1 Lecture Outline

Reading: Chapters 1 and 2. Notes from your previous DSP class.

We will continue our quick review of DSP

- Inverse Short-Time Fourier Transform
- Random Signal Analysis

2 Inverse Short-Time Fourier Transform

Although the DT-STFT

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m} \qquad (1)$$

and the discrete-STFT

$$X(n,k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_k m} \qquad (2)$$

where $\omega_k = 2\pi k/N$ are primarily used as analysis tools in digital speech processing, there are applications in which we need to synthesize the time-domain speech signal from the STFT. For example, there are applications such as in speech enhancement where we analyze a speech signal with the STFT, enhance the signal (remove noise) in the frequency-domain, and must synthesize the resulting time-domain signal from the modified STFT.

The inverse problem is simply stated: from $X(n,\omega)$ (DT-STFT) recover $x[n]$ and from $X(n,k)$ (discrete STFT) recover $x[n]$. Assuming the STFT is computed each each time instant n, we can show the DT-STFT is always invertible, whereas the discrete-STFT requires certain constraints on the number of frequency points, window length and window type for invertibility. We can also show that as long os the frame advance R is less than the window length L so that no samples are "skipped," the DT-STFT can always be inverted. However, inversion of the discrete-STFT requires the previous constraints and also constraints on the frame advance.

Since the STFT is always computed with the DFT we focus on inversion of the discrete-STFT.

2.1 Direct Inversion

The simplest approach is a straight-forward inversion of the STFT which involves for each frame the following steps:

1. compute the inverse DFT to determine the windowed signal
2. divide out the window function to determine each sample

Once the steps are completed, the resulting speech samples are used to reconstruct the signal.

In order to properly invert the discrete-STFT, several constraints must be followed. There are three parameters: the window length L, the number of frequency points in the DFT N, and the decimation factor R. In order to properly reconstruct each frame, we must have $L \leq N$. If $R < L$, the segments overlap and all samples are present in one or more windows; if $R > L$, some samples of the signal are "skipped" in the STFT and therefore cannot be reconstructed. Thus as one possibility, if the three discrete-STFT parameters satisfy $R \leq L \leq N$, then we can (in principle) recover samples of $x[n]$ block-by-block for all n from $X(n, k)$. When using the FFT to compute the DFT, we have $L = N$.

If the windows do not overlap, all recovered samples are required to reconstruct the signal; if the windows overlap, a given sample will be present in two or more frames and only one copy is required to reconstruct the signal.

Figure 1: Direct Inversion of the STFT

2.2 Overlap Add (OLA) Method

Although exact reconstruction is theoretically possible from the discrete-STFT, when modifications (such as those in enhancement or coding) are made to the STFT, direct inversion may lead to errors at the edges of the frame. Modifications to the STFT result in errors in the sample values which are then amplified by dividing by the small window values present in a tapered window. These edge effects than cause the frames to not fit together smoothly. In such applications, it is helpful to make R smaller than L and N so that the blocks overlap.

A different approach to inversion of the STFT that is more robust to modifications in $X(n, k)$ is to shift the windowed segments to their original time locations and simply add the segments together. In the Overlap Add (OLA) method, we take the inverse DFT for each fixed time in the discrete STFT. However, instead of dividing out the analysis window from each of the resulting short-time sections, we perform an overlap and add operation between the short-time sections. This method works provided the analysis window is designed such that the overlap and add operation effectively eliminates the analysis window from the synthesized sequence, i.e. that the overlapped window samples add to a constant value across time. The intuition here is that the redundancy within the overlapping segments and the averaging of the redundant samples remove the effect of windowing.

For a decimated discrete STFT, if the "shifted-by-R" copies of the window add to a constant, perfect reconstruction can be achieved. This is the condition for perfect reconstruction for the overlap-add method.

As a simple example, consider the rectangular window of length L samples. If $R = L$ the window segments simply fit together block-by-block with no overlap. If L for the rectangular window is even and $R = L/2$ the condition is also satisfied.

Two non-rectangular windows with which perfect reconstruction can be achieved are the Bartlett (triangular) window and Hann window. With the window length L odd and $R = (L - 1)/2$, it is easy to see that the shifted-by-R" copies of the window add to a constant. Although not as obvious, the same holds true for the

Figure 2: OLA contraint

Hann window.

Figure 3: OLA contraint Oppenheim Figure 10.17

3 Power Spectral Density

The autocorrelation sequence of a wide-sense stationary (WSS)[1] random process $x[n]$ is defined as

$$r_{xx}[m] \equiv E\left\{x[n+m]x^*[n]\right\} \tag{3}$$

where m is called the "lag" of the autocorrelation sequence. The power spectrum or Power Spectral Density (PSD) is DTFT of the autocorrelation sequence

$$\begin{aligned} S_{xx}(\omega) &\equiv \sum_{m=-\infty}^{\infty} r_{xx}[m]e^{-j\omega m} \qquad (4) \\ &= \lim_{N\to\infty} \frac{1}{2N+1} E\left\{\left|\sum_{n=-N}^{N} x[n]e^{-j\omega n}\right|^2\right\} \end{aligned}$$

The PSD tells us the average power spectrum for an infinitely long signal.

In many applications such as speech processing, the random process is not WSS, the autocorrelation is not known, and/or we have only a finite number of samples. Therefore the PSD must be estimated. In this class, we will utilize two methods for estimating the PSD: periodogram and Welch's method.

3.1 Periodogram

For a random signal of length N, we can estimate the autocorrelation sequence as

$$\hat{r}_{xx}[m] \equiv \frac{1}{N}\sum_{n=0}^{N-1} x[n+m]x^*[n]. \tag{5}$$

[1]WSS requires: 1) constant mean μ, 2) $r_{xx}[m]$ depends only on the lag m and not time n, 3) $\sigma^2 < \infty$.

Note that depending on m, the above equation requires $x[n]$ outside of the range $0 \leq n \leq N-1$. Taking the DTFT of the above estimate, leads to an estimate of the PSD known as the *periodogram*

$$\hat{S}_{xx}(\omega) \equiv \sum_{m=-N+1}^{N-1} \hat{r}_{xx}[m] e^{-j\omega m}. \tag{6}$$

In order to express (6), in terms of a length N random sequence $x[n]$ using only samples within the range $0 \leq n \leq N-1$, we define

$$x_N[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Equation (5) can then be written as

$$\begin{aligned} \hat{r}_{xx}[m] &= \frac{1}{N} \sum_{n=-\infty}^{\infty} x_N[n+m] x^*[n] \qquad (8) \\ &= \frac{1}{N} x_N[m] * x_N^*[-m] \end{aligned}$$

where $*$ denotes convolution and * denotes complex conjugation. Taking the DTFT above the above yields an equivalent (and more practical) version of the periodogram

$$\begin{aligned} \hat{S}(\omega) &= \frac{1}{N} X_N(\omega) X_N^*(\omega) \qquad (9) \\ &= \frac{1}{N} |X_N(\omega)|^2. \end{aligned}$$

3.2 Improvements to the Periodogram

There are several ways to improve the periodogram. First (7) amounts to a length N rectangular window. Other alternate windows such as Hamming, Hanning, Bartlett (triangular), etc. may be more useful. Application of any of these windows modifies (7) to

$$x_N[n] = x[n]w[n] \tag{10}$$

where $w[n]$ is the length N window and $0 \leq n \leq N-1$ and leads to what is known as the "modified periodogram."

The second method of improving the periodogram is known as "periodogram averaging" or "Bartlett's method." The idea here is to divide the length N random process $x_N[n]$ into K segments each of length L ($N = KL$). A periodogram is computed for each length L segment and these are then averaged:

$$\hat{S}(\omega) \equiv \frac{1}{K} \sum_{k=0}^{K-1} \left(\frac{1}{L} |X_{L,k}(\omega)|^2 \right) \tag{11}$$

where $\frac{1}{L}|X_{L,k}(\omega)|^2$ denotes the kth, periodogram of the length L random process.

3.3 Welch's Method

Welch proposed two modifications to Bartlett's method. The first is to allow the segments to overlap and the second is to allow a non-rectangular window to be applied to the segments. Welch's method for PSD estimation is then

$$\hat{S}(\omega) \equiv \frac{1}{KLU} \sum_{k=0}^{K-1} \left| \sum_{n=0}^{L-1} w[n] x[n+kD] e^{-j\omega n} \right|^2 \tag{12}$$

where

$$U = \frac{1}{L} \sum_{n=0}^{L-1} |w[n]|^2 \tag{13}$$

is a window normalization factor, D is the number of samples the window is advanced by, i.e. $L - D$ is the number of overlapping samples between segments.

3.4 MATLAB Functions for Random Signal Analysis

The following MATLAB functions implement several of the tools used in random signal analysis

- `c = xcorr(x,y,maxlags)`
- `[Pxx,f] = periodogram(x,window,nfft,fs)`
- `[Pxx,f] = pcov(x,p,nfft,fs)`
- `[Pxx,f] = pwelch(x,window,noverlap,nfft,fs)`

Note that MATLAB normalizes the periodogram by the sampling rate. Thus you may need to multiply the periodogram values by f_s in order to be consistent with the definition of the periodogram.