**The Cooper Union Department of Electrical Engineering**
**Prof. Fred L. Fontaine**
**ECE310 Digital Signal Processing**
**Problem Set III: Multirate Systems**
October 22, 2023

1. We want to perform a fractional sampling rate conversion of a real signal from $16kHz$ to $20kHz$. The original signal is assumed to have been properly bandlimited when sampled at $16kHz$ so no aliasing occurred. That is the bandwidth of the input signal we are assuming!

   We want to use a minimum number of separate filtering stages, and minimum value decimation factors for downsampling and upsampling stages, subject to the constraint that (assuming ideal filtering) either **no** information is lost or, if that is unavoidable, the smallest possible signal bandwidth would need to be discarded to avoid aliasing.

   (a) Draw a block diagram showing the proper process, including the decimation factors and the cutoff frequency (in digital radian form) for any filters. Also, specify the sampling rate at which any filter is operating.

   (b) Will this require any loss of information? If yes, what band of frequencies must be discarded (that is, assume [if necessary] we only keep a frequency band of the form $|f| \leq f_{\max}$ where $f_{\max}$ is less than the original signal bandwidth).

2. Repeat the above the process for converting $20kHz$ to $16kHz$. [We start with a signal sampled at $20kHz$ whose bandwidth is the largest it could be without aliasing at that initial step]

3. If $h[n]$ is a **real** FIR filter of length $L$ stored in a MATLAB row vector $h = [h[0], h[1], \cdots, h[L-1]]$, associated with $H(z) = h[0] + \cdots + h[L-1]z^{-(L-1)}$, then how is *fliplr(h)* related to $\tilde{H}(z)$?

4. Let $e_k[n]$, $0 \leq k \leq 2$, be the (Type I) polyphase components of $h[n]$ relative to a decimation factor of 3. We can write:

$$\begin{bmatrix} E_0(\omega) \\ E_1(\omega) \\ E_2(\omega) \end{bmatrix} = A \begin{bmatrix} H\left(\frac{\omega}{3}\right) \\ H\left(\frac{\omega}{3} - \frac{2\pi}{3}\right) \\ H\left(\frac{\omega}{3} - \frac{4\pi}{3}\right) \end{bmatrix}$$

   for some matrix $A$. Find $A$. **Hint:** From $e_0 = (\downarrow 3)h$, you should already know how to express $E_0(\omega)$ in terms of the quantities on the right. Then use the more general relation:

$$e_k = \left[(\downarrow 3) z^k\right] h$$

   which means first map $h[n] \rightarrow h[n+k]$, then decimate by 3 to get $e_k[n] = h[3n+k]$; this will allow you to express any $E_k(\omega)$ in terms of the quantities on the right.

   Continuing this problem, verify the following:

$$A = \frac{1}{3} diag\left\{1, e^{j\frac{\omega}{3}}, e^{j\frac{2\omega}{3}}\right\} \mathbf{W}_3$$

   where $\mathbf{W}_3$ is the $3 \times 3$ DFT matrix.

5. $\boxed{\textbf{Wavelets and Filter Banks}}$

Although there are many variations on *wavelets,* the basic form is closely connected with paraunitary PR filter banks. [Wavelets comprise orthonormal bases of certain orthogonal spaces in $L^2(\mathbb{R})$; the operations performed by the digital filter banks correspond to projection operators associated with these spaces] Consider the two channel analysis and synthesis filter banks shown in the figure below, with $H_0, F_0$ lowpass and $H_1, F_1$ highpass filters. The following code will produce the filter coefficients for Daubechies wavelets order $N$: [note- this uses functions in the Wavelets Toolbox; also here 'order' does not refer to a filter order, but what is called the order of the wavelet]

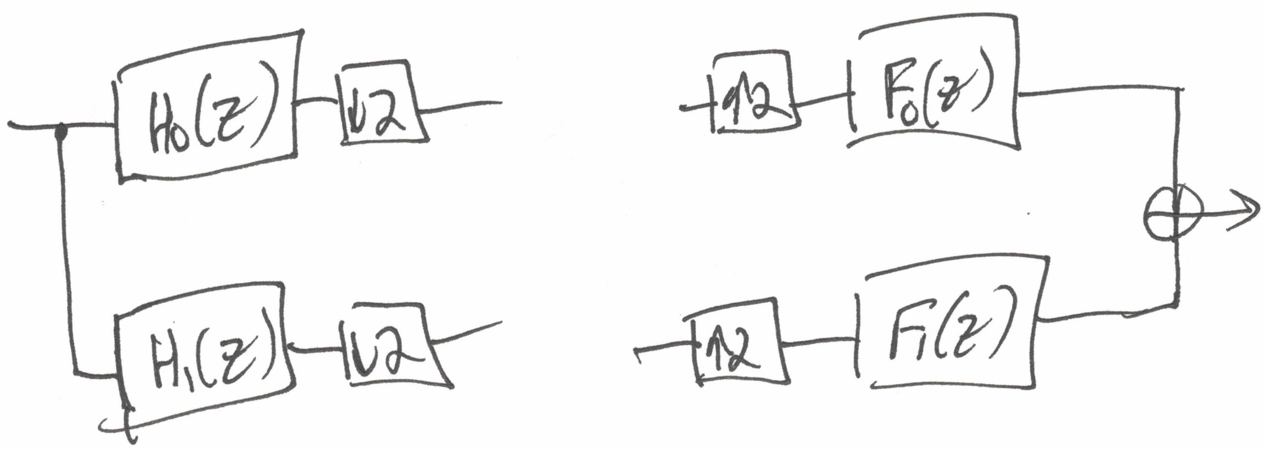
```
N= 5;  % substitute a value between 1 and 45

wname= ['db',int2str(N)];

[h0,h1,f0,f1]= wfilters(wname);
```

As a remark, the wavelets for the case *db*1 are associated with subspaces of $L^2(\mathbb{R})$ comprised of **step functions** with discontinuities at points of the form $m/2^k$, $m, k \in \mathbb{Z}$; these are called *Haar wavelets.* Higher order wavelets are smoother (differentiable up to certain orders). Here we focus on the filters, however. Wavelets that are zero outside a finite interval in $\mathbb{R}$ are associated with FIR filters; Daubechies wavelets were originally designed to achieve maximal smoothness given the length of the associated FIR filter.

In the frequency domain plots for this problem, the magnitude should be on a linear (not decibel) scale, frequency over the range $0 \le \omega \le \pi$ (rad), with $10^4$ points.

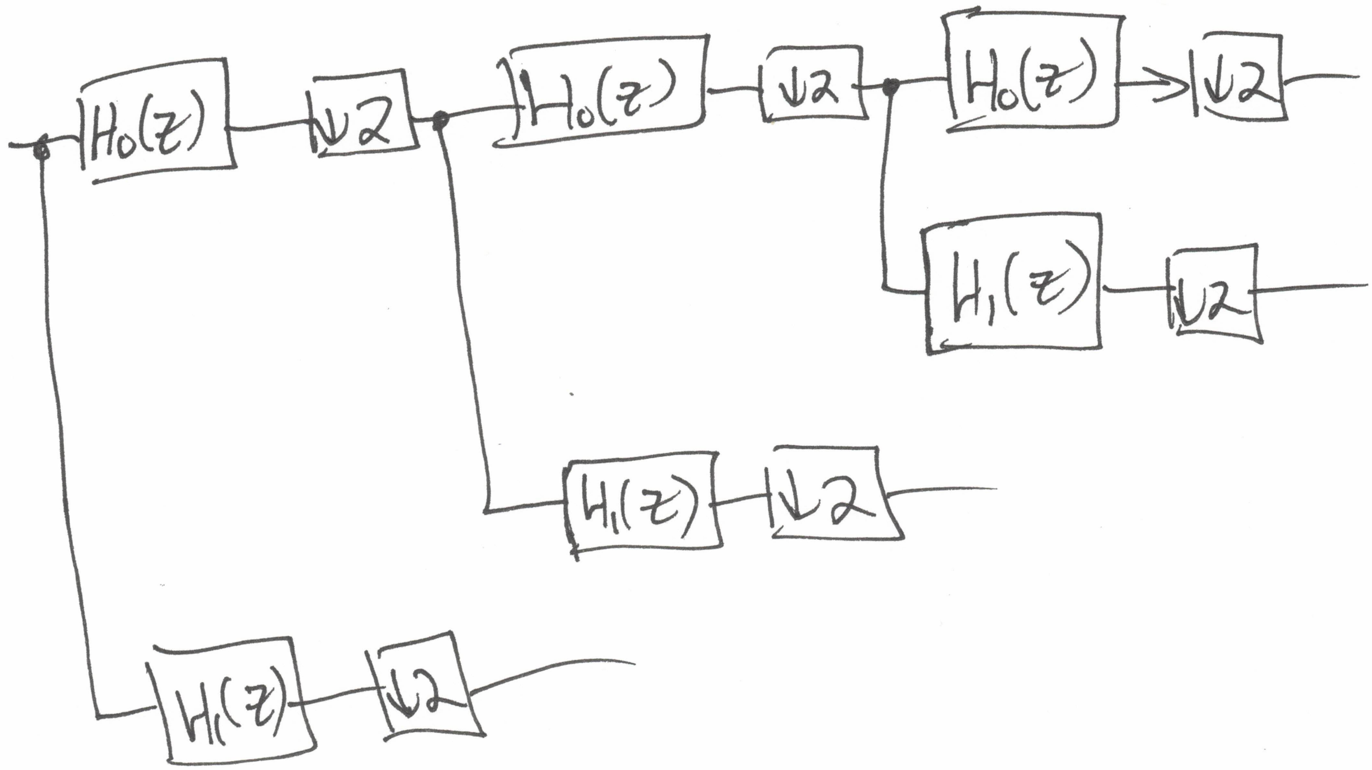



(a) The Haar wavelet filters are fairly basic. Write $H_0(z)$ and $H_1(z)$. Compute and superimpose plots for $|H_0(\omega)|$, $|H_1(\omega)|$ as described above. Also, generate the polyphase matrix $E(z)$ (it should have a simple representation!) and verify the paraunitary PR property (by hand), working in the $z$-domain.

(b) MATLAB normalizes the filters so $|H_0(\omega)|^2 + |H_1(\omega)|^2$ achieves a specific value. Compute this for the Haar wavelet filters, call it say $\mathcal{P}(\omega)$ (the power). Compute $\max(\mathcal{P}(\omega)) - \min(\mathcal{P}(\omega))$ and *mean* $(\mathcal{P}(\omega))$ to find this constant, and check that the variation (due to numeric error) is small.

(c) Now generate *db5* filters.

1. Look at $h_0, h_1, f_0, f_1$, also look at the *db1* case. First, propose the lengths of *dbN* filters in terms of $N$. Second, propose general relations for $h_1, f_0, f_1$ each in terms of $h_0$ for the general *dbN* case. For example, one of them will have the form $h_0[N_1 - n]$ for some $N_1$ in terms of $N$.
2. Break down the db5 filters $h_0, h_1$ into Type I polyphase components. We want to create a polyphase matrix $E(z)$ in MATLAB. You can use one of several approaches: (1)symbolic toolbox, (2)cell (3)three dimensional array (if the polyphase components do not have equal length, you would need to pad with 0s to make them equal length; however, in this case they will be same length so you don't need to do that). In any case, since $E(z)$ is $2 \times 2$, feel free to construct it "manually" (i.e., entry by entry) rather than writing a general loop or something. [I don't mean manually typing numbers in; I mean you can construct the 11, 12, 21, 22 elements separately, if necessary]
3. Compute $\tilde{E}E$ and check that is satisfies the condition for a PR paraunitary filter bank, as described in the notes. Again you can do the equivalent of the matrix multiply term by term, rather than a general loop, if you prefer. **Remark:** I suggest writing out the matrix multiply in terms of individual $ij$ entries; then figure out how to achieve the "multiplications". For example, if in symbolic form, you may need to ask MATLAB to do some simplifications to get nice results; if the entries are stored as vectors of polynomial coefficients, how would you represent polynomial multiplication in terms of these vectors? Keep in mind there may be small numerical error. If it is small, assume it is 0 and report it as such.
4. If we have an analysis/synthesis filter bank with the *db5* filters, what is the end-to-end delay?
5. Superimpose plots of $|H_0(\omega)|$ and $|H_1(\omega)|$ for db5.
6. Check the PR property for db5, as you have already done for db1.
7. Consider a three level tree structure as shown below, where the filters shown correspond to *db5*. Collapse this to a single stage with a bank of filters say $G_k(z)$ with respective decimators $\downarrow M_k$ (the decimation factors may be different).Now obtain an expression for the frequency response of each channel in terms of the underlying $H_0(\cdot)$ and $H_1(\cdot)$ in the frequency domain. Compute the magnitude response of each branch, and superimpose them in the frequency domain. What happens to the passband widths as the frequency increases? **Hint:** If we change a filter $H(z) \rightarrow H(z^M)$, what is the equivalent frequency domain mapping of $\omega$? As you compute the results here, feel free to call *freqz* multiple times (as often as you need to). **DO THIS WITHOUT DIRECTLY EVALUATING THE $g_k[n]$ FILTERS IN THE TIME DOMAIN!** Instead, express the $G_k(\omega)$'s as products of terms, each of which can be computed via *freqz*.
8. This filter bank has a kind of power complementary property, but with some scaling necessary. For the $k^{th}$ channel, say a filter $G_k(z)$ is associated with a




decimation factor $M_k$. Compute the sum:

$$\mathcal{P}(\omega) = \sum_k \frac{1}{M_k} |G_k(\omega)|^2$$

Compute $\max(\mathcal{P}) - \min(\mathcal{P})$ to convince yourself it is constant (up to numerical error), and find the nominal value as $mean(\mathcal{P})$. **Remark:** We can properly view this as a weighted average. Specifically, since the filter bank is maximally decimated, what is $\sum \frac{1}{M_k}$?