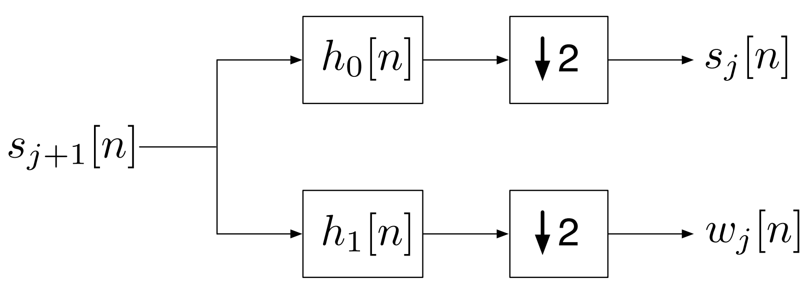
Haar wavelet filterbanks

Recall that we can compute the scaling coefficients $s_{j,n}$ and wavelet coefficients $w_{j,n}$ at scale j from the scaling coefficients $s_{j+1,n}$ at scale $j+1$:

$$
\begin{aligned}
s_{j,n} &= \frac{1}{\sqrt{2}}\, s_{j+1,2n} + \frac{1}{\sqrt{2}}\, s_{j+1,2n+1} \\
w_{j,n} &= \frac{1}{\sqrt{2}}\, s_{j+1,2n} - \frac{1}{\sqrt{2}}\, s_{j+1,2n+1}.
\end{aligned}
$$

If we think of the scaling/wavelet coefficients at scale j as a discrete time sequence, so $s_j[n] := s_{j,n}$ and $w_j[n] := w_{j,n}$, then the expressions above suggest that the scaling coefficients at scale j can be broken down scaling and wavelet coefficients at scale $j-1$ using **filters** and **downsampling** arranged in the following architecture:

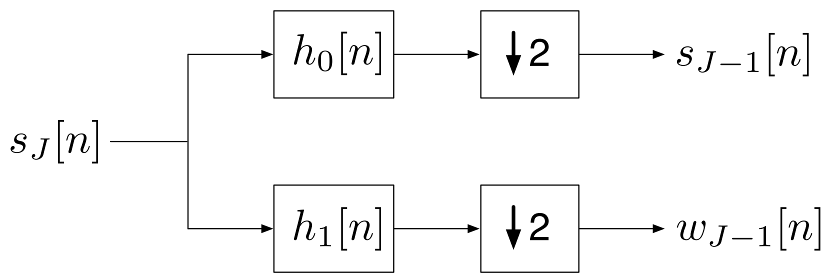


where the $\downarrow 2$ block means "downsample by 2" and the impulse responses for the filters are:

$$
h_0[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, -1 \\ 0 & \text{otherwise} \end{cases} \qquad h_1[n] = \begin{cases} -\frac{1}{\sqrt{2}} & n = -1 \\ \frac{1}{\sqrt{2}} & n = 0 \\ 0 & \text{otherwise.} \end{cases}
$$

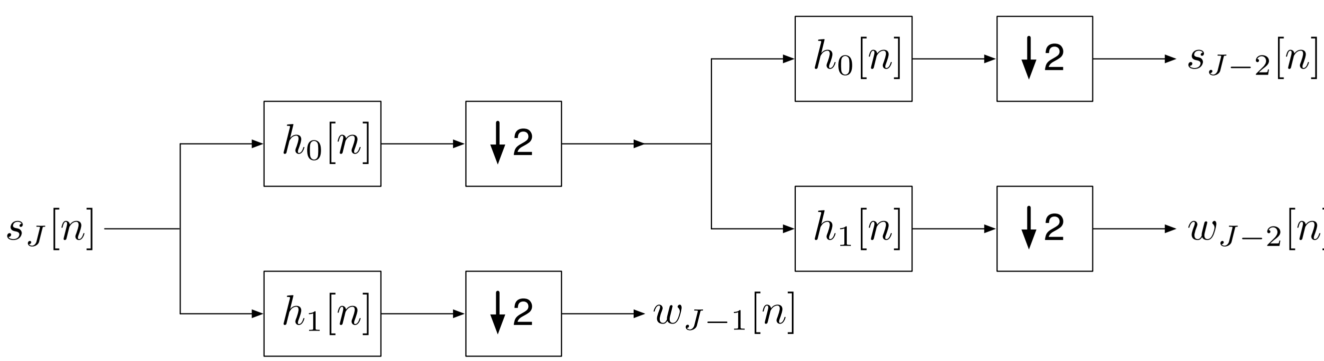
Of course, we can continue on and break up the $\{s_{j,n}\}$ into scaling and wavelet coefficients at the next coarsest scale. This gives rise to a **filter bank** structure that we can associate with each of the ways we can write the approximation at scale J, $\hat{x}_J(t) = \boldsymbol{P}_{\mathcal{V}_J}[x(t)]$.

$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_{J-1}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{J-1,n}\phi_{J-1,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-1,n}\psi_{J-1,n}(t)\end{aligned}$$



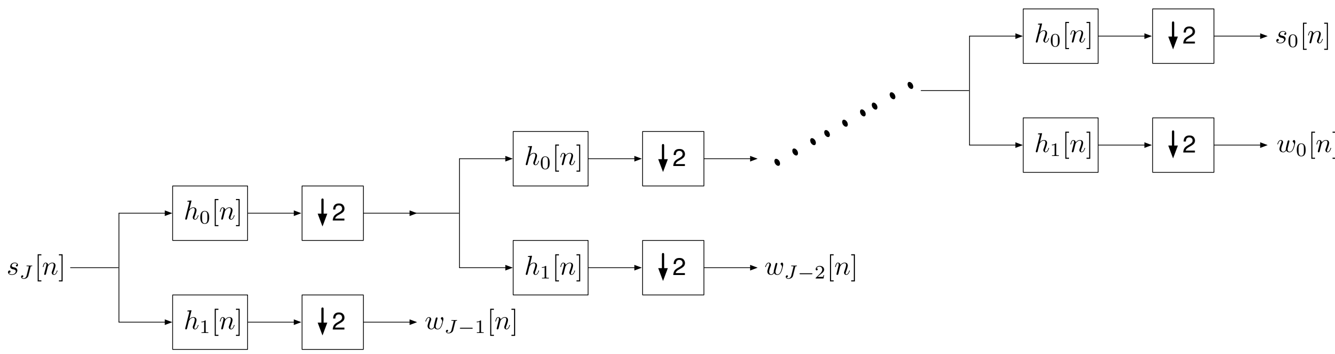
Iterating this process on $s_{J-1}[n]$ we obtain

$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_{J-2}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-2}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{J-2,n}\phi_{J-2,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-2,n}\psi_{J-2,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-1,n}\psi_{J-1,n}(t)\end{aligned}$$



We can continue this process to obtain

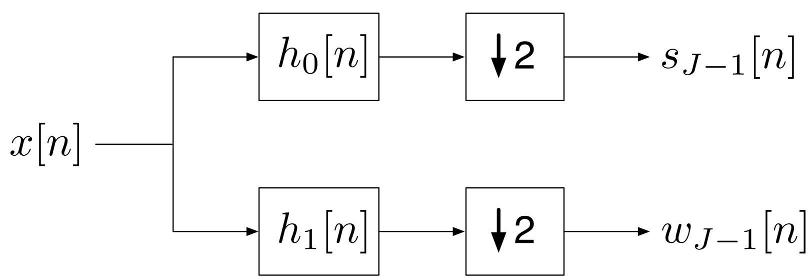
$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_0}[x(t)] + \boldsymbol{P}_{\mathcal{W}_0}[x(t)] + \cdots + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{0,n}\phi_{0,n}(t) + \sum_{j=0}^{J-1}\sum_{n=-\infty}^{\infty} w_{j,n}\psi_{j,n}(t)\end{aligned}$$



This provides an extraordinarily efficient way to compute the full set of scaling and wavelet coefficients given an initial approximation at scale J.

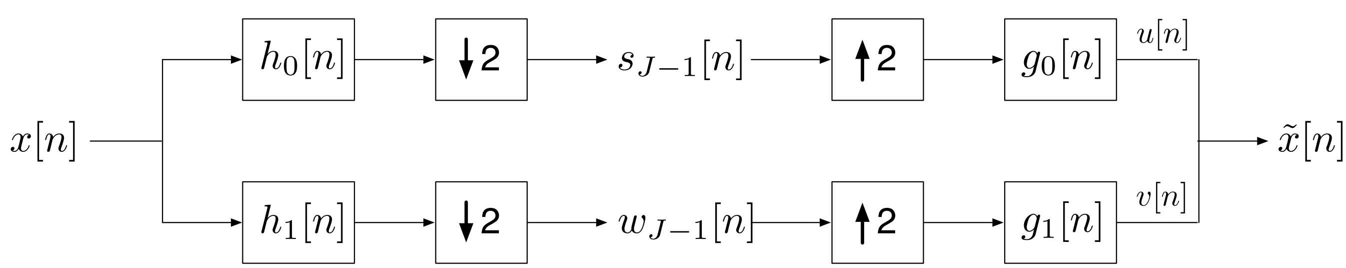
The discrete Haar transform

The connection to filter banks above gives us a natural way to define a wavelet transform for discrete-time signals. Basically, we just treat $x[n]$ like it was a sequence of scaling coefficients at fine scale, then apply as many levels of the filter bank as we like. So the following structure:



takes $x[n]$ and transforms it into two sequences, $s_{J-1}[n]$ and $w_{J-1}[n]$, each of which have **half the rate** of the input.

How do we **invert** this particular transform? With another filter filter bank. Consider the following structure:



If we take

$$g_0[n] = h_0[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

$$g_1[n] = h_1[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ -\frac{1}{\sqrt{2}} & n = 1 \\ 0 & \text{otherwise} \end{cases},$$

then we will have $\tilde{x}[n] = x[n]$. To see this, recall that

$$s_{J-1}[n] = \frac{1}{\sqrt{2}}(x[2n] + x[2n+1]),$$

and so

$$u[n] = \begin{cases} \frac{1}{2}\left(x[n] + x[n+1]\right) & n \text{ even} \\ \frac{1}{2}\left(x[n-1] + x[n]\right) & n \text{ odd} \end{cases},$$

that is, the values in $u[n]$ appear in pairs,

$$u[0] = u[1] = \frac{1}{2}(x[0] + x[1]), \quad u[2] = u[3] = \frac{1}{2}(x[2] + x[3]), \text{ etc.}$$

Similarly, since

$$w_{J-1}[n] = \frac{1}{\sqrt{2}}\left(x[2n] - x[2n+1]\right),$$

we have

$$v[n] = \begin{cases} \frac{1}{2}\left(x[n] - x[n+1]\right) & n \text{ even} \\ \frac{1}{2}\left(-x[n-1] + x[n]\right) & n \text{ odd} \end{cases},$$

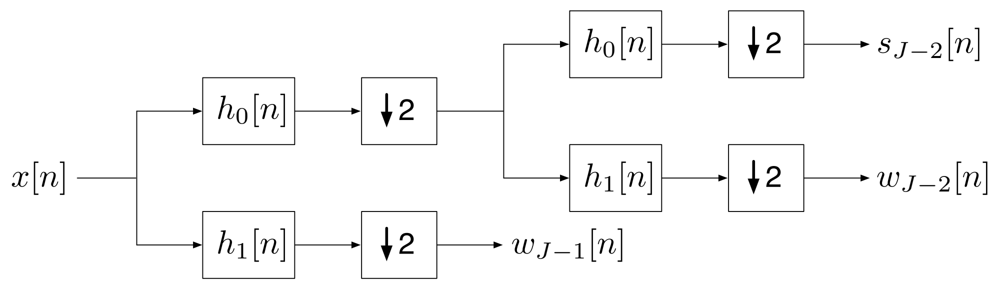
that is, the values in $v[n]$ appear in pairs of $\pm$ terms,

$$v[0] = -v[1] = \frac{1}{2}(x[0] - x[1]), \quad v[2] = -v[3] = \frac{1}{2}(x[2] - x[3]), \text{ etc.}$$

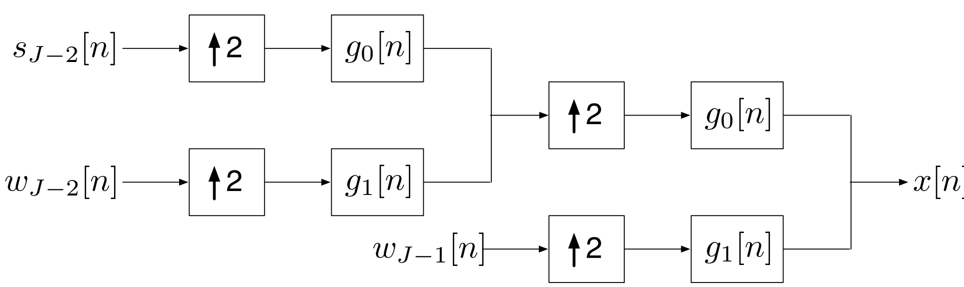
Now it is easy to see that

$$\tilde{x}[n] = u[n] + v[n] = x[n] \quad \text{for all } n \in \mathbb{Z}.$$

We can repeat this to as many levels as we desire. For example, the following structure



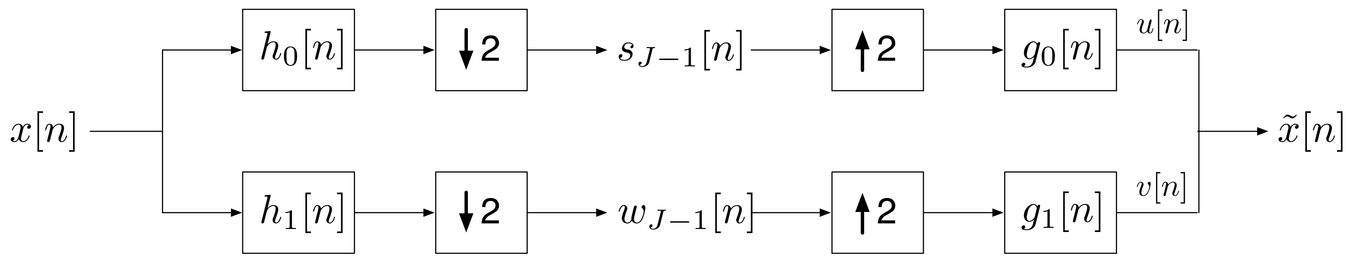
takes $x[n]$ and transforms it into three sequences, one of which is at half the rate of $x[n]$, and the other two are at a quarter of the rate. To invert it, we simply apply the inverse filter bank twice:



It should be clear how to extend this to an arbitrary number of levels.

Perfect reconstruction filterbanks

The filterbank architecture that defines the discrete Haar transform suggests a natural way in which to generalize our discussion of Haar wavelets. Specifically, recall the architecture:



You might wonder if there are other choices for the filters $h_0[n]$, $h_1[n]$, $g_0[n]$, and $g_1[n]$ that would correspond to alternative wavelet decompositions. We will see that this is indeed the case, but of course this will only be true when the filters satisfy certain restrictions.

To get some intuition for this, first we will consider a simpler question: when can we show that the above architecture gives rise to a **perfect reconstruction filterbank**, by which we simply mean a filterbank satisfying $\tilde{x}[n] = x[n]$?

In order to answer this question, we need to think a bit more carefully about what happens when we **downsample** and **upsample** a discrete-time sequence. In analyzing this as well as the full filterbank architecture, it will be much cleaner to use the $\boldsymbol{z}$**-transform**.

The z-transform

Recall that for a discrete-time signal $x[n]$, the z-transform is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n},$$

where z is a complex number.[1]

We can think of the z-transform as a generalization of the DTFT. Specifically, by evaluating the z-transform $X(z)$ at $z = e^{j\omega}$ we obtain the DTFT of $x[n]$, i.e. $X(e^{j\omega})$.

The z-transform also generalizes the familiar property of the DTFT that convolution in time is equivalent to multiplication in frequency. Specifically, if $y[n]$ denotes the convolution of $x[n]$ with $h[n]$, then we have

$$Y(z) = X(z)H(z).$$

Downsampling

Now consider the process of taking a signal $x[n]$ and **downsampling** it by a factor of 2. Specifically, let

$$y[n] = x[2n].$$

What is the relationship between $X(z)$ and $Y(z)$?

[1] For any given z, this sum may or may not converge, and so we also associate with $X(z)$ a *region of convergence* (which will depend on $x[n]$) that tells us the set of possible z for which the sum converges. Fortunately, we will not need to worry much about this.

Observe that

$$
\begin{aligned}
Y(z) &= \sum_{n=-\infty}^{\infty} x[2n]z^{-n} \\
&= \sum_{\ell \text{ even}} x[\ell]z^{-\ell/2} \\
&= \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell]z^{-\ell/2} + \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](-1)^{-\ell}z^{-\ell/2} \\
&= \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](z^{1/2})^{-\ell} + \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](-z^{1/2})^{-\ell} \\
&= \frac{1}{2}\left[X(z^{1/2}) + X(-z^{1/2})\right].
\end{aligned}
$$

This may seem a bit difficult to interpret, but things are a bit clearer when we look at the DTFT. Specifically, if we let $z = e^{j\omega}$, then

$$
X(z^{1/2}) = X(e^{j\omega/2}) \qquad\qquad X(-z^{1/2}) = X(-e^{j\omega/2}) = X(e^{j\omega/2+\pi}).
$$

Note that $X(e^{j\omega/2})$ is simply a dilated version of $X(e^{j\omega})$. The $X(e^{j\omega/2+\pi})$ term corresponds to a dilation of $X(e^{j\omega})$ shifted by π. This corresponds to exactly what one would have obtained if $x[n]$ corresponded to samples of a continuous-time signal which we then sampled at half of the original sampling rate – the spectrum is dilated (because of the lower sampling rate) but there is also potential *aliasing*, which is accounted for by the $X(e^{j\omega/2+\pi})$ term.

Upsampling

Now we turn to the problem of taking a signal $x[n]$ and **upsampling** it by a factor of 2. By this we mean generating a signal

$$y[n] = \begin{cases} x[n/2] & n \text{ even} \\ 0 & n \text{ odd} \end{cases}.$$

We again ask the question: what is the relationship between $X(z)$ and $Y(z)$?

The answer is straightforward:

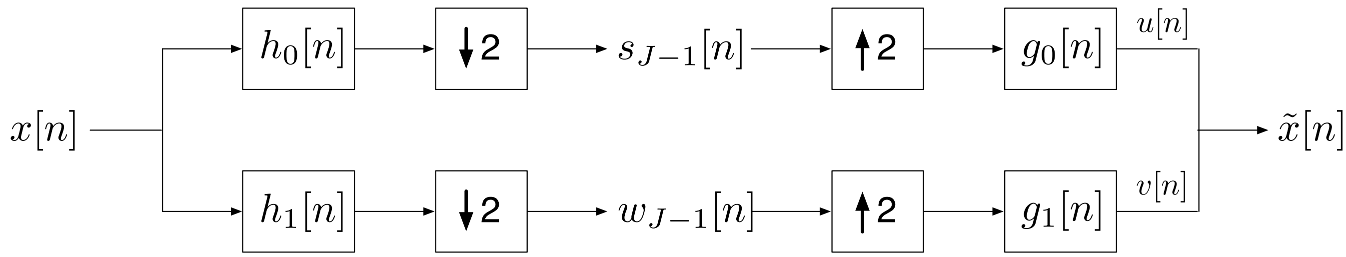
$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y[n]z^{-n} \\ &= \sum_{n \text{ even}} x[n/2]z^{-n} \\ &= \sum_{\ell=-\infty}^{\infty} x[\ell]z^{2\ell} \\ &= \sum_{\ell=-\infty}^{\infty} x[\ell](z^2)^{\ell} \\ &= X(z^2) \end{aligned}$$

Note that this implies a *compression* of the DTFT:

$$X(e^{j\omega}) \to X(e^{j2\omega}).$$

Perfect reconstruction conditions

We are now in a position to derive conditions on the filters in a filterbank (in terms of their z-transforms) that will ensure that we perfectly reconstruct the input. Recall the architecture:



We want to ensure that $\tilde{x}[n] = x[n]$. If we use causal filters, this is not quite possible and we instead relax our notion of perfect reconstruction to instead require $\tilde{x}[n] = x[n-m]$ for some delay m.

Towards this end, note that we can write

$$S_{J-1}(z) = \frac{1}{2}\left[H_0(z^{1/2})X(z^{1/2}) + H_0(-z^{1/2})X(-z^{1/2})\right]$$

and thus

$$U(z) = \frac{1}{2}G_0(z)\left[H_0(z)X(z) + H_0(-z)X(-z)\right].$$

Similarly, we have

$$V(z) = \frac{1}{2}G_1(z)\left[H_1(z)X(z) + H_1(-z)X(-z)\right].$$

Combining these and rearranging we have

$$\begin{aligned}\widetilde{X}(z) = {} & \frac{1}{2}\left[G_0(z)H_0(z) + G_1(z)H_1(z)\right]X(z) \\ & + \frac{1}{2}\left[G_0(z)H_0(-z) + G_1(z)H_1(-z)\right]X(-z).\end{aligned}$$

We want to have $\widetilde{X}(z) = z^{-m}X(z)$. The way to make this occur is straightforward: the filters must satisfy the following **perfect reconstruction conditions**:

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0 \qquad \text{(Alias cancellation)}$$

and

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2z^{-m}. \qquad \text{(No distortion)}$$

How can we design filters that will satisfy these conditions? Suppose for the moment that the filters $G_0(z)$ and $G_1(z)$ are given – what is a natural choice for $H_0(z)$ and $H_1(z)$?

$$H_0(z) = G_1(-z) \qquad \text{and} \qquad H_1(z) = -G_0(-z).$$

With these choices we immediately have that

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = G_0(z)G_1(z) - G_1(z)G_0(z) = 0,$$

and thus the alias cancellation condition is satisfied.

What do these filters look like? Consider the FIR filters with z-transforms

$$\begin{aligned} G_0(z) &= \alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} \\ G_1(z) &= \beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} + \beta_3 z^{-3}. \end{aligned}$$

In this case,

$$\begin{aligned} H_0(z) &= \beta_0 - \beta_1 z^{-1} + \beta_2 z^{-2} - \beta_3 z^{-3} \\ H_1(z) &= -\alpha_0 + \alpha_1 z^{-1} - \alpha_2 z^{-2}. \end{aligned}$$

The question then becomes, how can we design $G_0(z)$ and $G_1(z)$ to ensure that they satisfy the no distortion condition? With the choices for $H_0(z)$ and $H_1(z)$ given above, we can write this condition as

$$T(z) = G_0(z)G_1(-z) - G_1(z)G_0(-z) = 2z^{-m}.$$

There are an endless possible variety of choices at this point. We will discuss the most common approach next time, but for now we simply note that if we fix $G_0(z)$, then our design problem reduces to constructing a $G_1(z)$ such that $T(z) = 2z^{-m}$, at which point the rest of the filterbank $H_0(z)$ and $H_1(z)$ are determined.