Lecture Notes in Speech Production, Speech Coding, and Speech Recognition

Mark Hasegawa-Johnson
University of Illinois at Urbana-Champaign

February 17, 2000

Chapter 3

Short-Time Signal Processing

3.1 Short-Time Analysis

3.1.1 From 1D signal to Multidimensional "Frames"

Speech is nonstationary, but all of our analysis tools (e.g. transforms) assume a stationary signal. So the first step in speech analysis is to chop it up into "frames" which are short enough (∼20ms) that we can consider them stationary. Define

$$x_n(m) = x(n-m)w(m) \tag{3.1}$$

Where $w(m)$ is a finite-length windowing function of length L; for example, the rectangular window $w(m) = u(m) - u(m-L)$. We can slide the window by M points between frames, and output a matrix where each frame is a row in the matrix:

$$\begin{bmatrix} x(0) \\ x(1) \\ \dots \\ x(n) \\ \dots \end{bmatrix} \rightarrow \boxed{\text{Convert to Frames}} \rightarrow \begin{bmatrix} x_0(0) & \dots & x_0(m) & \dots & x_0(L-1) \\ x_M(0) & \dots & x_M(m) & \dots & x_M(L-1) \\ & & \dots & & \\ x_n(0) & \dots & x_n(m) & \dots & x_n(L-1) \\ & & \dots & & \end{bmatrix} \tag{3.2}$$

- **Frame rate:** there is one frame every M samples, or F_s/M frames/sec.
- **Overlap:** neighboring frames overlap by $L - M$ samples.

3.1.2 Time-Domain Analysis

Short-Time Energy

$$E_n = \sum_{m=0}^{L-1} x_n(m)^2 \tag{3.3}$$

Useful for finding silent portions.

Zero-Crossing Rate

$$\text{ZCR}_n = \sum_{m=1}^{L-1} \left| \frac{\text{sign}(x_n(m)) - \text{sign}(x_n(m-1))}{2} \right| \tag{3.4}$$

Tends to be higher for noise excitation, lower for unvoiced excitation.

Autocorrelation

$$R_n(k) = \sum_m x_n(m)x_n(m-k) \tag{3.5}$$

$R_n(k)$ is a measure of how similar $x(n)$ is to $x(n-k)$, so it is useful for pitch detection.

- Even function, $R_n(k) = R_n(-k)$.
- $R_n(k)$ is the projection of $x_n(m)$ onto $x_n(m-k)$, so it is maximum at $k=0$:

$$\sum_{m=k}^{L-1} x_n(m)x_n(m-k) \leq \sqrt{\sum_{m=k}^{L-1} x_n(m)^2}\sqrt{\sum_{m=k}^{L-1} x_n(m-k)^2} \leq \sum_{m=0}^{L-1} x_n(m)^2 \tag{3.6}$$

- If $x(n)$ is periodic, $R_n(k)$ approximately periodic with same period:

$$\begin{aligned} x_n(m) &= x_n(m+N) && (3.7)\\ x_n(m-N) &= x_n(m) && (3.8)\\ R_n(k+N) &= \sum_m x_n(m)x_n(m-k-N) && (3.9)\\ &\approx \sum_m x_n(m)x_n(m-k) && (3.10)\\ &= R_n(k) && (3.11) \end{aligned}$$

3.2 Short-Time Fourier Transform

The short-time Fourier transform is a function of both n and ω:

$$S_n(e^{j\omega}) = \sum_{m=0}^{L-1} s(m)w(n-m)e^{-j\omega m} \tag{3.12}$$

Two interpretations:

1. n fixed — Fourier transform:
$$X_n(e^{j\omega}) = \mathcal{F}\{x(m)w(n-m)\} \tag{3.13}$$
2. ω fixed — Frequency-shifted bandpass filter:
$$X_n(e^{j\omega}) = x(n)e^{-j\omega n} * w(n) \tag{3.14}$$

3.2.1 Fourier Transform Interpretation

Fourier transform interpretation — STFT is like a DFT convolved with the window spectrum:

$$\begin{aligned} S_n(e^{j\omega}) &= \mathcal{F}\{s(n)w(n-m)\} && (3.15)\\ &= \mathcal{F}\{s(n)\} * \mathcal{F}\{w(n-m)\} && (3.16)\\ &= S(e^{j\omega}) * W(e^{-j\omega})e^{-j\omega n} && (3.17)\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} S(e^{j(\omega-\theta)})W(e^{-j\theta})e^{-j\theta n}d\theta && (3.18) \end{aligned}$$

3.2.2 Filterbank Interpretation

$$S_n(e^{j\omega_i}) = e^{-j\omega_i n} \sum_{m=0}^{L-1} s(m)w(n-m)e^{j\omega_i(n-m)} \tag{3.19}$$

$$= e^{-j\omega_i n} \left[s(n) * w(n)e^{j\omega_i n} \right] \tag{3.20}$$

$$= e^{-j\omega_i n} s_i(n) \tag{3.21}$$

where we have defined a bank of filters $h_i(n)$ as follows:

$$s_i(n) = s(n) * h_i(n), \qquad h_i(n) = w(n)e^{j\omega_i n} \tag{3.22}$$

$w(n)$ can be ANY low-pass FIR filter. It can be a rectangular window, or a Hamming window, or it can be an FIR approximation to an ideal LPF, constructed (for example) using **remez** in matlab. We will assume that $w(n)$ has the following characteristics:

- The window length is L: $w(n)$ is non-zero for $0 \leq n \leq L-1$.
- The filter bandwidth is $2\omega_c$: $W(e^{j\omega})$ is non-zero for $|\omega| < \omega_c$, and approximately zero for all other ω.
- Important questions: How wide is ω_c? How high are the side-lobes?

3.2.3 Sampling in time

Suppose we downsample by a factor of M:

$$y_i(n') = S_{Mn'}(e^{j\omega}) = e^{-j\omega_i M n'} s_i(Mn') \tag{3.23}$$

If we choose $\omega_i = 2\pi r/M$ for some integer r, then

$$y_i(n') = s_i(Mn') = S_{Mn'}(e^{j\omega}) \tag{3.24}$$

This is because the spectrum becomes periodic with period $2\pi/M$. For the same reason, in order to avoid frequency-domain aliasing, M must be limited to

$$M < \frac{2\pi}{2\omega_c} \tag{3.25}$$

3.2.4 Sampling in frequency

Suppose we are interested in filters uniformly spaced in frequency,

$$\omega_i = \frac{2\pi i}{N} \tag{3.26}$$

If $N < L$, we will have aliasing in time, as follows:

$$S_n(e^{j\omega_i}) = \sum_{m=0}^{L-1} s(m)w(n-m)e^{j\frac{2\pi im}{N}} \tag{3.27}$$

$$= \sum_{k=0}^{N-1} \sum_r \left[s(Nr+k)w(n-Nr-k)e^{j2\pi ir} \right] e^{j\frac{2\pi ik}{N}}, \quad m = Nr+k \tag{3.28}$$

$$= \sum_{k=0}^{N-1} u_n(k)e^{j\frac{2\pi ik}{N}} \tag{3.29}$$

where

$$u_n(k) = \sum_r s(Nr+k)w(n-Nr-k) \tag{3.30}$$

If the goal is to eventually reconstruct $s(n)$, as in speech coding, time aliasing is bad. If we don't care about signal reconstruction, as in speech recognition, then time aliasing can be good – it simplifies computation! Thus:

- For perfect reconstruction:
$$N \geq L \tag{3.31}$$
- For spectral analysis, if perfect reconstruction isn't necessary, N is determined by the filter cutoff ω_c:
$$N \geq \frac{2\pi}{2\omega_c} \tag{3.32}$$

In particular, if $N = \frac{2\pi}{2\omega_c}$, the filters are non-overlapping, while if $N > \frac{2\pi}{2\omega_c}$, the filters are overlapping.

3.2.5 Filterbank Reconstruction

Suppose we want to reconstruct the signal $s(n)$ from the downsampled signals

$$y_i(n') = s_i(Mn') = S_{Mn'}(e^{j\frac{2\pi i}{N}}) \tag{3.33}$$

If there is no frequency aliasing, then we can interpolate and band-pass filter $y_i(n)$ to get

$$\sum_{m'=0}^{(L-1)/M} y_i(m')h_i(n - Mm') = s_i(n) * h_i(n) \tag{3.34}$$

But remember that $s_i(n) = s(n) * h_i(n)$, so

$$s_i(n) * h_i(n) = s(n) * h_i(n) * h_i(n) \tag{3.35}$$

Solve for $h_i(n) * h_i(n)$:

$$h_i(n) * h_i(n) = \sum_{m=0}^{L-1} w(m)e^{j\frac{2\pi im}{N}} w(n-m)e^{j\frac{2\pi i(n-m)}{N}} = e^{j\frac{2\pi in}{N}}(w(n) * w(n)) \tag{3.36}$$

We can add these up to get $\hat{s}(n)$:

$$\hat{s}(n) = \sum_{i=0}^{N-1} s(n) * h_i(n) * h_i(n) = s(n) * \sum_{i=0}^{N-1} (w(n) * w(n))e^{j\frac{2\pi in}{N}} \tag{3.37}$$

$$= s(n) * \left[(w(n) * w(n)) \sum_{i=0}^{N-1} e^{j\frac{2\pi in}{N}} \right] = s(n) * \left[(w(n) * w(n))N \sum_r \delta(n - rN) \right] \tag{3.38}$$

If $N \geq L$ (no time-domain aliasing), then

$$\hat{s}(n) = Nw^2(0)s(n) \tag{3.39}$$

Review:

1. Interpolate $y_i(n)$.
2. Band-pass filter using $h_i(n) = w(n)e^{j\frac{2\pi in}{N}}$.
3. Add them up to form $\hat{s}(n)$.
4. Divide by $Nw^2(0)$.

3.2.6 Implementing non-uniform filterbanks using the STFT

Suppose that we want filters which are non-uniformly spaced in frequency. It is possible to get non-uniformly spaced filters by just adding neighboring channels from an STFT-based filterbank. For example:

$$x_i(n) = \frac{1}{2}\left(s_i(n) + s_{i+1}(n)\right) \tag{3.40}$$

$$= s(n) * \frac{1}{2}\left(w(n)e^{j\frac{2\pi in}{N}} + w(n)e^{j\frac{2\pi(i+1)n}{N}}\right) \tag{3.41}$$

$$= s(n) * w(n)\cos(\frac{\pi n}{N})e^{j\frac{2\pi n}{N}(i+\frac{1}{2})} \tag{3.42}$$

This results in a new filter with the following characteristics:

- Bandwidth: $2\omega_c + \frac{2\pi}{N}$.
- Center frequency: $\frac{2\pi n}{N}(i+\frac{1}{2})$.
- Filter length: L.

3.3 Window Characteristics

Different analysis windows $w(n)$ can give different analysis results! For example, STFT is like a DFT convolved with the window spectrum $W(e^{j\omega})$. Therefore we want $W(e^{j\omega})$ to look as much as possible like an impulse ($\delta(\omega)$). In particular, we want:

- Main lobe as narrow as possible.
- Side-lobes as small as possible.

It's not possible to get both narrow main lobe and low-amplitude side-lobes, so we must figure out an engineering trade-off based on system requirements.

3.3.1 Rectangular Window

$$w_1(n) = u(n) - u(n-L) \tag{3.43}$$

$$W_1(e^{j\omega}) = \frac{1 - e^{-j\omega L}}{1 - e^{j\omega}} \tag{3.44}$$

$$= e^{j\frac{\omega(L-1)}{2}}\frac{\sin\frac{\omega L}{2}}{\sin\frac{\omega}{2}} \tag{3.45}$$

$$= e^{j\frac{\omega(L-1)}{2}}\tilde{W}(e^{j\omega}), \quad \tilde{W}(e^{j\omega}) \equiv \frac{\sin\frac{\omega L}{2}}{\sin\frac{\omega}{2}} \tag{3.46}$$

Characteristics:

- Main lobe:
$$W_1(e^0) = L \tag{3.47}$$
- First null:
$$W_1(e^{j\frac{2\pi}{L}}) = 0 \tag{3.48}$$
- First sidelobe:
$$\frac{|W_1(e^{j\frac{3\pi}{L}})|}{|W_1(e^0)|} \approx \frac{1}{L}\left|\frac{\sin\frac{3\pi}{2}}{(3\pi/2L)}\right| = \frac{2}{3\pi} \qquad (-13dB) \tag{3.49}$$

3.3.2 Hamming and Hanning Windows

$$w_2(n) = w_1(n)\left(\beta_1 - 2\beta_2\cos(\alpha n)\right), \qquad \alpha \equiv \frac{2\pi}{L-1} \tag{3.50}$$

$$W_2(e^{j\omega}) = \beta_1 W_1(e^{j\omega}) - \beta_2 W_1(e^{j(\omega+\alpha)}) - \beta_2 W_1(e^{j(\omega-\alpha)}) \tag{3.51}$$

Pulling out the exponential terms, and simplifying, we get

$$= e^{j\frac{\omega(L-1)}{2}}\left[\beta_1\frac{\sin\frac{\omega L}{2}}{\sin\frac{\omega}{2}} + \beta_2\frac{\sin\frac{(\omega+\alpha)L}{2}}{\sin\frac{(\omega+\alpha)}{2}} + \beta_2\frac{\sin\frac{(\omega-\alpha)L}{2}}{\sin\frac{(\omega-\alpha)}{2}}\right] \tag{3.52}$$

The first term in the sum is 180° out of phase with the second and third terms for every side-lobe, so it is possible to make the side-lobes of $W_2(e^{j\omega})$ arbitrarily close to zero by choosing the correct β_1 and β_2. Some typical choices:

- Hanning window: $\beta_1 = 0.5$, $\beta_2 = 0.25$.
- Hamming window: $\beta_1 = 0.54$, $\beta_2 = 0.23$. With these constants, the first side-lobe amplitude is less than 1% of the main lobe amplitude – that is, the first side-lobe is down more than 40dB.

Characteristics of the Hamming window:

- Main lobe:
$$W_2(e^0) = 0.54L \tag{3.53}$$
- First null:
$$W_2(e^{j\frac{4\pi}{L}}) = 0 \tag{3.54}$$
- First sidelobe:
$$20\log_{10}\left(\frac{|W_2(e^{j\frac{5\pi}{L}})|}{|W_2(e^0)|}\right) \approx -41dB \tag{3.55}$$

3.3.3 Window Length

Optimum window length depends on what you want to look at!

- If you want to resolve F_0 (maybe for pitch detection), the main lobe width ($2\times\omega_c$) should not exceed F_0. For example, for a Hamming window,
$$\frac{8\pi}{L} \le \frac{2\pi F_0}{F_s} \tag{3.56}$$
- If you want to resolve the formants, main lobe should be wide enough to "smooth together" the pitch harmonics, but not so wide that it blurs the formant peaks.
- If you want to look at rapidly changing events (e.g. stop release), L should be as short as possible (5-10ms max).

3.4 Exercises

1. A speech signal is sampled at a rate of 20,000 samples per second ($F_s = 20$kHz). A 12-ms window is used for short-time spectral analysis. Assume that a radix-2 FFT is used to compute DFTs.

   (a) How many speech samples are used in each segment?

   (b) If the window is a rectangular window, what analysis frame rate (in frames per second) will guarantee that no frequency-aliasing occurs? (You may assume that the side-lobes have zero amplitude.)

   (c) If the window is a Hamming window, what analysis frame rate (in frames per second) will guarantee that no frequency-aliasing occurs? (You may assume that the side-lobes have zero amplitude.)

   (d) What size FFT is required to guarantee that no time-aliasing will occur?

   (e) Suppose that the window $w(n)$ is a low-pass filter, designed using the **remez** function in matlab so that the analog cutoff frequency is $f_c = 312.5$Hz. What size FFT should you use in order to construct a filterbank with non-overlapping filters? Will time-aliasing occur?