and K is the ratio of the window size and the fundamental period.

Since we're dealing with signals that have real values in the time domain, the STFT is conjugate symmetric:

$$S_l((k))_{M(l)} = S_l((-k))_{M(l)} \tag{4.10}$$

So, we need to do the processing on $S_l(k)$ only for $k \in \{0,1,...M_h(l) - 1\}$, where $M_h(l)$ is defined as:

$$\begin{aligned} M_h(l) &= \frac{M(l)}{2} + 1 && \text{if } M(l) \text{ is even} \\ &= \frac{M(l) + 1}{2} && \text{if } M(l) \text{ is odd} \end{aligned} \tag{4.11}$$

Since most of the energy of $W(k)$ is centred around $k = 0$, $S_l(k)$ can be approximated in the vicinity of $k = \tilde{k}_{r,l}$ as:

$$\begin{aligned} S_l(k) &\approx \frac{A_{r,l}}{2}\, e^{j\varphi'_{r,l}}\, W_l((k - \tilde{k}_{r,l}))_{M(l)} \\ &= \frac{A_{r,l}}{2} \left| W_l((k - \tilde{k}_{r,l}))_{M(l)} \right| e^{j\phi_{r,l}(k)} \end{aligned} \tag{4.12}$$

where

$$\phi_{r,l}(k) = 2\pi f_{r,l} l H + \varphi_{r,l} + \phi_l^W(k - \tilde{k}_{r,l}) \tag{4.13}$$

Here, $\phi_l^W(k - \tilde{k}_{r,l})$ is the phase of $W_l((k - \tilde{k}_{r,l}))_{M(l)}$. If $M(l)$ is odd, $W_l(k)$ has a linear phase. Because of the high side lobe attenuation in PSR, the approximation in equation (4.12) is more accurate compared to the case where a constant window size is used. This is one of the advantages using PSR for phase reconstruction.

Equation (4.12) assumes that the frequency resolution is high enough so that after windowing, the main lobes of neighboring harmonics are separated. The window size should not be too low—otherwise, the main lobes of neighboring harmonics would interfere with each other.

From equation (4.12), we see that:

$$\angle S_l(\tilde{k}_{r,l}) \approx \text{princ}\{\phi_{r,l}\} \tag{4.14}$$

where $\text{princ}\{\cdot\}$ denotes the principal value operator, which maps the angle onto the range $[-\pi, \pi)$.

From one voiced frame to the next, $f_{r,l}$ and $\varphi_{r,l}$ do not change much. Also, $\phi_l^W(k - \tilde{k}_{r,l}) = 0$ at $k = \tilde{k}_{r,l}$. Thus, using equation (4.13), we get

$$\phi_{r,\,l}(\tilde{k}_{r,l}) \approx \phi_{r,\,l-1}(\tilde{k}_{r,l}) + 2\pi f_{r,\,l} H \tag{4.15}$$

Thus, for bins that contain harmonics $(\tilde{k}_{r,l})$, we can reconstruct the phase along time using the following algorithm:

1. At the onset of voiced speech, the estimated phase $\angle\hat{S}_l(\tilde{k}_{r,l})$ is initialized to the noisy phase $\angle X_l(\tilde{k}_{r,l})$.
2. If a frame is voiced but not the onset of a voiced region, the estimated phase for that frame is given by:

$$\boxed{\angle\hat{S}_l(\tilde{k}_{r,l}) = \text{princ}\{\angle\hat{S}_{l-1}(\tilde{k}_{r,l-1}) + 2\pi f_{r,\,l} H\}} \tag{4.16}$$

This is another way in which using PSR is beneficial: the SNR of the noisy speech signal is maximum at the peaks of the speech signal, which are at the harmonic frequencies of the speech signal. When PSR is used, the

harmonic frequencies are directly mapped to DFT bins. Thus, the SNR is maximum at $k = \tilde{k}_{r,l} \forall r \in \{1,2,...,R_l\}$, and the estimates $\angle\hat{S}_l(\tilde{k}_{r,l})$ obtained using the algorithm described above are more accurate.

To reconstruct the phase along frequency, we go back to equation (4.12). We see that in the vicinity of $k = \tilde{k}_{r,l}$, the estimated phase can be obtained as:

$$\boxed{\angle\hat{S}_l(\tilde{k}_{r,l} + i) = \text{princ}\{\angle\hat{S}_l(\tilde{k}_{r,l}) + \phi_l^W(i)\}}$$

$$\forall i \in \{-\Delta k, \ldots, \Delta k\} \qquad (4.17)$$

where

$$\Delta k = \left\lceil \frac{K}{2} \right\rceil \qquad (4.18)$$

Equation (4.17) shows another advantage of using PSR. If a constant window size is used, we have to compute the angle of the DTFT of the window function for phase reconstruction along frequency. However, when PSR is used, we simply need to compute the angle of the DFT of the window function, which is much more efficient.

Since this method uses the harmonic model of speech, it does not take into account the effect of inharmonicity. Thus, this method works best when lower sampling rates are used, since we have to take fewer harmonics into account when the sampling rate is low (see equation 2.10).

4.3. Wiener Filter

For discrete-time signals, the Wiener filter is a linear shift-invariant (LSI) filter whose output is an estimate of the desired signal when its input is the noisy signal $x(n)$. It minimizes the mean squared error (MSE) between the estimated signal $\hat{s}(n)$ and the desired signal $s(n)$.

The signal model is described by equation (1.1), which is rewritten here for convenience:

$$x(n) = s(n) + v(n) \qquad (4.19)$$

If the impulse response of the Wiener filter is $h(n)$, then:

$$\hat{s}(n) = \sum_{m \in B} h(m)x(n-m) \tag{4.20}$$

- If the filter is casual and IIR, $B = \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of all nonnegative integers.
- If the filter is casual and FIR, $B = \{0,1,2,...,P\}$, where P is the filter order.
- If the filter is noncausal, $B = \mathbb{Z}$, where $\mathbb{Z}$ is the set of all integers.

The MSE is:

$$\varepsilon = E\left[\left(s(n) - \hat{s}(n)\right)^2\right] \tag{4.21}$$

In order to minimize the MSE, we differentiate it with respect to $h(i)\ \forall i \in B$. The result is the following set of equations:

$$\sum_{m \in B} h(m)R_{xx}(i-m) = R_{sx}(i) \tag{4.22}$$

where $R_{sx}(i)$ is the cross-correlation between $s(n)$ and $x(n)$ at lag i. These equations are called the **Wiener-Hopf equations**.

If the filter is not restricted to be causal, we can obtain the transfer function $H(e^{j\omega})$ by taking the DTFT of equation (4.22) and rearranging the terms:

$$H(e^{j\omega}) = \frac{S_{sx}(e^{j\omega})}{S_{xx}(e^{j\omega})} \tag{4.23}$$

where $S_{sx}(e^{j\omega})$ is the cross spectral density of $s(n)$ and $x(n)$ and $S_{xx}(e^{j\omega})$ is the power spectral density (PSD) of $x(n)$.

If we assume that the underlying speech signal $s(n)$ and the noise $v(n)$ are uncorrelated, then using equation (4.19), we get:

$$R_{xx}(i) = R_{ss}(i) + R_{vv}(i) \tag{4.24}$$

$$R_{sx}(i) = R_{ss}(i) \tag{4.25}$$

Taking the DTFT of equations (4.24) and (4.25) and substituting in equation (4.23), we get:

$$\boxed{H(e^{j\omega}) = \frac{S_{ss}(e^{j\omega})}{S_{ss}(e^{j\omega}) + S_{vv}(e^{j\omega})}} \tag{4.26}$$

Since $S_{ss}(e^{j\omega})$ and $S_{vv}(e^{j\omega})$ are PSDs, they are real and positive. Thus, $H(e^{j\omega})$ is also real and positive. As a result, *the Wiener filter described by equation (4.26) only changes the magnitude spectrum of the signal*.

Equation (4.26) can be rewritten as:

$$H(e^{j\omega}) = \frac{SNR(e^{j\omega})}{SNR(e^{j\omega}) + 1} \tag{4.27}$$

where $SNR(e^{j\omega}) = S_{ss}(e^{j\omega}) / S_{vv}(e^{j\omega})$ is the signal-to-noise ratio at angular frequency ω. Thus, for Wiener filtering, accurate estimation of the SNR is necessary.

4.4. The Proposed Algorithm

The following is a block diagram of the proposed algorithm:

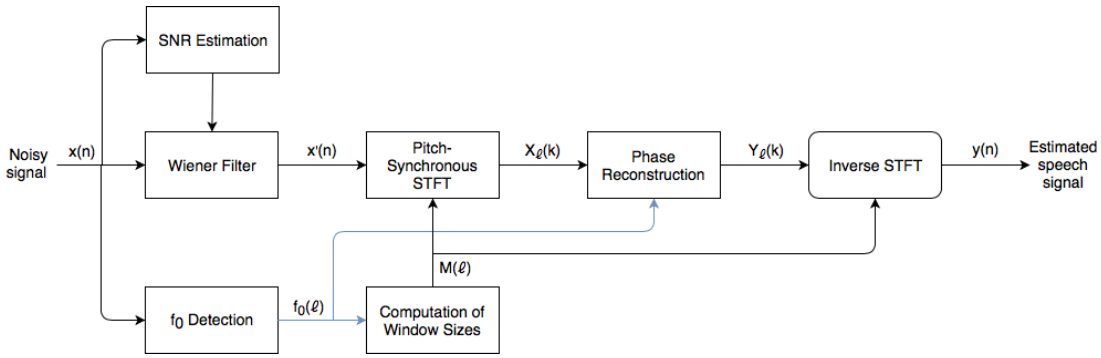


Fig. 4.1. Block diagram of the algorithm.

The steps of the algorithm are explained below:

1. The SNR of the noisy input signal $x(n)$ is estimated. This information is used by the Wiener filter.
2. $x(n)$ is then filtered by the Wiener filter. This changes the magnitude spectrum but does not affect the phase spectrum. A constant window size is used for Wiener filtering. The speech in the resulting signal $x'(n)$ is more intelligible, but further improvement is possible through reconstruction of the phase spectrum.
3. The fundamental frequency estimator (PEFAC) computes the fundamental frequencies $f_0(l)$ of different frames of $x(n)$. Since PEFAC is robust to high levels of noise, the f_0 estimation is done directly on the noisy signal $x(n)$.
4. The window sizes $M(l)$ are computed using equation (3.8) and rounded off to nearest integers.
5. The pitch-synchronous STFT, $X_l(k)$, of $x'(n)$ is computed.
6. The phase spectrum is reconstructed using the algorithm proposed in section 4.2.1 to give $\angle\hat{S}_l(k)$, while the magnitude spectrum is left unchanged. The result is $Y_l(k) = \left|X_l(k)\right| \exp(j\angle\hat{S}_l(k))$.
7. The inverse STFT of $Y_l(k)$ is computed as explained in section 3.5.1.

CHAPTER 5
EXPERIMENTAL RESULTS

We implemented the speech enhancement algorithm proposed in MATLAB to evaluate its performance and compare it with other algorithms. For SNR estimation, the harmonic regeneration noise reduction (HRNR) technique proposed by Plapous, Marro and Scalart is used. For fundamental frequency detection, PEFAC is used.

5.1. Parameters

For the algorithm, the sampling rate f_s is chosen to be 16 kHz, since it is sufficient to represent most of the information present in speech. If the input audio signal has a higher sampling rate, it is first downsampled to 16 kHz.

The window function used for PSR is the Hamming window, since it offers a good tradeoff between frequency resolution and side lobe attenuation. Three other window functions were considered: Hann window, square-root Hamming window and square-root Hann window. However, for phase reconstruction, the Hamming window was found to give the best results.

The default frame length is chosen to be 32 ms. Since f_s is 16 kHz, the default window size, i.e., the window size of unvoiced frames, is therefore $32 * 16 = 512.$

When the Hamming window is used, the minimum overlap between adjacent frames must be around 73%, or equivalently, the hop size H must be around 27% of the window size. Since PSR is used, the hop size is chosen to be 12.5% of the default window size in order to ensure that the overlap between frames of sizes lower than the default size is also sufficient. Thus, the hop size H is $12.5 * 512 = 64.$

For PSR, K (the ratio of the window size and the fundamental period) is chosen to be 6 so that the window sizes of voiced frames stay in the appropriate range (neither too high nor too low).

5.2. Observations

We tested the algorithm on different types of speech signals. For the purpose of simulation, additive white Gaussian noise is added to the speech signals. These are the observations:

- Phase reconstruction is faster with PSR method as compared to the constant window size method. When a constant window size is used, the phase of the DTFT of the window function has to be computed at different values of ω for each frame. On the other hand, when PSR is used, the phase of the DFT of the window function has to be computed (which is much faster than computation of the phase of the DTFT), and only once for each frame. The speeds are compared in Table 5.1.
- When it comes to intelligibility, whether the result is better for the PSR case or the constant window size case is subjective. The output for the PSR case tends to sound less artificial compared to the constant window size case. For low SNRs, the difference between the noisy signal and the reconstructed signal is more pronounced when a constant window size is used.

Table 5.1. Comparison of speeds of the phase reconstruction algorithm for the PSR case and the constant window size (M) case. This includes the time taken for STFT and inverse STFT computations.

File name	Clip duration	Time taken for phase reconstruction (PSR case)	Time taken for phase reconstruction (constant M case)
speech female 1.wav	11 s	3.46 s	6.87 s
speech female 2.wav	70 s	21.86 s	48.23 s
speech female 3.wav	82 s	22.89 s	50.1 s
speech male 1.wav	4 s	1.74 s	3.61 s
speech male 2.wav	8 s	3.23 s	6.67 s
speech child.wav	24 s	6.69 s	13.91 s

- The Wiener filter does not give a good result when the SNR is very low. Thus, when the SNR is very low, only phase reconstruction should be used.

(For comparison between phase reconstruction for the PSR case and the constant window size case, a sampling rate of 8 kHz is used, since the original algorithm is recommended only for sampling rates of up to 8 kHz.)

For cases where the SNR is not too low, the addition of Wiener filter improves the result. The spectrograms of noisy and enhanced signals for two different speech sounds are shown on the next page. (Since phase reconstruction does not affect the magnitude spectrum, the magnitude spectrum of the overall enhanced signal is the same as the magnitude spectrum of the signal after Wiener filtering.)

5.3. Conclusion

When the harmonic model of speech is used, PSR offers several advantages due to its properties discussed in section 3.5. The high side lobe attenuation in PSR is advantageous. In addition to that, the fact that harmonic frequencies directly map to DFT bins is also advantageous, especially for application in parameter estimation.

The algorithm for phase reconstruction using PSR presented in this report presents better speed as compared to the methods that use constant window size. In terms of intelligibility, the difference is not very noticeable.

Wiener filtering is used for changing the magnitude spectrum of the speech signal. For relatively high SNRs, the addition of Wiener filtering gives better results compared to when phase reconstruction alone is used. However, for very low SNRs, the addition of Wiener filtering in fact makes the result worse. Thus, for very low SNRs, phase reconstruction alone should be used, or another method of enhancing the magnitude spectrum should be used.

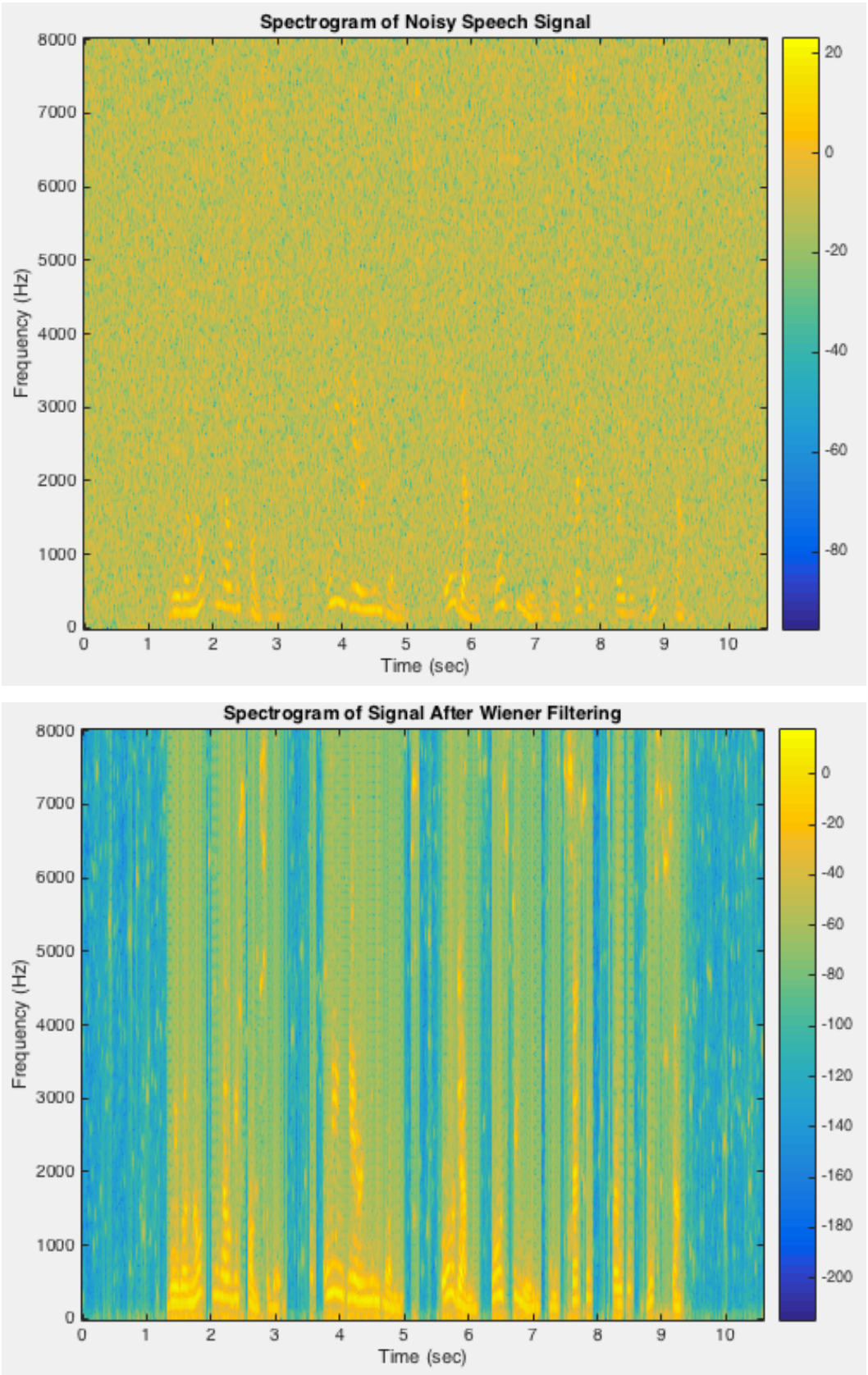


Fig. 5.1. Spectrograms of noisy signal and enhanced signal for the file ‘speech female 1.wav’. The SNR is 30 dB.

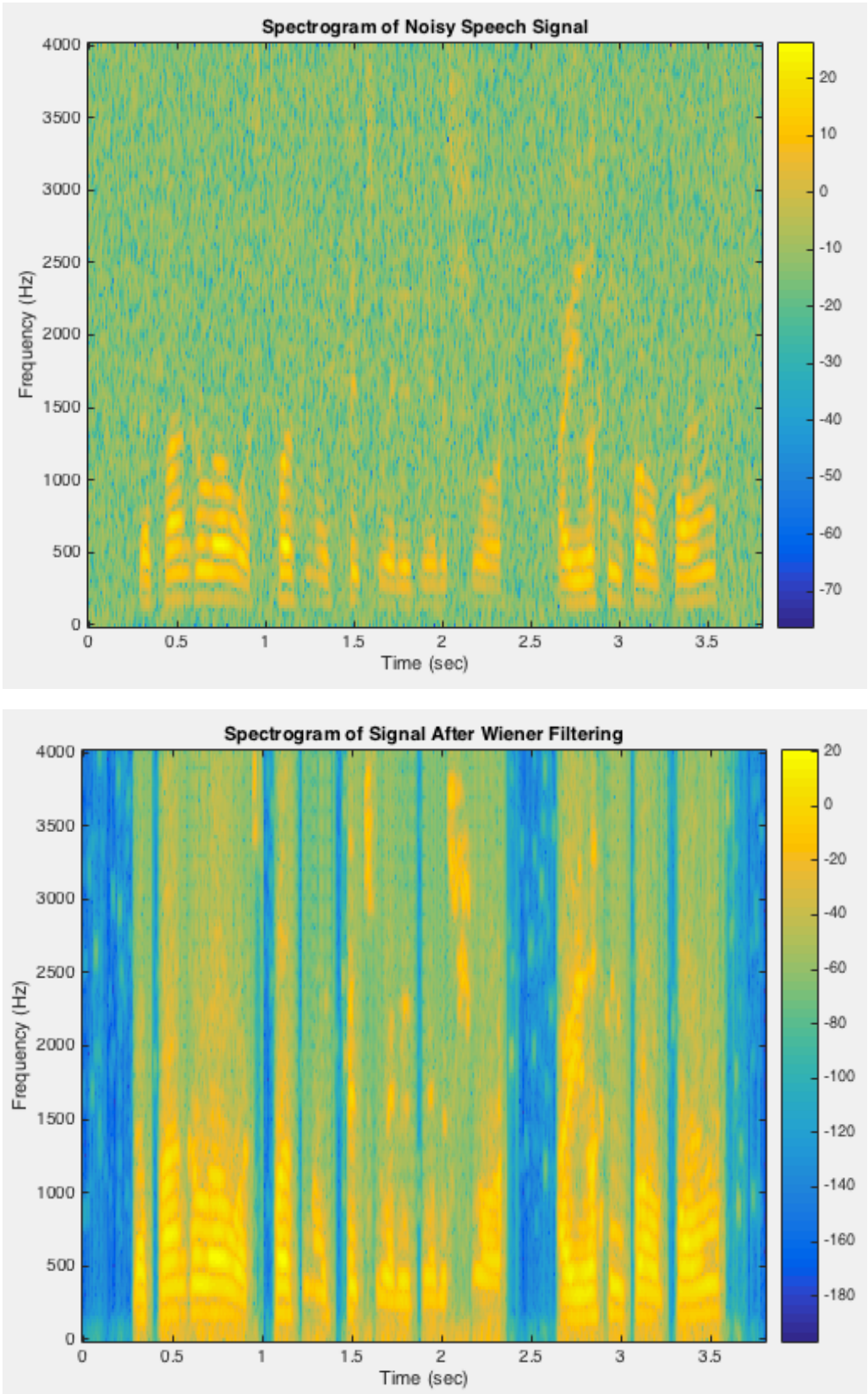


Fig. 5.2. Spectrograms of noisy signal and enhanced signal for the file 'speech male 1.wav'. The SNR is 30 dB.

REFERENCES

[1] Martin Krawczyk and Timo Gerkmann, “STFT Phase Reconstruction in Voiced Speech for an Improved Single-Channel Speech Enhancement,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, Dec. 2014.

[2] Johannes Stahl and Pejman Mowlaee, “A Pitch-Synchronous Simultaneous Detection-Estimation Framework for Speech Enhancement,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 26, no. 2, Feb. 2018.

[3] S. Gonzalez and M. Brookes, “PEFAC—A pitch estimation algorithm robust to high levels of noise,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 2, pp. 518–530, Feb. 2014.

[4] Alain de Cheveigné and Hideki Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *Journal of the Acoustical Society of America*, vol. 111, no. 4, Apr. 2002.

[5] Robert C. Maher and James W. Beauchamp, “Fundamental frequency estimation of musical signals using a two-way mismatch procedure,” Journal of the Acoustical Society of America, vol. 95, no. 4, Apr. 1994.

[6] Cyril Plapous, Claude Marro and Pascal Scalart. “Improved Signal-to-Noise Ratio Estimation for Speech Enhancement,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, Nov. 2006.

[7] Julius O. Smith III, *Spectral Audio Signal Processing*, W3K Publishing.

[8] Alan V. Oppenheim, Ronald W. Schafer and John R. Buck, *Discrete-Time Signal Processing*, 3rd ed., Pearson.

[9] Xavier Serra and Julius O Smith, III, “Audio Signal Processing for Music Applications,” [Online]. Available: https://www.coursera.org/learn/audio-signal-processing

[10] A. H. Sayed, “Voiced and Unvoiced Speech Overview,”, [Online]. Available: http://www.seas.ucla.edu/dsplab/vus/over.html

[11] “What is Self-Noise (or Equivalent Noise Level)?,” [Online]. Available: http://www.neumann.com/homestudio/en/what-is-self-noise-or-equivalent-noise-level

[12] Alan V. Oppenheim and George C. Verghese, “Signals, Systems, and Inference—Class Notes for 6.011: Introduction to Communication, Control and Signal Processing Spring 2010,” [Online]. Available: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-011-introduction-to-communication-control-and-signal-processing-spring-2010/readings/

[13] Dan Ellis, “Lecture 5: Speech modeling,” [Online]. Available: https://engineering.purdue.edu/~ee538/SpeechLPC.pdf
[14] M. Brookes, “VOICEBOX: A speech processing toolbox for MATLAB,” [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
[15] Pascal Scalart, “Wiener filter for Noise Reduction and speech enhancement,” [Online]. Available: https://in.mathworks.com/matlabcentral/fileexchange/24462-wiener-filter-for-noise-reduction-and-speech-enhancement
[16] KK Paliwal and LD Alsteris, “On the usefulness of STFT phase spectrum in human listening tests,” *Speech Communication*, vol. 45, no. 2, Feb. 2005.