Multi-stage Interpolation and Decimation*

Phil Schniter

Abstract

This module covers the fundamentals of multistage decimation.

1 Multistage Decimation

In the single-stage interpolation structure illustrated in Figure 1, the required impulse response of $H(z)$ can be very long for large L.

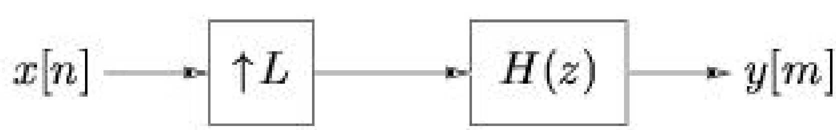

Figure 1

Consider, for example, the case where $L = 30$ and the input signal has a bandwidth of $\omega_0 = 0.9\pi$radians. If we desire passband ripple $\delta_p = 0.002$ and stopband ripple $\delta_s = 0.001$, then Kaiser's formula approximates the required FIR filter length to be

$$N_h \simeq \frac{-10\log(\delta_P \delta_S) - 13}{2.3\Delta(\omega)} \simeq 900$$

choosing $\Delta(\omega) = \frac{2\pi - 2\omega_0}{L}$ as the width of the first transition band (*i.e.*, ignoring the other transition bands for this rough approximation). Thus, a polyphase implementation of this interpolation task would cost about 900 multiplies per input sample.

Consider now the two-stage implementation illustrated in Figure 2.

*Version 2.13: Oct 4, 2005 9:57 am -0500
†http://creativecommons.org/licenses/by/1.0

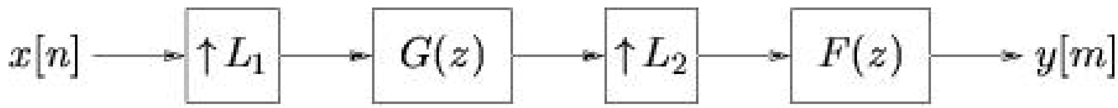

Figure 2

We claim that, when L is large and ω_0 is near Nyquist, the two-stage scheme can accomplish the same interpolation task with less computation.

Let's revisit the interpolation objective of our previous example. Assume that $L_1 = 2$ and $L_2 = 15$ so that $L_1 L_2 = L = 30$. We then desire a pair $\{F(z), G(z)\}$ which results in the same performance as $H(z)$. As a means of choosing these filters, we employ a Noble identity to reverse the order of filtering and upsampling (see Figure 3).

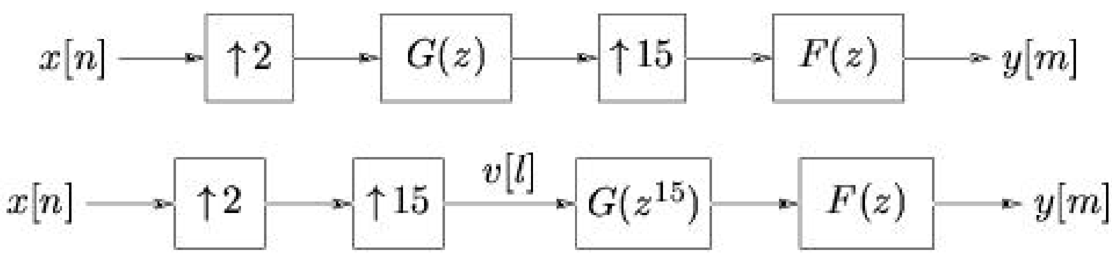

Figure 3

It is now clear that the composite filter $G\left(z^{15}\right) F(z)$ should be designed to meet the same specifications as $H(z)$. Thus we adopt the following strategy:

1. Design $G\left(z^{15}\right)$ to remove unwanted images, keeping in mind that the DTFT $G\left(e^{i15\omega}\right)$ is $\frac{2\pi}{15}$-periodic in ω.
2. Design $F(z)$ to remove the remaining images.

The first and second plots in Figure 4 illustrate example DTFTs for the desired signal $x[n]$ and its L-upsampled version $v[l]$, respectively. Our objective for interpolation, is to remove all but the shaded spectral image shown in the second plot. The third plot (Figure 4) shows that, due to symmetry requirements $G\left(z^{15}\right)$ will be able to remove only one image in the frequency range $\left[0, \frac{2\pi}{15}\right)$. Due to its periodicity, however, $G\left(z^{15}\right)$ also removes some of the other undesired images, namely those centered at $\frac{\pi}{15} + m\frac{2\pi}{15}$ for $m \in \mathbb{Z}$. $F(z)$ is then used to remove the remaining undesired images, namely those centered at $m\frac{2\pi}{15}$ for $m \in \mathbb{Z}$ such that m is not a multiple of 15. Since it is possible that the passband ripples of $F(z)$ and $G\left(z^{15}\right)$ could add constructively, we specify $\delta_p = 0.001$ for both $F(z)$ and $G(z)$, half the passband ripple specified for $H(z)$. Assuming that the transition bands in $F(z)$ have gain no greater than one, the stopband ripples will not be amplified and we can set $\delta_s = 0.001$ for both $F(z)$ and $G(z)$, the same specification as for $H(z)$.

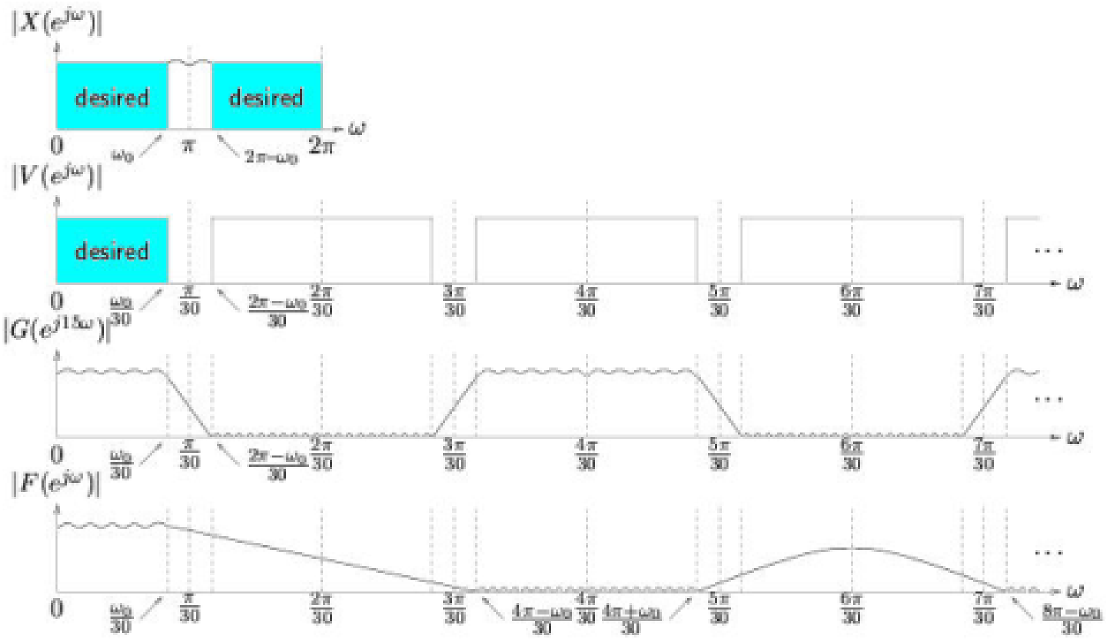

Figure 4

The computational savings of the multi-stage structure result from the fact that the transition bands in both $F(z)$ and $G(z)$ are much wider than the transition bands in $H(z)$. From the block diagram (Figure 4), we can infer that the transition band in $G(z)$ is centered at $\omega = \frac{\pi}{2}$ with width $\pi - \omega_0 = 0.1\pi\text{rad}$. Likewise, the transition bands in $F(z)$ have width $\frac{4\pi - 2\omega_0}{30} = \frac{2.2}{30}\pi\text{rad}$. Plugging these specifications into the Kaiser length approximation, we obtain

$$N_g \simeq 64$$

and

$$N_f \simeq 88$$

Already we see that it will be much easier, computationally, to design two filters of lengths 64 and 88 than it would be to design one 900-tap filter.

As we now show, the computational savings also carry over to the operation of the two-stage structure. As a point of reference, recall that a polyphase implementation of the one-stage interpolator would require $N_h \simeq 900$ multiplications per input point. Using a cascade of two single-stage polyphase interpolators to implement the two-stage scheme, we find that the first interpolator would require $N_g \simeq 64$ per input point $x[n]$, while the second would require $N_f \simeq 88$ multiplies per output of $G(z)$. Since $G(z)$ outputs two points per input $x[n]$, the two-stage structure would require a total of

$$\simeq 64 + 2 \times 88 = 240$$

multiplies per input. Clearly this is a significant savings over the 900 multiplies required by the one-stage structure. Note that it was advantageous to choose the first upsampling ratio (L_1) as small as possible, so that the second stage of interpolation operates at a low rate.

Multi-stage decimation can be formulated in a very similar way. Using the same example quantities as we did for the case of multi-stage interpolation, we have the block diagrams and filter-design methodology illustrated in Figure 5 and Figure 6.

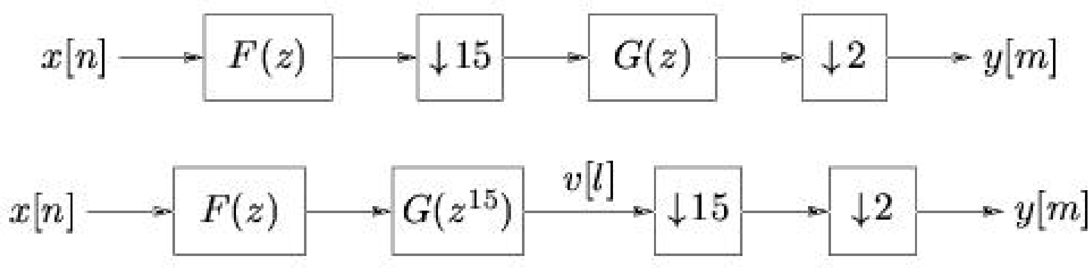

Figure 5

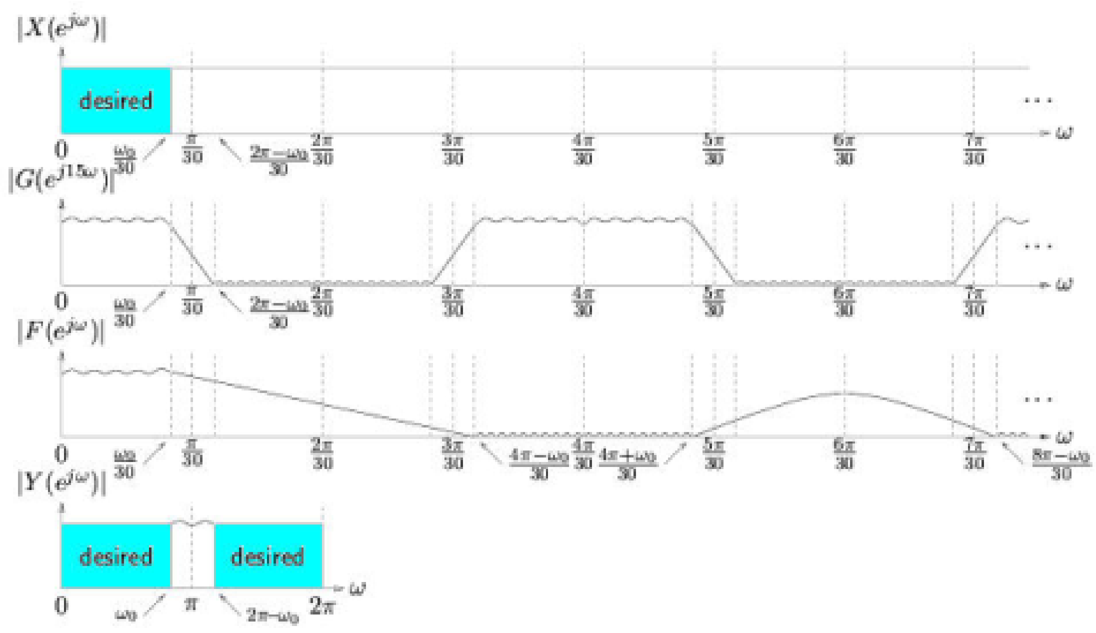

Figure 6