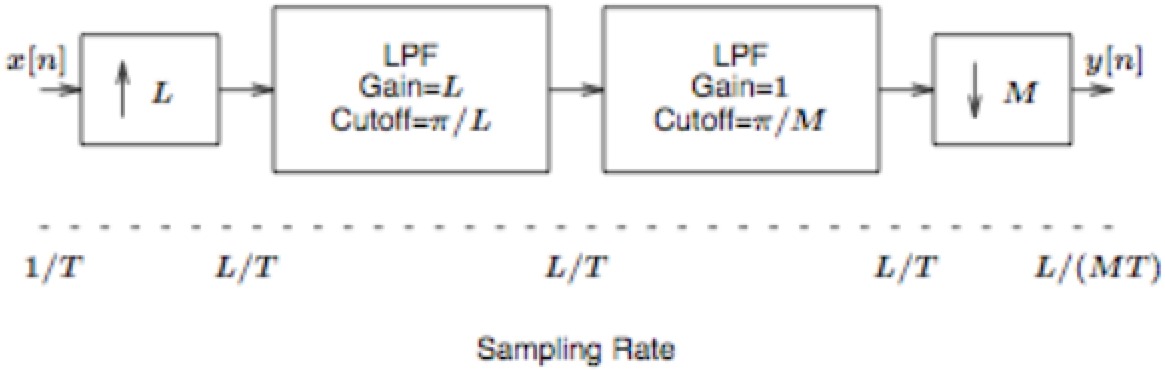
Polyphase filters

Lesson 7

Rate Conversion

Rate conversion looks like this:



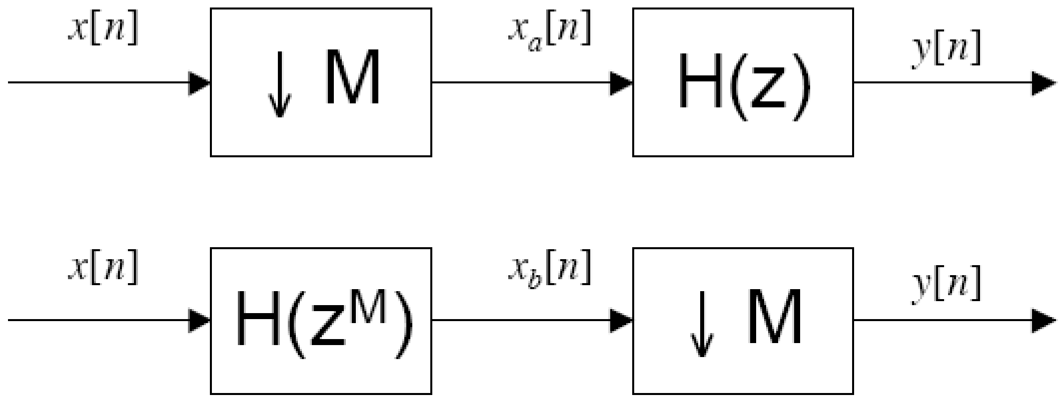
Simply cascade an interpolator and a decimator so that desired sampling rate factor is achieved (L/M)

Why is interpolation occuring first? Does it need to? (think Nyquist)

Note: the two lowpass filters can be combined into one:

- gain of L (for interpolator)
- cutoff frequency will be smaller of π/L and π/M (both need to be satisfied)

Filtering and Downsampling interchange



The above operations are equivalent:

$$X_b(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega})$$

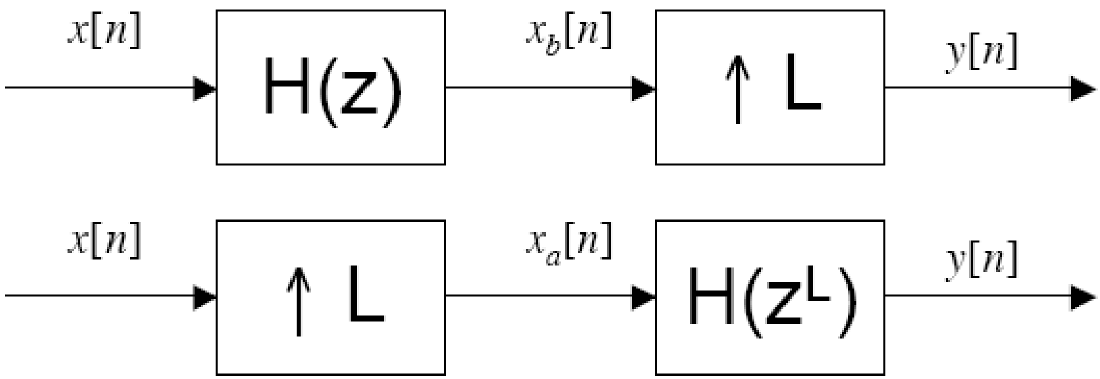
$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b(e^{j(\omega-2\pi i)/M})$$

$$= \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega-2\pi i)/M})H(e^{j(\omega-2\pi i)})$$

$$= \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega-2\pi i)/M})\left[H(e^{j\omega})\right]$$

$$= \left[H(e^{j\omega})\right]X_a(e^{j\omega})$$

Filtering and Upsampling interchange



$$Y\left(e^{i\omega}\right) = X_b\left(e^{i\omega L}\right) =$$

$$= X\left(e^{i\omega L}\right) H\left(e^{i\omega L}\right) =$$

$$= \left[X_a\left(e^{i\omega}\right)\right] H\left(e^{i\omega L}\right)$$

Thus, we can interchange up/downsampling with linear filtering as long as we modify the linear filter appropriately

Polyphase decomposition

The polyphase decomposition of a sequence involves splitting the sequence into M subsequences, each consisting of every Mth value of the sequence

Each subsequence is offset by one sample (hence, polyphase)

Since each subsequence is nonzero at only every Mth sample, the subsequences can be downsampled by a factor of M

- no data are lost

Polyphase Filtering

Define the M subsequences with subindex k for the impulse response, h[n], of a DT LTI filter:

$$h_k[n] = \begin{cases} h[n+k], & n = \text{ integer multiple of M} \\ 0, & \text{otherwise} \end{cases}$$

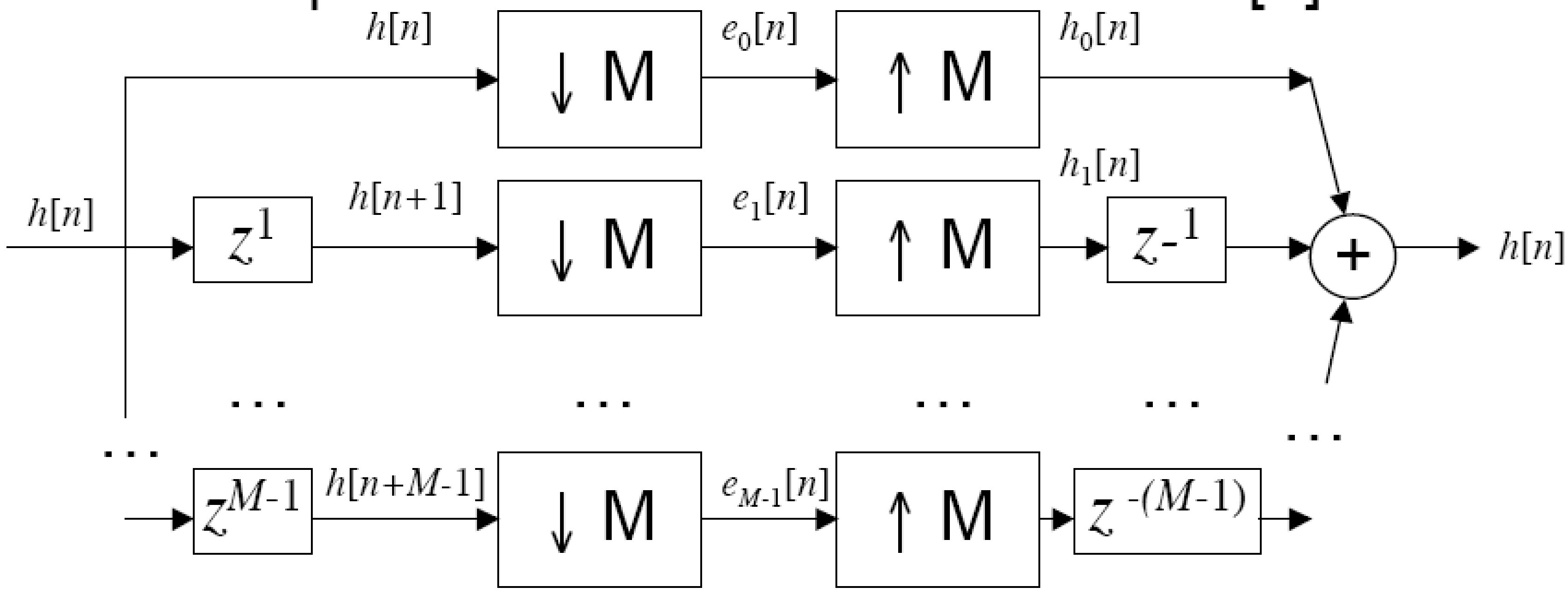
The impulse response, h[n], can be reconstructed:

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k]$$

Define: $e_k[n] = h[nM+k] = h_k[nM]$

Polyphase Filtering

which gives us the following picture of a polyphase decomposition and reconstruction of h[n]:

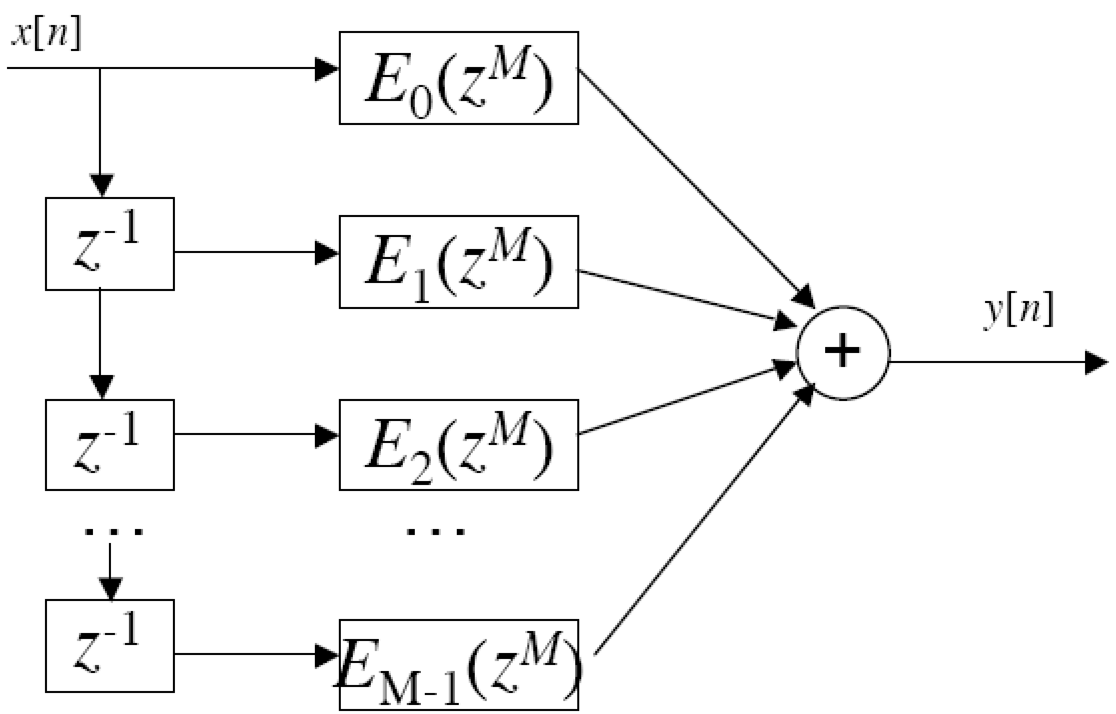


From the structure, we can see:

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

Polyphase filtering

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

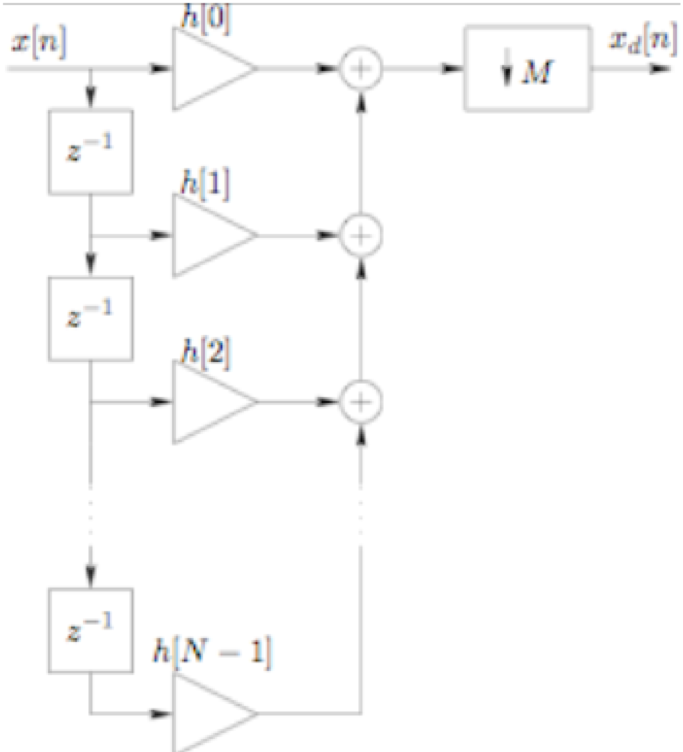


Example

Polyphase Implementation of Decimator

For filters whose output is downsampled, e.g. a decimator, significant efficiency can be achieved

Consider the following system, and assume the lowpass filter is an L-tap FIR filter:



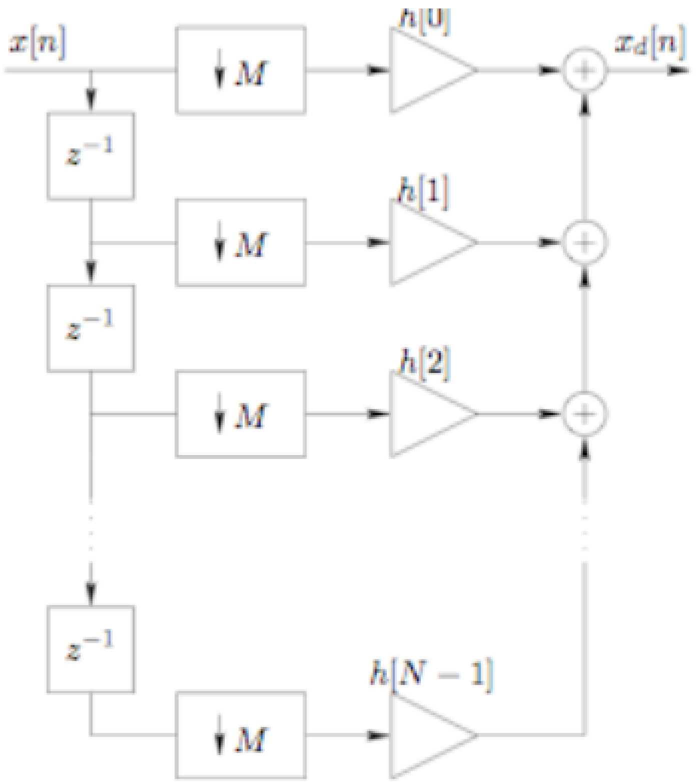
It seems wasteful to perform multiplications on so many samples that are thrown away

Example

Polyphase Implementation of Decimator

So apply earlier result and interchange the downsamplers with the multipliers

Now multipliers operating at 1/Mth the rate



Example

Suppose we had a requirement to construct a filter with hundreds of taps to operate at Gigabits/sec speed and that output is downsampled by a factor of 2

- e.g. to oversample an incoming analog carrier of digital data for channel equalization (need to downsample to data rate afterwards)

Not likely to be able to buy such a filter, but we can use polyphase decomposition to break the computation up so that it is manageable

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n}$$

Split into even and odd coefficients:

$$H(z) = \sum_{n=-\infty}^{\infty} h[2n] z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h[2n+1] z^{-2n}$$

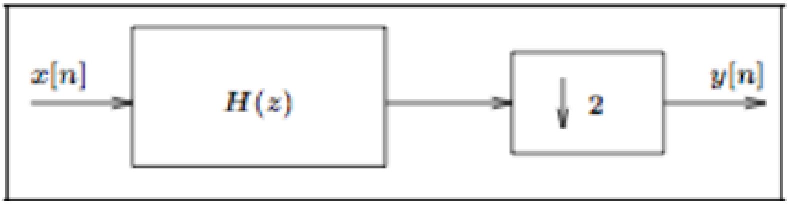
Example

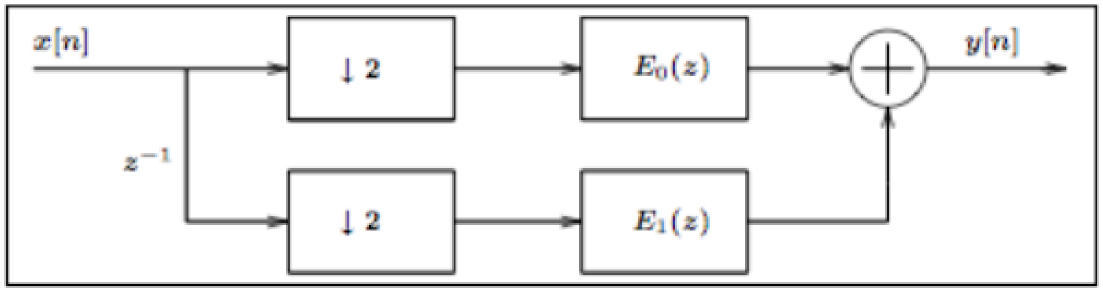
Define: $E_0(z) = \sum_{n=-\infty}^{\infty} h[2n] z^{-n}$

$$E_1(z) = \sum_{n=-\infty}^{\infty} h[2n+1] z^{-n}$$

$$H(z) = E_0(z^2) + z^{-1} E_1(z^2)$$

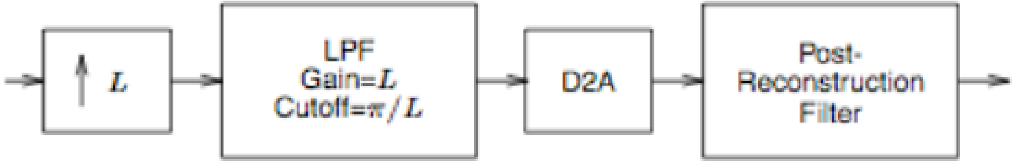
Then we can make the following substitution, halving the multipler data rate and the filter lengths:



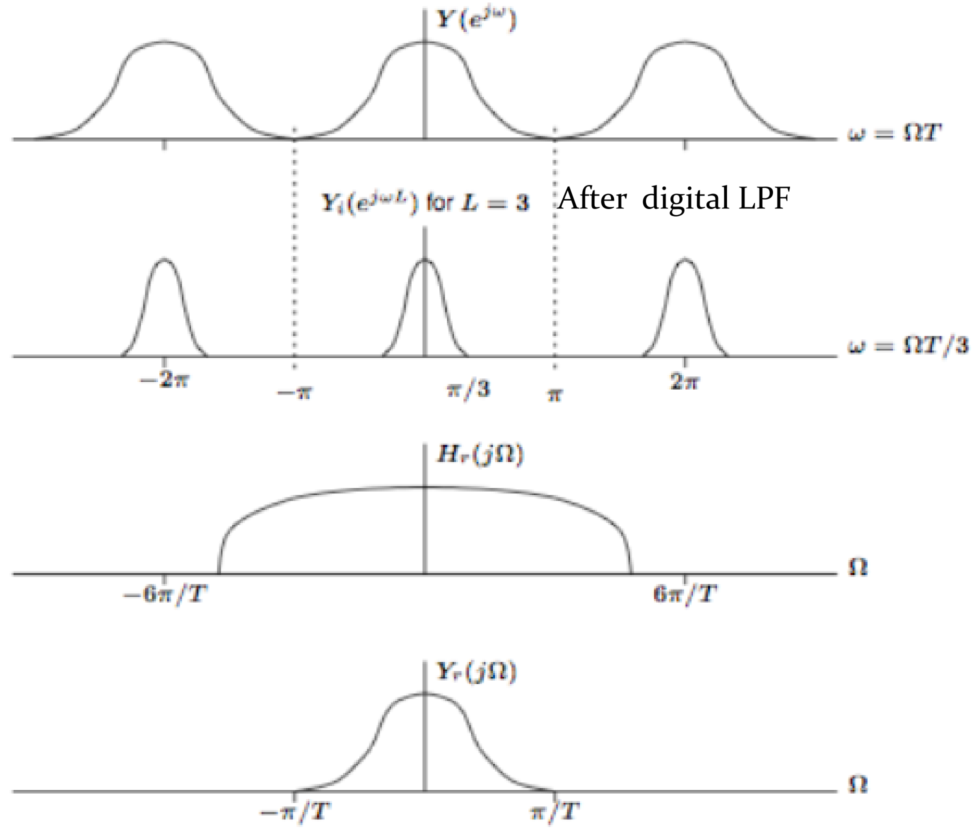


Example

Oversampled D/A converters



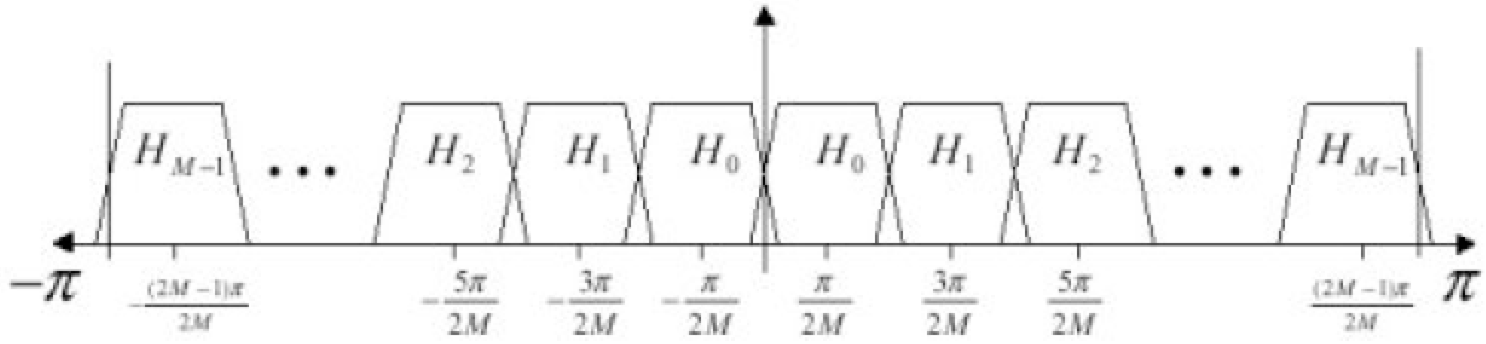
We can employ upsampling before the analog reconstruction filter to relax the filter requirements



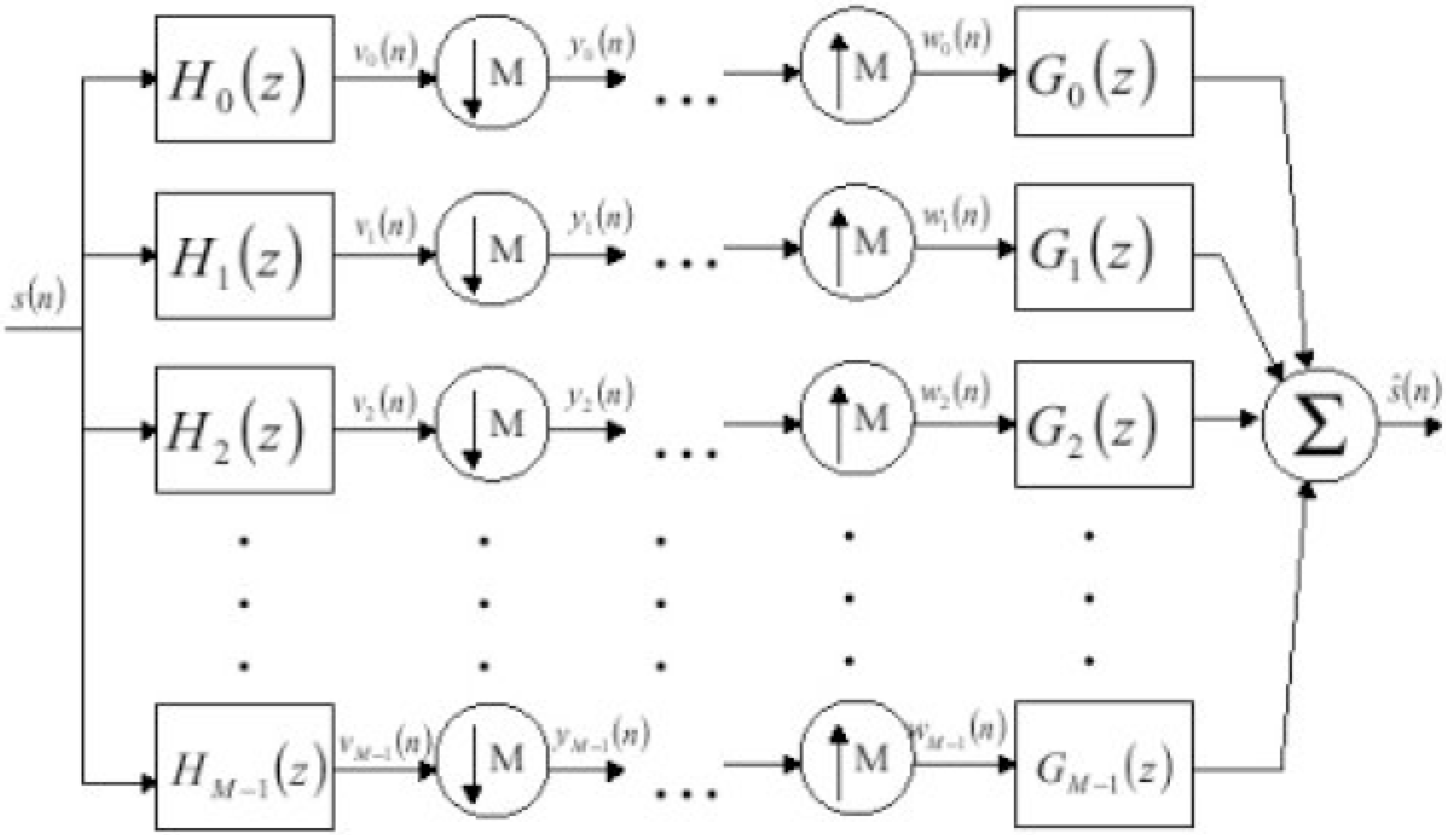
Filter banks

Downsampling and upsampling can be used to implement multi-rate filters, which decompose signals to various frequency resolutions

Essentially decompose input signal into multiple band-limited components by use of a bank of bandpass filters



Filter Banks



The output of bandpass filters is downsampled so that the total number of samples is the same as at input (no information lost)

Filter Banks

Filter banks can be designed to perfectly reconstruct the input at output

Why might filter banks be useful?

- Because of the quantized reality of digital signals, we may have a limited number of bits to encode the signal
- Some frequency bands may be more important, or require higher resolution and thus receive more coding bits than others
- Filter banks allow for this encoding scheme