Filter Banks, Short-Time Fourier Analysis, and the Phase Vocoder

Henry D. Pfister

April 18, 2018

1 Introduction

Many real-world signals (e.g., speech and music) have different properties on different time scales. Over short periods of time, many signals look like excited linear time-invariant (LTI) systems. While, over longer periods of time, their behavior can be much more complex.

For example, speech signals are generated by your vocal chords exciting the resonant cavity formed by your throat, mouth, and nose. Over short periods of the time, the result is essentially a filtered version of the excitation signal. Moreover, the excitation signal is well-modeled by either a periodic pulse train (from your vocal chords) or a noise signal generated by turbulence as air from your lungs is forced through a construction formed by your tongue and lips.

Thus, it is common to model such signals as excited LTI systems where the excitation and system vary much more slowly than the signal. For speech signals, the effective bandwidth of the signal is roughly 4 KHz (e.g., digital telephony often uses a sample rate of 8 KHz) while the parameters of the excitation and filter are essentially constant over periods of 20-40 ms (25-50 Hz). One can exploit this structure by breaking the signal into overlapping segments of 20-40 ms and computing the discrete Fourier transform (DFT) of each segment. This approach is called "short-time Fourier analysis".

In many ways, music also has a similar structure. Individual instruments typically generate roughly periodic waveforms by exciting an oscillatory physical system (e.g., plucking a string or blowing air to excite a resonant cavity). Music is formed by adding these waveforms together. Over short periods of time, Fourier analysis captures the sum of the spectral shapes of the instruments. Over longer periods of time, the changes in these spectral shapes define the music.

2 Filter Bank Analysis

The processing and analysis of audio often involves decomposing the audio signal into multiple channels based on frequency content. A filter bank is a typical method for achieving such a decomposition. Moreover, uniform filter banks can be implemented efficiently using the fast Fourier transform (FFT) and the resulting analysis is sometimes called the "short-time Fourier transform" (STFT).

Consider a discrete-time signal $x[n]$ and and a bank of N filters with impulse responses $h_k[n]$ for $k = 0, 1, \ldots, N-1$. The N output signals from the *analysis* filter bank are given by

$$y_k[n] = \sum_{\ell=-\infty}^{\infty} h_k[\ell] x[n-\ell],$$

for $k = 0, 1, \ldots, N-1$. A filter bank is called *uniform* if all the filters can be constructed by frequency shifting a single filter

$$H_k\left(e^{j\omega}\right) = H_0\left(e^{j(\omega - 2\pi k/N)}\right).$$

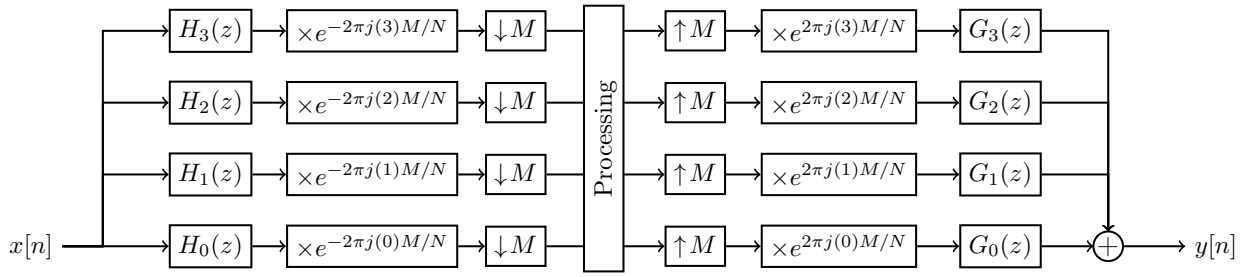

Figure 1: Example filter bank with $N = 4$.

In that case, the modulation property of the DTFT implies that

$$h_k[n] = h_0[n]e^{2\pi jkn/N}.$$

If $h_0[n]$ is a low-pass filter whose response is centered at DC, then $h_k[n]$ is a band-pass filter whose response is centered at $\omega_k = 2\pi k/N$. If $H_0(e^{j\omega})$ has a narrow bandwidth, then the signals $y_k[n]$ are oversampled relative to their bandwidth and their sample rate can be reduced. In particular, we define

$$z_k[m] = e^{-2\pi jkmM/N}y_k[mM]$$

to be the result of frequency shifting the k-th output to DC and then decimating by M. To make things work out nicely, we assume additionally that M divides N.

Now, we can put all of this together. To do this, we first choose $h_0[n] = w[-n]$ where $w[n]$ is a real window function that satisfies $w[n] = 0$ if $n < 0$ or $n \geq N$. Then, we can write the n-th output of the filter bank as

$$\begin{aligned}
z_k[m] &= e^{-2\pi jkmM/N}y_k[mM] \\
&= e^{-2\pi jkmM/N}\sum_{n'=-\infty}^{\infty} h_k[n']x[mM - n'] \\
&= e^{-2\pi jkmM/N}\sum_{n'=-\infty}^{\infty} w[-n']e^{2\pi jkn'/N}x[mM - n'] \\
&= e^{-2\pi jkmM/N}\sum_{n=0}^{N-1} x[mM + n]w[n]e^{-2\pi jkn/N}.
\end{aligned}$$

Note: The analysis filters, $h_k[n]$, are chosen to be non-causal in order to the match the standard definition of the STFT given below.

3 Short-Time Fourier Analysis

The STFT is computed by breaking the input signal, $x[n]$, into length-N blocks (shifted by M), windowing, and computing the DFT of each block. The m-th block is denoted by

$$x_m[n] = x[mM + n]$$

and its DFT (after windowing) is given by

$$\begin{aligned}
X_m[k] &= \sum_{n=0}^{N-1} x_m[n]w[n]e^{-2\pi jkn/N} \\
&= \sum_{n=0}^{N-1} x[mM + n]w[n]e^{-2\pi jkn/N}.
\end{aligned}$$

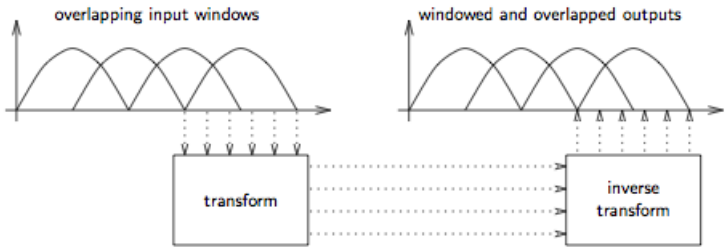

Figure 2: Illustration of STFT and inverse with $M = N/2$.

Thus, $z_k[m] = e^{-2\pi jkMm/N} X_m[k]$ and the STFT generates essentially the same output as the uniform filter bank described above! The only difference is that the multiplier $e^{-2\pi jkMm/N}$ is used to cancel the predictable phase shift associated with that frequency bin.

To get a better feel for the STFT, consider the case where the input $x[n] = e^{2\pi jk_0 n/N}$ is a complex exponential and the window function is $w[n] = u[n] - u[n-N]$. In this case, we find that

$$\begin{aligned} X_m[k] &= \sum_{n=0}^{N-1} x_m[n] w[n] e^{-2\pi jkn/N} \\ &= \sum_{n=0}^{N-1} e^{2\pi jk_0(mM+n)/N} e^{-2\pi jkn/N} \\ &= e^{2\pi jk_0 mM/N} \sum_{n=0}^{N-1} e^{2\pi j(k_0-k)n/N} \\ &= N\delta[(k-k_0) \bmod N] e^{2\pi jk_0 mM/N} \end{aligned}$$

and $z_k[m] = e^{-2\pi jkmM/N} N\delta[k-k_0] e^{2\pi jk_0 mM/N} = N\delta[k-k_0]$. Thus, for a complex-exponential input whose frequency matches a DFT bin, the output $z_k[m]$ is constant.

4 Short-Time Fourier Reconstruction

Now, we consider the reconstruction of $x[n]$ from either $X_m[k]$ (or equivalently $z_k[m]$). The key observation is that, for each m, one can invert the DFT $X_m[k]$ to get the input signal $x_m[n]w[n]$. Thus, the inversion process will provide N/M different ways to compute $x[mM+n]$. To understand this process, it helps to assume that, before reconstruction, some processing is applied to the signal in the STFT (or filter bank) domain. Let $\tilde{X}_m[k]$ denote the STFT coefficients after processing. The processing results in a desired change (e.g., time-scale modification) but it also introduces some distortion (e.g., modeled some additive noise). For example, a reasonable model is that $\tilde{X}_m[k] = X_m[k] + V_m[k]$ where $V_m[k]$ are samples of uncorrelated zero-mean Gaussian noise with variance σ^2. A good reconstruction process should minimize the effect of this noise.

The inversion process for the STFT starts by inverting each DFT to get the estimates

$$\hat{x}_m[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_m[k] e^{2\pi jkn/N}.$$

Since the sample $x[n]$ appears in N/M distinct overlapping blocks, the reconstructed samples $\hat{x}_m[n]$ contain N/M different observations of $x[n]$. Next, we combine these multiple observations into a single estimate $\hat{x}[n]$

by computing the weighted average

$$\begin{aligned}\hat{x}[n] &= \sum_{m=-\infty}^{\infty} w[n-mM]\hat{x}_m[n-mM] \\ &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]\hat{x}_m[n-mM], \end{aligned} \tag{1}$$

where we can reduce the range of summation because $x[n]$ appears in N/M blocks and block $\lfloor n/M \rfloor$ is the last block containing sample $x[n]$. Intuitively, this formula forms a weighted average of the observations of $x[n]$ that appear in N/M adjacent blocks. A more detailed analysis shows that the chosen weights maximize the signal-to-noise ratio of the estimate assuming that processing is corrupted by uncorrelated zero-mean Gaussian noise with variance σ^2.

To understand this process better, we can try to carry out the whole process at once with

$$\begin{aligned}\hat{x}[n] &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]\hat{x}_m[n-mM] \\ &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_m[k]e^{2\pi jk(n-mM)/N} \\ &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]\frac{1}{N}\sum_{k=0}^{N-1}\left(X_m[k]+V_m[k]\right)e^{2\pi jk(n-mM)/N} \\ &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]\left(x_m[n-mM]w[n-mM]+v_m[n-mM]\right) \\ &= x[n]\sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]^2 + \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]v_m[n-mM]. \end{aligned}$$

In this expression, the first term is the signal term and the second term is the noise term. It turns out that both terms depend mainly on the value of

$$S[n] \triangleq \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]^2.$$

In the first term, this is obvious. Since the second term is the sum of independent zero-mean random variables with variance σ^2, it follows that the variance of the sum is given by $\sigma^2 S[n]$.

Thus, to simplify further, we compute

$$\begin{aligned}
S[n] &= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]^2 \\
&= \sum_{m'=0}^{N/M-1} w[n-(m'+\lfloor n/M \rfloor - N/M+1)M]^2 \\
&= \sum_{m'=0}^{N/M-1} w[n-\lfloor n/M \rfloor M + N - (m'+1)M]^2 \\
&= \sum_{m'=0}^{N/M-1} w[n \bmod M + N - (m'+1)M]^2 \\
&= \sum_{m=0}^{N/M-1} w[n \bmod M + mM]^2.
\end{aligned}$$

It is important to notice that this expression depends on n only through the expression $n \bmod M$. Thus, we only need to compute its value for $n \in \{0, 1, \ldots, M-1\}$. Also, the value of this expression depends on the choice of $w[n]$ and one popular choice for the window function is

$$w[n] = \sqrt{\frac{2M}{N}} \sin\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right). \tag{2}$$

This is because, for all $n \in \{0, 1, \ldots, M-1\}$, we can compute

$$\begin{aligned}
S[n] &= \sum_{m=0}^{N/M-1} w[n+mM]^2 \\
&= \frac{2M}{N} \sum_{m=0}^{N/M-1} \sin\left(\frac{\pi}{N}\left(n+mM+\frac{1}{2}\right)\right)^2 \\
&= \frac{2M}{N} \sum_{m=0}^{N/M-1} \frac{1-\cos\left(\frac{2\pi}{N}\left(n+mM+\frac{1}{2}\right)\right)}{2} \\
&= 1 - \frac{M}{N} \sum_{m=0}^{N/M-1} \cos\left(\frac{2\pi}{N}\left(n+mM+\frac{1}{2}\right)\right) \\
&= 1 - \frac{M}{N} \sum_{m=0}^{N/M-1} \left(e^{2\pi j(n+1/2)/N} e^{2\pi jmM/N} + e^{-2\pi j(n+1/2)/N} e^{-2\pi jmM/N}\right) \\
&= 1.
\end{aligned}$$

For this window, we see that the signal portion of $\hat{x}[n]$ equals $x[n]$ and the noise portion has variance equal to σ^2.

5 Filter Bank Reconstruction

The filter bank reconstruction of $x[n]$ is based on reversing each step of the analysis process. Starting with $z_k[m]$, we define $\hat{z}_k[n]$ with

$$\hat{z}_k[n] = \delta[n - \lfloor n/M \rfloor M] e^{2\pi jkM\lfloor n/M \rfloor/N} z_k[\lfloor n/M \rfloor].$$

Thus, the phase rotation is directly inverted and the decimation procedure is reversed by adding zeros between the samples. Next, we reverse the filtering process by convolving with the reconstruction filters $g_k[n] = h_k[-n]^*$ to get

$$\begin{aligned}
\hat{y}_k[n] &= \sum_{\ell=-\infty}^{\infty} g_k[n-\ell]\hat{z}_k[\ell] \\
&= \sum_{\ell=-\infty}^{\infty} g_k[n-\ell]\delta[\ell - \lfloor \ell/M \rfloor M] e^{2\pi jkM\lfloor \ell/M \rfloor/N} z_k[\lfloor \ell/M \rfloor] \\
&= \sum_{m=-\infty}^{\infty} g_k[n-mM] e^{2\pi jkmM/N} z_k[m] \\
&= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] e^{2\pi jk(n-mM)/N} e^{2\pi jkmM/N} z_k[m] \\
&= e^{2\pi jkn/N} \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] z_k[m].
\end{aligned}$$

Finally, the all of the reconstruction filters are averaged to give

$$\begin{aligned}
\hat{x}[n] &= \frac{1}{N}\sum_{k=0}^{N-1} \hat{y}_k[n] \\
&= \frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi jkn/N} \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] z_k[m] \\
&= \frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi jkn/N} \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] e^{-2\pi jkmM/N} \sum_{n'=0}^{N-1} x[mM+n']w[n']e^{-2\pi jkn'/N} \\
&= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] \sum_{n'=0}^{N-1} x[mM+n']w[n'] \frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi jk(mM-n+n')/N} \\
&= \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM] \sum_{n'=0}^{N-1} x[mM+n']w[n']\delta[(mM-n+n') \bmod N] \\
&= x[n] \sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]^2.
\end{aligned}$$

Thus, the reconstruction is perfect as long as

$$\sum_{m=\lfloor n/M \rfloor - N/M+1}^{\lfloor n/M \rfloor} w[n-mM]^2 = 1$$

for all $n \in \{0, 1, \ldots, M-1\}$. As we saw previously, this condition is satisfied by (2).

6 The Phase Vocoder

The phase vocoder was introduced in 1966 by Flanagan and Golden [1]. The goal of the phase vocoder is to change the duration of an audio signal without changing its pitch. In combination with resampling, this

also allows one to change the pitch of a signal without changing its duration. The basic idea is decompose the signal using a STFT, interpolate the STFT parameters to the new timing, and then reconstruct the modified signal. Let α be the time-scale factor (e.g., $\alpha = 1/2$ corresponds to making the sound twice as long while $\alpha = 2$ corresponds to making it twice as long). The interpolation of the STFT parameters occurs in the energy/phase space defined by

$$E_m[k] = |X_m[k]|^2$$
$$P_m[k] = \angle X_m[k].$$

Let t denote a continuous "block" parameter and define the interpolated energy as

$$\tilde{E}_t[k] = (\lceil t \rceil - t)E_{\lfloor t \rfloor}[k] + (1 - \lceil t \rceil + t)E_{\lceil t \rceil}[k].$$

Similarly, the delta phase is defined to be

$$D_m[k] = (P_m[k] - P_{m-1}[k] - 2\pi kM/N) \bmod 2\pi,$$

where $\theta \bmod 2\pi \in [-\pi, \pi]$ can be implemented in Matlab as `mod(theta+pi)-pi`. It can help to think of $D_m[k]$ as an estimate of the derivative of the phase evaluated at time $t = m - \frac{1}{2}$ (relative to the input signal). We note that the $-2\pi kM/N$ term in $D_m[k]$ occurs for the same reason as the $e^{-j2\pi kmM/N}$ term in $z_k[m]$. Namely, it shifts the frequency of the k-th filter bank output to DC before interpolation.

Then, the interpolated delta phase and the accumulated phase are defined by

$$\tilde{D}_t[k] = \left(\left\lceil t + \frac{1}{2} \right\rceil - t - \frac{1}{2}\right) D_{\lfloor t+\frac{1}{2} \rfloor}[k] + \left(1 - \left\lceil t + \frac{1}{2} \right\rceil + t + \frac{1}{2}\right) D_{\lceil t+\frac{1}{2} \rceil}[k]$$
$$\tilde{P}_{\ell+1}[k] = \tilde{P}_\ell[k] + \tilde{D}_{\alpha(\ell+\frac{1}{2})}[k] + 2\pi kM/N,$$

where we initialize $\tilde{P}_0[k] = P_0[k]$ and $\tilde{P}_{\ell+1}[k]$ is the phase estimate at time $t = \ell\alpha$ based on accumulating the delta phase interpolated to midpoints between samples. We note that the $2\pi kM/N$ term in $\tilde{P}_\ell[k]$ shifts the frequency of the downconverted k-th filter bank output from DC back to the correct center frequency.

Due to boundary effects, the first block of the input sequence should be assigned the block index of -1. This guarantees that we can linearly interpolate the delta phase to $t = \frac{\alpha}{2}$ from D_0 and D_1 (and that D_0 is computable). This process can be used to compute $\tilde{P}_\ell[k]$ as long as there is a block with index $\lceil \alpha(\ell - \frac{1}{2}) + \frac{1}{2} \rceil$. This condition defines the natural termination point of the process. Finally, we reconstruct the resulting signal by applying (1) to

$$\tilde{X}_\ell[k] = e^{j\tilde{P}_\ell} \sqrt{\tilde{E}_{\alpha\ell}[k]}.$$

Due to the above resampling the length of $\hat{x}[n]$ will be roughly a factor $\frac{1}{\alpha}$ longer than the original signal. To shift the pitch without changing the time scale, one should use band-limited interpolation to create the output signal $\hat{x}[n/\alpha]$.

The phase vocoder can also be used to implement the following effects:

- Harmony: For audio clips of singing, each note may shifted in frequency so the addition creates harmonies.
- Chorus: An audio clip of singing may be shifted slightly in time and frequency and then added together. This emulates the sound of many different voices singing together.
- Robot: Set all interpolated phases $\tilde{D}_{\alpha m}[k] = 0$.
- Whisper: Replace all phases by uniform random variables on $[0, 2\pi]$ and post-process with a high-pass filter to match the spectrum of a whisper excitation.

References

[1] J. L. Flanagan and R. Golden, "Phase vocoder," vol. 45, no. 9, pp. 1493–1509, 1966.